Object viewpoint estimation in the wild

Yang Xiao

École des Ponts ParisTech

THÈSE DE DOCTORAT
de l'École des Ponts ParisTech

Object Viewpoint Estimation in the Wild

École doctorale N° 532, MSTIC

Signal, Image, Automatique

Thèse préparée au sein du LIGM-IMAGINE / École des Ponts ParisTech

Thèse soutenue le 12 Octobre 2021, par
Yang XIAO

Composition du jury :

Vincent, LEPETIT Directeur de recherche, ENPC	*Président*
Josef, SIVIC Directeur de recherche, CTU Prague / INRIA	*Rapporteur*
Matthieu, CORD Professeur, Université Sorbonne	*Rapporteur*
Nikos, KOMODAKIS Professeur adjoint, Université Crète	*Examinateur*
Slobodan, ILIC Chargée de recherche, TUM	*Examinateur*
Diane, LARLUS Chargée de recherche, Naver Labs Europe	*Examinatrice*
Renaud, MARLET Directeur de recherche, ENPC	*Directeur de thèse*

École Doctorale MSTIC
Mathématiques & Sciences et Technologies
de l'Information et de la Communication

Thèse de doctorat
de École des Ponts

Domaine : Traitement du Signal et des Images

Présentée par
Yang Xiao
pour obtenir le grade de

Docteur de École des Ponts

Object Viewpoint Estimation in the Wild

Soutenue publiquement le (date here) devant le jury composé de :

Josef SIVIC	Distinguished Researcher, Czech Technical University	Rapporteur
Matthieu CORD	Professor, Sorbonne University	Rapporteur
Nikos KOMODAKIS	Assistant Professor, University of Crete	Examinateur
Slobodan ILIC	Guest Research Scientist, Technical University of Munich	Examinateur
Diane LARLUS	Principal Research Scientist, Naver Labs Europe	Examinatrice
Vincent LEPETIT	Director of Research, École des Ponts	Président de Jury
Renaud MARLET	Senior Researcher, École des Ponts	Directeur de thèse

Abstract

The goal of this thesis is to develop deep-learning approaches for estimating the 3D pose (viewpoint) of an object pictured in an image in different situations: (i) the object location in the image and the exact 3D model of the corresponding object are known, (ii) both the object location and the class are predicted and an exemplar 3D model is provided for each object class, and (iii) no 3D model is used and object location is predicted without the object being classified into a specific category.

The key contributions of this thesis are the following. First, we propose a deep-learning approach to category-free viewpoint estimation. This approach can estimate the pose of any object conditioned only on its 3D model, whether or not it is similar to the objects seen at training time. The proposed network contains distinct modules for image feature extraction, shape feature extraction and pose prediction. These modules can have different variants for different representations of 3D models, but remain trainable end-to-end. Second, to allow inferring without exact 3D object models, we develop a class-exemplar-based viewpoint estimation approach that learns to condition the viewpoint prediction on the corresponding class feature extracted from a few 3D models during training. This approach differs from the previous approach in the sense that we extract an exemplar feature for each class instead of treating them independently for each object. We show that the proposed approach is robust against the precision of the provided 3D models and that can be adapted quickly to novel classes with using a few labeled examples. Third, we define a simple yet effective unifying framework that tackles both few-shot object detection and few-shot viewpoint estimation. We exploit, in a meta-learning setting, task-specific class information present in existing datasets, such as images with bounding boxes for object detection and exemplar 3D models of different classes for viewpoint estimation. And we propose a joint evaluation of object detection and viewpoint estimation in the few-shot regime. Finally, we develop a class-agnostic object viewpoint estimation approach that estimates the viewpoint directly from an image embedding, where the embedding space is optimized for object pose estimation through a geometry-aware contrastive learning. Rather than blindly pulling together features of the same object in different augmented views and pushing apart features of different objects while ignoring the pose difference between them, we propose a pose-aware contrastive loss that pushes away the image features of objects having different poses, ignoring the

class of these objects. By sharing the network weights across all categories during training, we obtain a class-agnostic viewpoint estimation network that can work on objects of any category. Our method achieve state-of-the-art results in the Pascal3D+, ObjectNet3D and Pix3D category-level object pose estimation benchmarks, under both intra-dataset and inter-dataset settings.

Résumé

Le but de cette thèse est de développer des approches d'apprentissage profond pour estimer la pose 3D (point de vue) d'un objet représenté dans une image dans différentes situations: (i) la localisation de l'objet dans l'image et le modèle 3D exact de l'objet correspondant sont connus, (ii) la localisation et la classe d'objet sont prédits et un exemplaire de modèle 3D est fourni pour chaque classe d'objets, et (iii) les modèles 3D ne sont pas pris en compte et seul la localisation de l'objet est prédite sans que l'objet soit classé dans une catégorie spécifique.

Les principales contributions de cette thèse sont les suivantes. Tout d'abord, nous proposons une approche d'apprentissage profond pour l'estimation du point de vue sans catégorie. Cette approche permet d'estimer la pose de tout objets conditionné uniquement sur son modèle 3D, qu'il soit similaire ou non aux objets vus au moment de l'apprentissage. Le réseau proposé contient des modules distincts pour l'extraction de caractéristiques d'image, l'extraction de caractéristiques de forme et la prédiction de pose. Ces modules peuvent avoir différentes variantes pour différentes représentations de modèles 3D, mais s'intègrent dans une architecture entraînable de bout en bout. Deuxièmement, pour permettre l'inférence sans modèle d'objet 3D exact, nous développons une approche d'estimation du point de vue basée sur des exemples de classe qui apprend à conditionner la prédiction du point de vue à des caractéristiques de la classe correspondante extraite de quelques modèles 3D pendant l'entraînement. Cette approche diffère de l'approche précédente en ce sens que nous extrayons des caractéristiques générales pour chaque classe au lieu de les traiter indépendamment pour chaque objet. Nous montrons que l'approche proposée est robuste par rapport à la précision des modèles 3D fournis et qu'elle peut être adaptée rapidement à de nouvelles classes avec seulement quelques exemples étiquetés. Troisièmement, nous définissons un cadre simple mais efficace qui traîte à la fois la détection d'objets et l'estimation du point de vue à partir de seulement un petit nombre d'images d'apprentissage. Nous exploitons, dans un contexte de méta-apprentissage, des informations de classe spécifiques aux tâches et présentes dans des bases de données existants, telles que des images avec des boîtes 2D pour la détection d'objets et des exemplaires de modèle 3D de différentes classes pour l'estimation du point de vue. De plus, nous proposons une évaluation conjointe de la détection d'objets et de l'estimation du point de vue pour le cas d'un très petit jeu de données d'apprentissage. Enfin, nous développons une

approche d'estimation du point de vue d'objet indépendante de la classe qui estime le point de vue directement à partir d'une représentation de l'image, où l'espace de représentations est optimisé pour l'estimation de la pose d'objet grâce à un apprentissage contrastif sensible à la géométrie. Plutôt que de rassembler aveuglément les représentations d'un même objet dans différentes vues augmentées et d'écarter les représentations d'objets differents tout en ignorant la différence de pose entre eux, nous proposons une fonction de perte contrastive sensible à la pose qui éloignent entre elles les représentations d'objets ayant des poses différentes, ignorant la classe de ces objets. En partageant les poids du réseau entre toutes les catégories pendant l'entraînement, nous obtenons un réseau d'estimation de point de vue indépendant de la classe qui peut fonctionner sur des objets de n'importe quelle catégorie. Notre méthode obtient des résultats à l'état de l'art pour l'estimation de pose 3D dans les benchmarks Pascal3D+, ObjectNet3D et Pix3D, à la fois pour chaque jeu de données indépendamment et entre jeux de données (en entraînant sur l'un et en testant sur l'autre).

Acknowledgments

I would like to thank Renaud for having believed in me and having decided to take me as a PhD student in the first place. Thanks for continuously guiding me to find interesting research directions, and for teaching me how to make right decisions and ask right questions. I feel extremely lucky to have you as my advisor and to talk with you about research ideas and academic experience. I really enjoyed working with you and I have definitely learned a lot from you during the three years.

I am grateful to all the collaborators that I have been lucky to work with: Mathieu, Vincent, Xi, Shell, Yuming, Xuchong, Pierre-Alain, Nguyen and Georgy. I would like to thank especially Mathieu for showing me the "deep" way to present research ideas with nice figures in a simple but effective fashion. And I owe a sincere thank you to Vincent for allowing me to participate in many interesting and challenging research projects, which not only broaden my research horizons but also improve my collaboration skills. It was a great pleasure to collaborate with all of you. And I am looking forwarding to working with you again.

I would also like to thnank Frederic and Staphane for hosting me as an applied scientist intern at Amazon. It was a great pleasure to work with you, and I learned a lot from you.

All members of IMAGINE, thanks for the best lab environment ever. Thanks to Xi and Shell for guiding me into the PhD student life and for introducing me into the local Chinese group at Champs-sur-Marne. And special thanks to Shell for bringing his daughter to join our group, which gave us lots of precious moments. Thanks to Thibault for bringing the encouraging vibe and for organizing the legendary California trip, those days are unforgettable. Thanks to my office-mates, Xuchong, Pierre-Alain, Francois, Tom, Hugo, and Yuming for sharing the space and fun moments. Especially thanks to Yuming for hosting me at Grenoble for several times, hiking in the mountains and *discovering* novel things is definitely something I would miss a lot.

I would also like to thank my friends Doris and JJ for all the chats and visits to Paris. Thanks for being with we, despite the physical distance.

In particular, I wish to give special thanks to my girlfriend Jiaqi for our magical encountering at the very end of my PhD journey. When coming to this important crossroad of my life, thank you for allowing me continue the following path with you. After years of oscillating between two lines, I now realize that they have become a circle with you at the center. *C'est comme ouvrir un rideau, rien n'a changé, mais tout est différent.*

Last but not least, I am deeply grateful to my parents for their love and support.

Contents

Chapter 1

Introduction

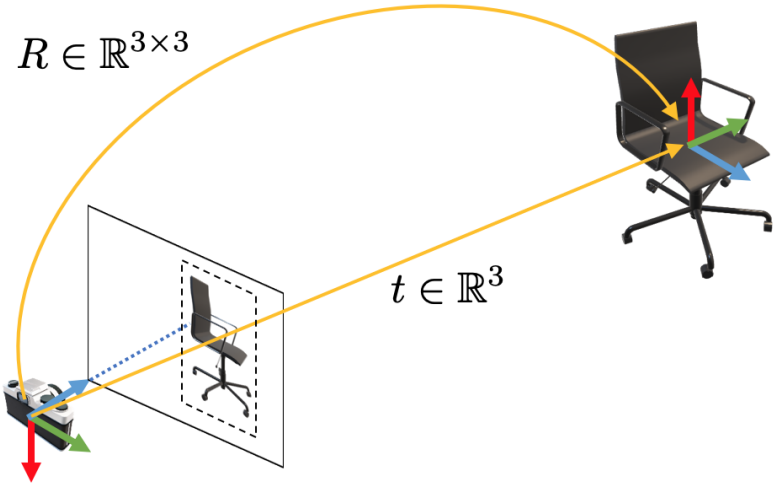


Figure 1.1: Illustration of 3D object pose estimation. Given an image picturing an object, the 3D pose consists of a 3×3 rotation matrix R and a 3-dimensional translation vector t. Viewpoint estimation methods estimate only the 3D rotation from the image crop centered on the object (shown by dashed bounding box).

1.1 Motivations

The goal of 3D pose estimation is to find the 3D rigid motion between the coordinate system attached to the object and the camera coordinate system. An illustration is shown in Figure 1.1. In this thesis, we focus on the estimation of the 3D rotation matrix in particular, which is alos known as the object viewpoint estimation.

3D object pose estimation is motivated by a wide range of industrial applications spanning from robotic manipulation, autonomous navigation, scene understanding to augmented reality.

Robotics. The estimation of 3D object pose can mainly benefit to two robotic applications: object manipulation and robot navigation (Figure 1.2). With many objects being placed in a cluttered scene, if a robot wants to grasp a target object, it must detect it and estimate its 3D pose before taking any action. Similarly, when a robot aims to navigate in an environment filled by various objects, it must estimate their 3D poses before interacting with them. For example, if a robot is commanded to go in front of the refrigerator and take something out of it, it must know the location and orientation of the refrigerator before planning the motion.

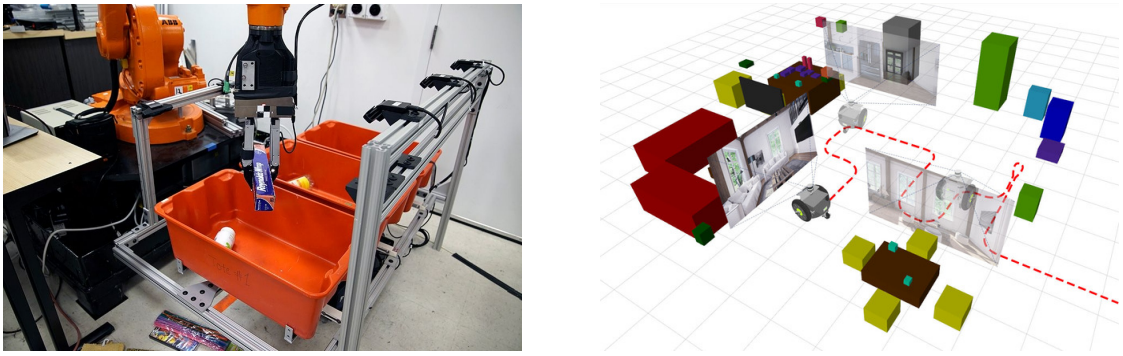


Figure 1.2: Robotics applications. **Left:** A robot arm manipulating an object [a]. **Right:** A robot vision system navigating in an environment [b].

[a] news.mit.edu/2018/robo-picker-grasps-and-packs-0220
[b] nikosuenderhauf.github.io/roboticvisionchallenges/scene-understanding

Figure 1.3: Augmented reality applications. **Left:** An application helping field technicians repair complex equipment with AR technology [a]. **Right:** An online shopping application letting people check what new products might look like in their homes before buying them [b].

[a] www.xerox.com/en-us/innovation/insights/augmented-reality-assistant
[b] www.youtube.com/watch?v=uhdOzpblrm0&t=49s

Augmented Reality. Augmented Reality (AR) allows us to insert a virtual object into 3D scenes seamlessly. As illustrated in Figure 1.3, we can, for instance, have a virtual assistant that gives us a step-by-step visual guide to accomplish any task, or have a powerful shopping tool letting us create full rooms of AR furniture when planning out what to buy. This is made possible by rendering and aligning a 3D model in the scene in a physically acceptable and visually plausible way. For this purpose, an estimation of the underlying geometry and the camera viewpoint is required.

Autonomous driving. When developing an autonomous driving system, it is important to make the system aware of the 3D poses of other objects in the environment,

Figure 1.4: Figure from (Kundu et al., 2018). Instance-level 3D object reconstruction for cars in real world scenarios.

especially for objects getting closer to the autonomous vehicle. As illustrated in Figure 1.4, when a car is moving forward in the road, the 3D poses of visible cars in front of it are estimated, together with the 3D models. This information informs the system about the moving objects around it and serves a vital role in motion planning and safety driving.

3D model retrieval. Retrieving 3D models for objects pictured in 2D images is extremely useful for 3D scene understanding. With the recent advances in creating large-scale databases of 3D models with rich categories, finding out which model we are actually interacting with in the scene becomes a meaningful task. As illustrated in Figure 1.5, having estimated the 3D pose of an object, the 3D model retrieval can be done in an efficient way by comparing the input image against a set of synthetic images rendered under the estimated 3D pose.

Keypoint estimation. Given an input image, solving the task of keypoint prediction from merely local appearance is difficult, sometimes even impossible due to ambiguities. For example, the "front-wheel" and "back-wheel" of a car could be very similar based on local patches. Knowing specifically the object's viewpoint can provide additional information such as which parts of the object should be visible and where the visible keypoints should be approximately located.

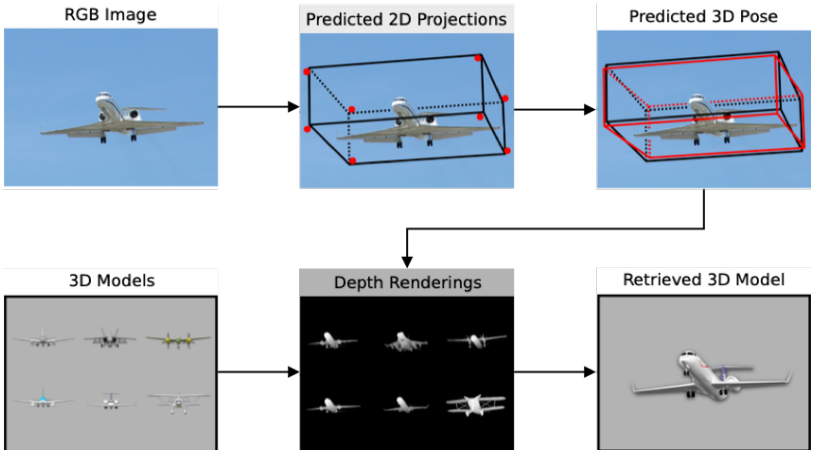


Figure 1.5: Figure from (Grabner et al., 2018). Knowing the 3D pose of an object can help retrieve the exact 3D model from the 2D image.

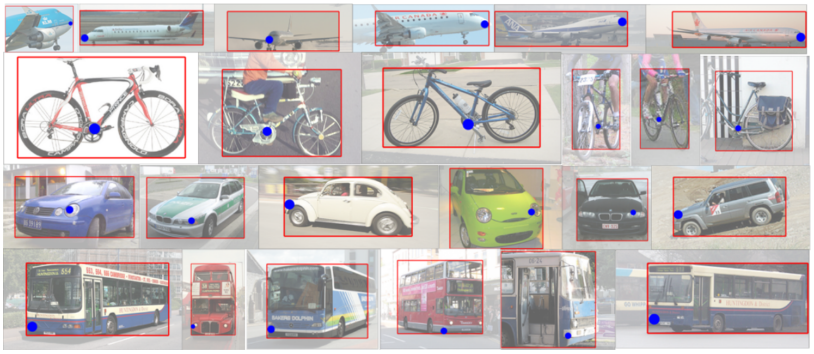


Figure 1.6: Figure from (Tulsiani and Malik, 2015). Viewpoint conditioned keypoint estimation for rich objects from different categories.

1.2 Challenges

When estimating the 3D pose of objects in the wild from RGB images, we must address several challenges. We detail the most critical challenges below:

Generalization to unseen objects. The scalability of object pose estimation methods is a determining factor for most industrial applications. Since learning based approaches requires training on images of specific objects with 3D pose labels, a severe constraint to these methods is that they are not able to generalize to new objects at test time. We summarize in Table 1.1 the statistics of several commonly-used datasets for object pose estimation. Given rich images captured in indoor scenes, 6 DoF object pose methods are able to estimate both translation and rotation of the labeled objects.

	Year	Categories	3D shapes	Images	Mask	Depth	Alignment
3 DoF object viewpoint datasets (Indoor and Outdoor)							
Pascal3D+	2014	12	79*	30,899	✗	✗	coarse
ObjectNet3D	2016	100	791*	90,127	✗	✗	coarse
Pix3D	2018	9	395	10,069	✓	✗	accurate
6 DoF object pose datasets (Indoor)							
LINEMOD	2012	–	15	18,273	✓	✓	accurate
T-LESS	2017	–	30	$\sim 49K$	✓	✓	accurate
YCB-Video	2018	–	21	133,827	✓	✓	accurate

Table 1.1: Statistics for commonly-used 3-Degree-of-Freedom (DoF) object viewpoint datasets and 6-DoF object pose datasets. *Only coarsely aligned and approximate 3D models are considered, the actual 3D shapes included in these datasets could be larger.

However, it is difficult to annotate translation for objects captured in the wild, e.g., an airplane in the sky. Besides, the estimation of translation also relies on the knowledge of camera intrinsics, which is sometimes hard to obtain. Due to these limitations, 6D object pose estimation datasets are usually limited to tens of objects captured in controlled environments. By contrast, object viewpoint estimation methods can work on various objects of different categories rather than a small number of specific instances, while estimating only the rotation and ignoring the translation.

Even though class-level object viewpoint estimation methods can generalize towards new objects belonging to the same categories, they are yet unable to work on unseen objects from novel categories. This remains an important limitation for the development of autonomous systems that should work with a large set of objects, regardless of being included in the training data or not. A possible solution is training keypoint-based approaches to estimate generic keypoints for all objects from different categories and obtain the 3D pose by solving a PnP-like problem. But these keypoint-based methods require well-defined keypoints on 3D models, and they are very sensitive to object appearances and large shapes variations.

3D pose annotations. In order to train a deep neural network for 3D object pose estimation, abundant images with 3D pose labels are usually required for covering a large space of pose distribution of various objects. As illustrated in Figure 1.7, annotating 3D poses of the objects in RGB images requires a great effort. First, the image-shape pairs need to be collected by either crawling web images and aligning them with CAD models from a repository of 3D CAD models, or scanning 3D objects

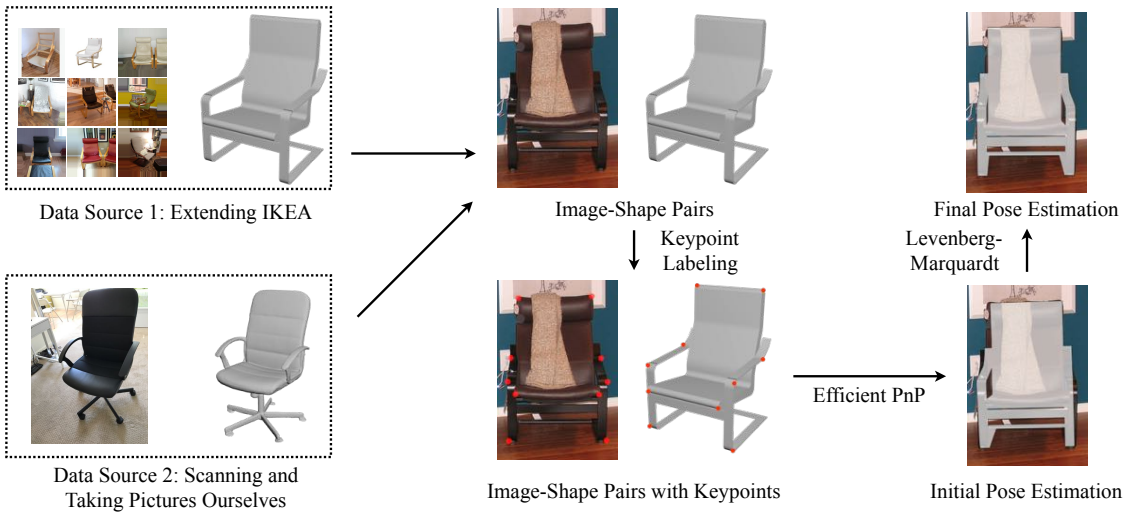


Figure 1.7: Figure from (Sun et al., 2018). Dataset construction pipeline used in Pix3D.

with specific sensors and taking pictures of them in different viewpoints and environments. Then, annotators are required to label a set of pre-defined keypoints on the images as well as their corresponding 3D coordinates on the CAD models, which is quite difficult when the images are of a low quality. Finally, the main camera parameter (focal length) and 3D poses (translation and rotation) can be obtained by minimizing the reprojection errors of the labeled keypoints. Regarding the potential occlusion and truncation in the real pictures, only visible keypoints are considered in the optimization. Thus, obtaining 3D pose estimations for objects pictured in the wild is a tedious process, and an object pose estimation method that generalizes beyond training data is highly desired for saving time and human labors. This aspect is implicitly linked to the previously mentioned challenge of generalization to unseen objects.

Suitable pose formalization and losses. Since 3D pose is a continuous quantity, a natural way to estimate it is to adopt a regression approach that directly predicts the 3D pose in a chosen representation of rotation matrices (Euler angles, axis-angles or quaternions). A disadvantage of the regression-based approaches is that they are unimodal in natural and fail to capture the multimodal distributions of the pose space for certain objects. An alternative approach is to discretize the 3D pose space into discrete bins and formalize the 3D pose estimation as a classification problem. While being better at handling the cases where multiple hypotheses are of

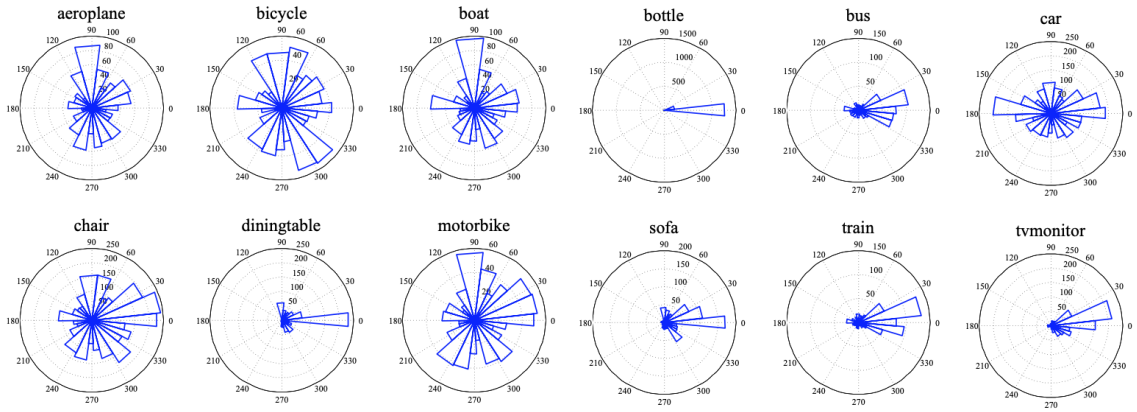


Figure 1.8: Figure from (Xiang et al., 2014). Polar histograms showing the azimuth distribution among images for the 12 object categories in Pascal3D+.

high probability due to ambiguity, the classification-based approaches requires a fine-grained discretization and geometry-aware loss functions for predicting accurate 3D pose. Dealing with the drawbacks of both types of approaches and combining their advantages together is a key challenge for accurate and robust object pose estimation.

Imbalances in training data. Based on the RGB images from large-scale datasets such as PascalVOC (Everingham et al., 2012) or ImageNet (Deng et al., 2009), object viewpoint datasets are built by annotating rich rigid object classes with 3D poses. Figure 1.8 shows the azimuth distribution among Pascal3D+ images for the 12 categories, where azimuth 0° corresponds to the frontal view of the object. The distribution is highly biased for certain categories such as "sofa" and "tvmonitor", where very few images are taken from the back view of the object. While not reflected in the histograms, there exists an even more severely biased distribution for the elevation and inplane rotation angles. Besides the imbalance of viewpoint distributions for most categories, the number of images can also vary a lot for different categories. As annotating objects in full 3D pose space is difficult and tedious, sometimes even impossible, how to train an object pose estimation method with only a few labeled images becomes a relevant research problem.

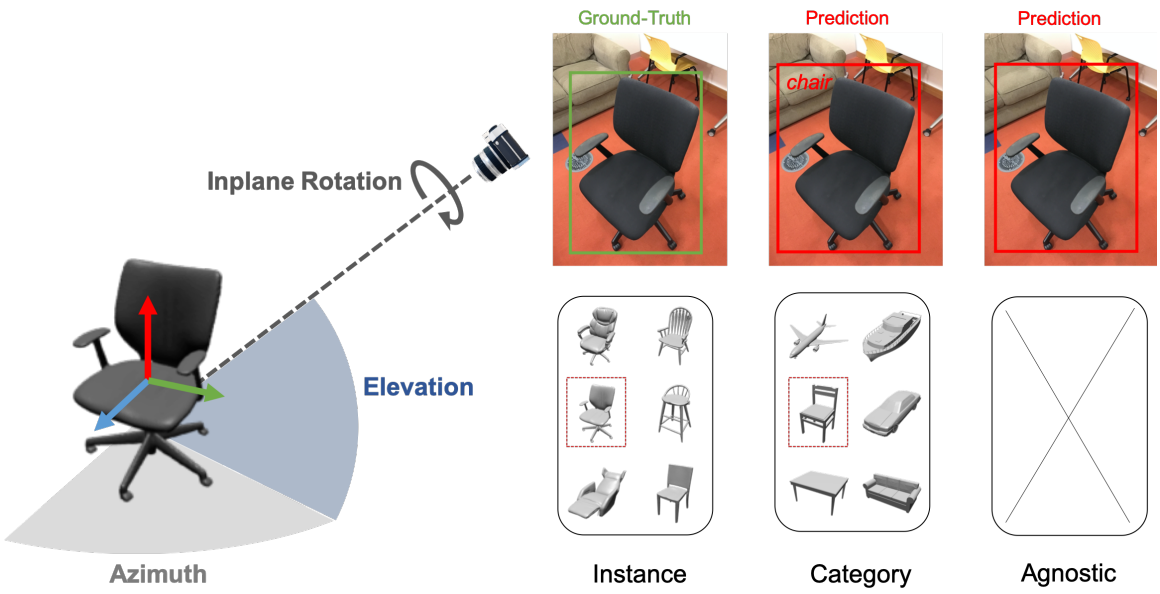
Objects in the wild. Object pose estimation methods developed on 6 DoF object pose datasets usually target only a small number of objects pictured in table-top setting, where the variety of object appearance and environment are both limited. Even though these methods can already satisfy some industrial applications in specific

scenarios, it could be an important limitation when applying the methods on rich objects pictured in the uncontrolled environments. For handling objects in the wild, viewpoint datasets built on rigid object classes with images in the wild have been proposed. These datasets contain images taken in various environments, including indoor and outdoor scenes, and feature various objects from different classes. Based on these datasets, many category-specific methods have been developed to estimate the 3D pose of a specific object class with an independent model or separate prediction branches. While being natural for the category-level evaluation, this category-specific design prevents the network to learn rich geometry similarities shared across different object classes and prevents the network to be applied on new object classes without changing the architecture.

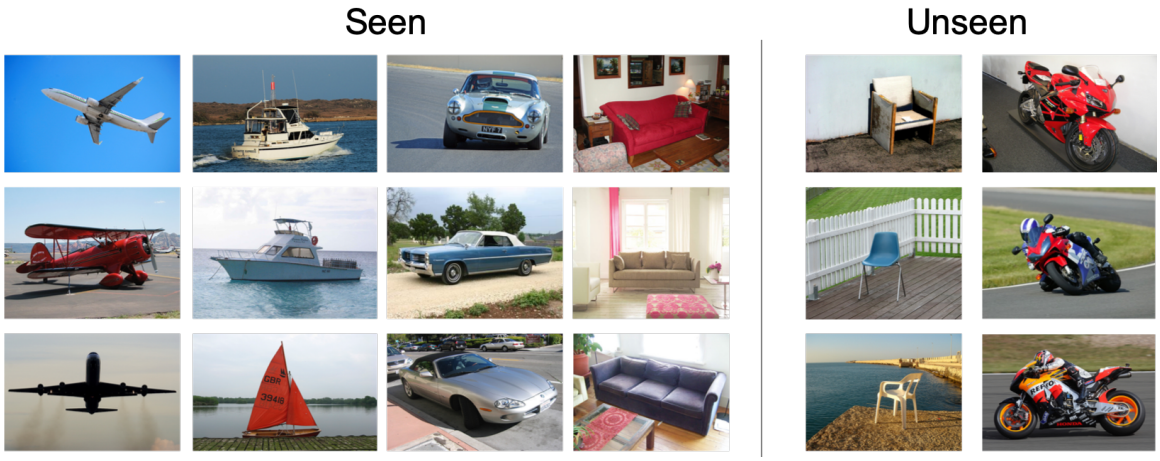
1.3 Goals

The goal of this thesis is to develop methods for estimating the 3D pose of an object pictured in an image. As we focus on estimating the 3D rotation of the object, the term "3D pose" and "viewpoint" will be used interchangeably in this thesis. We develop methods in different situations: (i) the object location in the image and the exact 3D model of the corresponding object are known, (ii) both the object location and the class are predicted and an exemplar 3D model is provided for each object class, and (iii) no 3D model is used and object location is predicted without the object being classified into a specific category. An illustration of the goal of this thesis is presented in Figure 1.9 (top).

Although existing methods based on hand-crafted features or deep neural networks have been used to successfully estimate the 3D pose of objects in many challenging situations, their application is restricted to the seen objects or classes that must be included in the network training procedure. This limits the development of generic artificial intelligent systems that have to interact with various objects in different environments, where novel objects different to the training objects could come in any time. Some template-based methods attempt to solve this problem by learning a feature extraction model to generate pose-equivariant image embeddings that are robust against other factors such as texture or light variations. On the other hand, keypoint-based approaches propose to define a set of class-agnostic keypoints on 3D models, e.g., the corners of 3D bounding box, and to learn a keypoint prediction



Task: given an input image, we aim to estimate the viewpoint (azimuth, elevation, inplane rotation) of the pictured object in different settings.



Setting: given a large-scale dataset of objects with viewpoint annotations, we want to learn a neural network that can generalize towards unseen objects.

Figure 1.9: **Goal.** We seek to solve the challenging task of viewpoint estimation from a single RGB image in different situations, such as (i) instance-level scenario where the exact 3D model of the query object is provided, (ii) category-level scenario where the class of the query object is detected, or (iii) agnostic scenario where the only input is a color image. Moreover, we want to develop an approach that can generalize towards unseen objects in the wild.

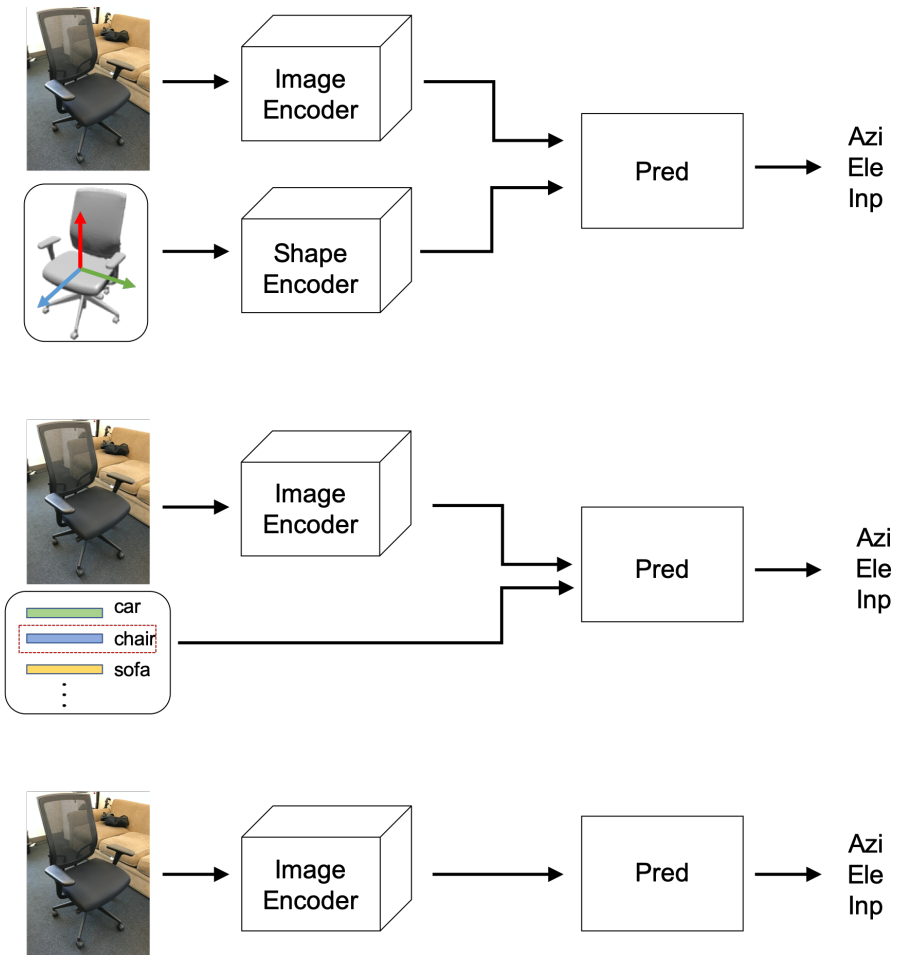
network that estimates the keypoint projections in RGB images for all objects. While

demonstrating some interesting results on novel objects, they are easily affected by the large variation in object appearances and environment illuminations. Moreover, their performances are largely lagging behind those of methods trained on the testing objects.

Object pose estimation with 3D models aims at inferring the 3D pose of an object from an image, given the exact 3D model of the corresponding object instance or a 3D exemplar model of the corresponding object class. In contrast with early approaches that focus on tens of objects or a small number of object categories, the goal of this thesis is to develop a deep learning solution for estimating the 3D pose of arbitrary objects in the wild, with a large variety in object appearances and environments (see Figure 1.9 (bottom)). In particular, we focus on the problem of finding a network architecture that is able to estimate the 3D pose of an object from an image given its instance-level or category-level shape information. This network should work for any object, seen or unseen, conditioned on its shape information.

In chapter 3, we introduce the first deep method that can estimate the pose of any object conditioned only on its exact 3D model, which can be represented by a set of rendering images picturing the object from different viewpoints or by a point cloud in the canonical frame. This method can be directly tested on unseen objects without retraining. Instead of relying on the exact 3D models, in chapter 4, we address the problem of category-level viewpoint estimation with exemplar 3D models. By leveraging the exemplar 3D models and extracting a pose-aware class representation for each class, our approach can not only work on base classes with rich labeled images, but also adapt quickly to novel classes with scarce labeled examples. Moreover, to go one step further into the full 3D scene understanding containing novel object classes, we propose a unified meta-learning framework for both few-shot object detection and few-shot viewpoint estimation. Our approach improves state-of-the-art performances on various few-shot object detection and viewpoint estimation benchmarks.

RGB-only class-agnostic object pose estimation aims at inferring the 3D pose of an object from RGB images only, without knowing its geometry or any semantic information. Previous methods rely on class-agnostic keypoints and estimate 2D-3D keypoint correspondences for computing the pose. Besides being indirect, these methods need a suitable design of keypoints on various objects and can be easily affected by occlusion and truncation. In chapter 5, we propose a direct method that



"Azi" – Azimuth; *"Ele" – Elevation;* *"Inp" – Inplane-Rotation*

Figure 1.10: Different testing scenarios tackled in this thesis: (top) the exact instance-level 3D model of query object is assumed to be known; (middle) only a set of pre-computed class-level features is provided; (bottom) no additional information is given, only the color image is used as input.

is trainable end-to-end, and we introduce a contrastive learning approach to learn a geometry-aware image embedding space that is optimized for object pose estimation.

1.4 Contributions

In this thesis, we make contributions in viewpoint estimation for unseen objects in the wild and develop 3D model based methods and contrastive learning based methods for object pose estimation. The key contributions of this thesis are the following:

1. Object pose estimation conditioned on shape instance. Our first contribution consists of a deep learning approach to category-free viewpoint estimation. This approach can estimate the pose of any object conditioned only on its 3D model, whether or not it is similar to the objects seen at training time. As illustrated at the top of Figure 1.10, the proposed network contains distinct modules for image feature extraction, shape feature extraction and pose prediction. These modules can have different variants for different representations of 3D models, but they all allow end-to-end training.

2. Pose-related data augmentation. In particular, for conditioning the pose prediction on both image and shape inputs, we propose a specific data augmentation technique that adds random rotations to the input shapes and modifies the orientation labels accordingly. An ablation study shows the superiority to the standard data augmentations which are only applied on the input images. We believe that the proposed data augmentation prevents the network to overfit the 3D model orientation when all models are aligned to the canonical frame. By changing the orientation labels according to the rotated shapes, the network is forced to learn to predict the pose of the objects with respect to the reference 3D model, which does not have to be consistent with a canonical frame.

3. Object pose estimation conditoned on shape exemplars. The previously presented approach requires the knowledge of a well-aligned 3D model for each query object and is sensitive to the precision of the provided 3D model of the pictured object. In order to mitigate the prerequisite of knowing the accurate 3D model for each object during testing, we develop a class-exemplar-based viewpoint estimation approach that learns to predict the viewpoint of an object based on a latent feature, which is extracted from the exemplar 3D models of the same class during training (middle of Figure 1.10). This approach differs from the previous approach in the sense that we extract a pose-aware class-wise feature for each class instead of treating each object independently, regardless of their classes. We show that the proposed approach can work with 3D models that differ from the actual objects pictured in the images, and it can be adapted quickly to novel classes with only a few labeled examples.

4. Few-shot object detection and viewpoint estimation. In addition, we define a simple, yet effective unifying framework that tackles both few-shot object detection and few-shot viewpoint estimation. We exploit, in a meta-learning setting, task-specific class information present in existing datasets for both tasks. More specifically, we leverage images with bounding boxes for object detection and exemplar 3D models of different classes for viewpoint estimation. Using this strategy, we address the joint problem of learning to detect novel objects in images and to estimate their viewpoints from only a few labeled examples. Moreover, we propose a full evaluation scheme for the joint task of few-shot object detection and few-shot viewpoint estimation for objects in the wild.

5. Pose-aware contrastive learning. Finally, we develop a class-agnostic object viewpoint estimation approach that estimates the viewpoint directly from an image embedding, which is optimized for viewpoint estimation through a geometry-aware contrastive learning. Instead of blindly pulling together features of the same object in different augmented views and pushing apart features of different objects while ignoring the pose difference between them, we propose a pose-aware contrastive loss that pushes away the image features of objects having different poses, ignoring the class of these objects. By sharing the network weights across all categories during training, we obtain a class-agnostic viewpoint estimation network that can work on objects of any category (bottom of Figure 1.10). This allows us to test the network directly on unseen objects, without the need to know their 3D shapes or classes.

1.5 Thesis Outline

This thesis is organized as follows:

Chapter 2: Related Work. We start by providing an overview of the related works in the literature with a specific focus on 3D object pose estimation. In particular, we first review model-based matching methods and keypoin-based methods, then deep-learning-based direct estimation methods that are most related to this thesis.

Chapter 3: Pose from Shape. This chapter introduces the first three contributions of this thesis: our new object pose estimation approach conditioned on instance shape and a pose related data augmentation. We first explain how pose estimation

can be conditioned on 3D shapes. We then present our data augmentation, used for adding a specific link between the object pose estimation and the reference frame of the 3D shape. We empirically compare our direct estimation approach with other keypoint-based methods for the task of viewpoint estimation on various benchmarks. In particular, we show that our shape-conditioned method can generalize better to unseen categories that are not included in training. We also provide results showing its further applications to generic objects, including unseen object classes of Pix3D (Sun et al., 2018), texture-less objects of LINEMOD (Hinterstoißer et al., 2012b), and horse images of ImageNet (Deng et al., 2009).

Chapter 4: FSDetView. We develop a simple, yet effective framework to unify the task of few-shot object detection and few-shot viewpoint estimation. We show how to exploit task-specific class information present in different data structures and how to leverage them for object detection and viewpoint estimation. We experimentally compare against other few-shot object detection methods as well as few-shot viewpoint estimation methods, and show that our approach improves on both tasks on various benchmarks. Moreover, instead of treating the two tasks separately, we also propose a joint evaluation where the viewpoint estimation is based on the object detection results and provide promising results on ObjectNet3D (Xiang et al., 2016) and Pascal3D+ (Xiang et al., 2014).

Chapter 5: PoseContrast. We present a pose-aware contrastive learning approach to perform viewpoint estimation in a class-agnostic way, in the absence of 3D shapes. In order to learn an image embedding that is optimized for viewpoint estimation, we first separate the application of pose-variant and pose-invariant data augmentations and apply them correctly in different steps of the training data preparation. We then introduce a pose-aware contrastive loss that enhances the contrast between images with similar poses and images with dissimilar poses. We experimentally compare against other class-agnostic viewpoint estimation approaches and show state-of-the-art results on Pascal3D+. We also show that our approach can be combined with a class-agnostic object detection model for evaluating on objects from any category, and show comparable results with state-of-the-art methods on no-shot and few-shot viewpoint estimation.

Chapter 6: Conclusion. We conclude with a summary of contributions of this

thesis and suggestions on future directions.

1.6 Publication List

Three papers are presented in the manuscript.

- **Yang Xiao**, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet. Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects. In *British Machine Vision Conference (BMVC)*, 2019.
- **Yang Xiao** and Renaud Marlet. Few-Shot Object Detection and Viewpoint Estimation for Objects in the Wild. In *European Conference on Computer Vision (ECCV)*, 2020.
- **Yang Xiao**, Yuming Du, and Renaud Marlet. PoseContrast: Class-Agnostic Object Viewpoint Estimation in the Wild with Pose-Aware Contrastive Learning. In *International Conference on 3D Vision (3DV)*, 2021.

We open-sourced the code corresponding to the three papers [1], and created webpages [2] for each project. I also maintain a GitHub repository [3] tracking and summarizing the recent advances in 3D object pose estimation. The various codes on GitHub received hundreds stars and forks.

During my PhD, I also took apart in three other projects which are not discussed in this manuscript:

- Shell Xu Hu, Pablo Moreno, **Yang Xiao**, Xi Shen, Guillaume Obozinski, Neil Lawrence, and Andreas Damianou. Empirical Bayes Transductive Meta-Learning with Synthetic Gradients. In *International Conference on Learning Representations (ICLR)*, 2020.
- Xuchong Qiu, **Yang Xiao**, Chaohui Wang, and Renaud Marlet. Pixel-Pair Occlusion Relationship Map (P2ORM): Formulation, Inference & Application. In *European Conference on Computer Vision (ECCV)*, 2020.
- Yuming Du, **Yang Xiao**, Vincent Lepetit. Learning to Better Segment Objects from Unseen Classes with Unlabeled Videos. In *International Conference on Computer Vision (ICCV)*, 2021.

[1] https://github.com/YoungXIAO13
[2] https://youngxiao13.github.io
[3] https://github.com/YoungXIAO13/ObjectPoseEstimationSummary

- Xi Shen, **Yang Xiao**, Shell Xu Hu, Othman Sbai, Mathieu Aubry. Re-ranking for image retrieval and transductive few-shot classification. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Chapter 2

Literature Review

In this chapter, we present an overview of 3D pose estimation methods of rigid objects. 3D pose estimation has a long history in computer vision with a large variety of proposed methods.

We begin by model based matching methods that retrieve the 3D object pose by comparing the observed image against a set of templates generated under known poses, then we discuss keypoint based methods relying on 2D-3D correspondences and PnP algorithm for pose estimation. Finally, we review more recent trainable methods for predicting directly the 3D pose from image embeddings, most related to our work.

Note that we simply give an overview of these different types of approaches, the reader can refer to (Sahin et al., 2019, 2020; Lepetit, 2020) for a more detailed survey.

Assumptions and clarifications. Object viewpoint refers to the three Euler angles (azimuth, elevation and inplane rotation) as shown in Figure 1.9, by contrast, the 3D object pose could refer to a 3D transformation that has 6 degrees of freedom: 3 for the 3D translation and 3 for the 3D rotation. The term *"6-DoF pose"* is thus used in some literature to represent the full 3D pose that has 6 degrees of freedom. In this thesis, our goal is to estimate the object viewpoint, without considering the 3D translation. Besides, viewpoint estimation takes the image crop centered at the target object as input, which can be provided manually by an annotator or automatically by an object detection algorithm.

2.1 Model Based Matching

In this section, we discuss model based matching methods that retrieve the 3D object pose by comparing the template images against the query object image. The templates can be represented by real images or synthetic images rendered from CAD object models. For comparing different images, the similarities can be computed either locally for different image patches, or globally through an image embedding module that calculates a descriptor for each image.

We start by reviewing matching methods based on local features such as image contours (Gavrila and Philomin, 1999; Holzer et al., 2009) and image gradients (Hinterstoißer et al., 2010, 2012a). We then discuss methods relying on both RGB and depth images for obtaining robust features based on multi-modal inputs. Finally, we discuss methods that compute global descriptors by learning with triplet

(a) (b) (c) (d)

Figure 2.1: Figure from (Holzer et al., 2009). Given an input image (a), the corresponding edge image is obtained by applying a Canny detector (b), the distance transformation is computed from the edge image (c) and a set of templates are extracted for comparison (d).

loss (Wohlhart and Lepetit, 2015), domain adaptation loss (Massa et al., 2016b), or image reconstruction loss (Sundermeyer et al., 2018).

By contrast to the model based matching methods that retrieve the most similar template for the query object image, our approach in Chapter 3 leverage the 3D models in an implicit way by extracting deep shape features from the 3D models. These shape features are combined together with the image features extracted from the input images for getting the final viewpoint estimation. We show in Chapter 3 that our approach is robust against the domain gaps between synthetic images and real images, moreover, it can work easily with different representations of 3D models.

2.1.1 Local feature matching

A natural way to compute the similarity between an observed image and a template image is to compare local features extracted from different image locations and then aggregate them together to get a global score.

Matching with image contours. A shape-based object detection method based is introduced in (Gavrila and Philomin, 1999). Based on the edges extracted from two images, a distance between them is computed for measuring how one image is matched to the other. They use Chamfer distance that computes the average distance to the nearest point. A closely related image matching method is proposed in (Huttenlocher et al., 1993), where the Hausdorff distance is considered. Unlike Chamfer distance where an average distance is computed, Hausdorff distance takes the average truncated distance or some quantile value of the distances. This renders the Hausdorff distance more robust against outliers caused by occlusion or detection errors. To get a more robust distance measurement, (Holzer et al., 2009) propose an

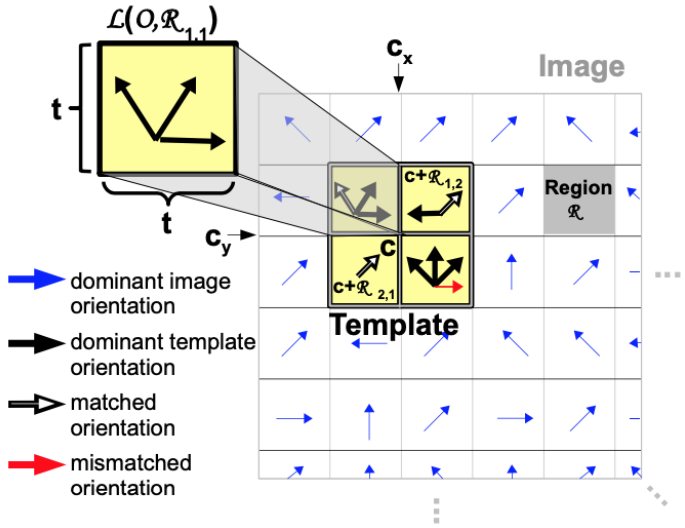


Figure 2.2: Figure from (Hinterstoißer et al., 2010). Only the local dominant orientations are considered for each image region.

approach based on normalized Distance Transform Contour Templates, as illustrated in Figure 2.1. They use a training phase to learn the poses, identities and the spatial relations of the Distance Transform Templates. The normalized templates are obtained by transforming the image contour to a canonical scale and orientation.

Matching with image gradients. (Steger, 2002) proposes a method by considering the image gradients in contrast to the image contours. A directional vector is computed for each image location without image segmentation. The similarity between the input image and the template is measured as the sum of dot products between the template gradients and the image gradients. This results in robustness to illumination changes.

To further improve the robustness of matching with image gradients, (Hinterstoißer et al., 2010) propose a novel template representation called DOT (Dominant Orientation Templates) that is robust to small image transformations. The representation is related to the HoG (Histograms-of-Gradients) representation (Dalal and Triggs, 2005). Instead of simply computing the histogram for each image patch, DOT relies on locally dominant orientations, and is made explicitly invariant to small translations (Figure 2.2). Moreover, by binarizing the template representation and using an efficient exploration technique, DOT is faster than HoG by orders of magnitude. Rather than just focus on the dominant gradient orientations, an extended version of

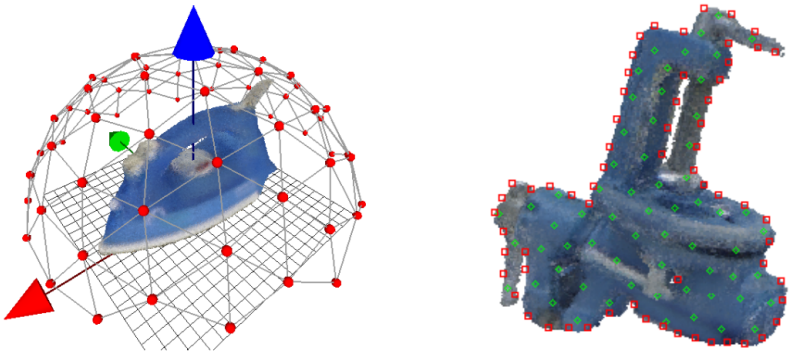


Figure 2.3: Figure from (Hinterstoißer et al., 2012b). For each 3D model, a set of uniformly sampled virtual cameras are placed on the upper hemisphere for template generation. The selected features contain both color image gradients (red) and surface normal (green).

DOT is proposed in (Hinterstoißer et al., 2012a) to consider all gradient orientations in local image neighborhoods.

Matching in multi-modalities. To simultaneously leverage the information of multiple modalities, (Hinterstoißer et al., 2011) focus on the combination of a color image and a dense depth map for constructing robust templates. More specifically, image gradients are extracted from the object contour for color image integration, while 3D surface normal vectors are extracted on the body of the object for depth integration. Features extracted from both modalities are quantized, and the "linearized response maps" is used in (Hinterstoißer et al., 2012a) for heavy parallelization. This template generated from multiple modalities are demonstrated to reduces greatly the false positive rate in spite of heavy clutter.

While templates proposed in (Hinterstoißer et al., 2011) are robust through the addition of depth maps, they are learned online from a set of RGB-D images taken around the object. This requires a human or a robot to interact with the environment in a careful manner such that the collected images cover the whole pose range. To bypass this tedious sample collection step, (Hinterstoißer et al., 2012b) propose to automatically generate templates from the 3D object models, which can be quickly created. Based on a 3D model, a sparse set of viewpoints regularly distributed on the full view hemisphere are generated by recursively dividing an icosahedron, as illustrated in Figure 2.3. In addition to these two out of plane rotations, templates are also created for different in-plane rotations. Moreover, (Hinterstoißer et al., 2012b) show that only a subset of the features used in (Hinterstoißer et al., 2011) need to

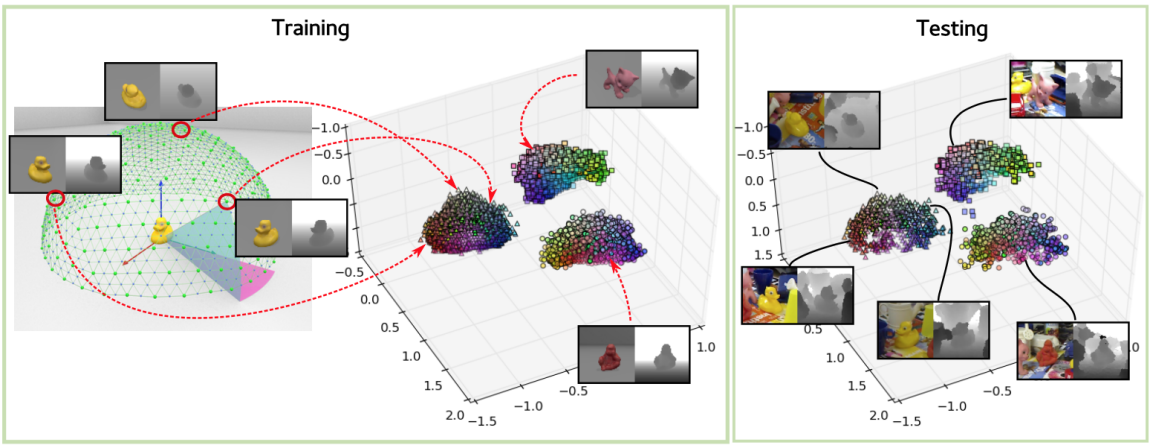


Figure 2.4: Figure from (Wohlhart and Lepetit, 2015). Descriptors of different objects are well separated and the descriptors of the same object in different views capture the geometry of the corresponding poses.

be considered for speeding up the detection without losing accuracy. (Rios-Cabrera and Tuytelaars, 2013) further extend this approach by training a linear SVM to learn better templates and tuning each of the templates separately.

A deep learning based local RGB-D patch descriptor is proposed in (Kehl et al., 2016). In particular, a convolutional autoencoder is trained to regressing the descriptive features from local RGB-D patches on a large collection of random local patches, where an image reconstruction error is minimized. Once trained, the deep model is used to create codebooks from synthetic patches sampled from different object views, where each codebook entry holds a local 6D pose vote. During testing, descriptors regressed from local input image patches are matched against the codebooks to get a number of candidate votes for pose estimation.

2.1.2 Global descriptor matching

While local descriptors extracted from RGB or RGB-D images have been demonstrated to work well with textureless objects, computing descriptors for each image location is yet expensive for real-time application. Besides, matching across numerous local patches efficiently requires a carefully designed pipeline. To mitigate these disadvantages, more recent works propose to compute global descriptors capturing holistic object representations in different views, and the matching is done via scalable nearest neighbor search methods for large number of objects.

Triplet loss based descriptor learning. (Wohlhart and Lepetit, 2015) propose to compute descriptors for object views that efficiently capture both the object identity and 3D pose. For this purpose, a compact and discriminative description vector is extracted for each object view using a convolutional neural network, which is trained with a triplet loss such that the Euclidean distance between descriptors from two different objects is large and that between the descriptors from the same object is representative of the similarity between their poses, as illustrated in Figure 2.4. This specific loss makes the descriptors of different objects lie in separate clusters in the embedding space, and the descriptors of the same object change with the viewpoint accordingly. Furthermore, a pair-wise loss is also used to make the descriptors robust to noises and other distracting artifacts such as changing illumination. During testing, we can recognize the object and estimate its pose by matching its descriptor against a database of pre-computed descriptors.

(Balntas et al., 2017) further improve the triple loss and the pair-wise loss of (Wohlhart and Lepetit, 2015) by exploring the direct usage of pose labels in the feature learning process. A pair-wise pose loss is used to enforce a direct relation between the pose differences and the embedding distances, while a triple object loss is used to enforce embeddings of the same object in different poses to have smaller distances compared to embeddings from different objects, irrespective of the pose differences. (Balntas et al., 2017) exploit the combination of the feature learning loss with a direct pose regression loss, which is shown to further boost the pose estimation accuracy. Another extension is proposed in (Zakharov et al., 2017), where a dynamic margin is added to the triplet loss for faster training without losing accuracy. Furthermore, (Zakharov et al., 2017) add the inplane rotations into consideration and improve the robustness by using surface normals.

Domain adaptation for 2D-3D alignment. To bridge appearance gaps between real images and synthetic views rendered from CAD models, (Massa et al., 2016b) introduce an adaptation approach to adapt natural image features for the task of 2D-3D exemplar detection. Based on a large training set of well-aligned pairs of composited images and rendered views of CAD models, both inputs are first transformed into gray-scale images and then projected into a high dimensional feature space via a pre-trained deep network. Then, the real image features pass through an adaptation layer thus the transformed real features are similar to the features extracted from the corresponding rendered views. As illustrated in Figure 2.5, this learnt feature

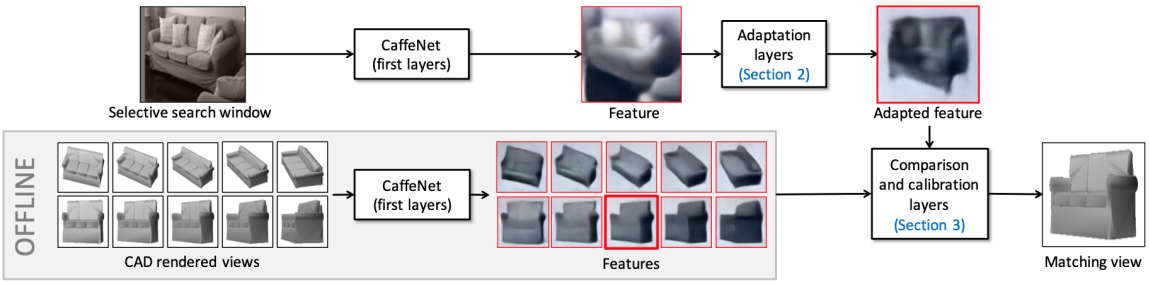


Figure 2.5: Figure from (Massa et al., 2016b). For cross-domain image matching, a feature adaptation layer is used to transform the feature space of real images to the feature space of CAD rendered views.

adaption layer can be easily integrated into existing feature extraction modules for aligning objects in real images to rendered views that match the style and pose of depicted objects. Note that all features are extracted from color images without the use of depth information, which facilitates the application in real world scenarios.

Autoencoder based template matching. Similar to (Kehl et al., 2016), (Sundermeyer et al., 2018) also leverages an Autoencoder for learning discriminative image descriptors that can be used for object pose estimation. However, instead of relying on RGB-D patches, (Sundermeyer et al., 2018) take the inspiration from the Denoising Autoencoder (Vincent et al., 2010) and combine it with a novel domain randomization strategy to learn implicit representation of objects from RGB images only. As shown on the top of Figure 2.6, clean rendered views are augmented with dramatic color transformation before passing through the Autoencoder, which is trained to reconstruct the clean rendered views in the same pose as the input images. This enables the learned image representations to specifically encode 3D orientations while achieving robustness against occlusion and cluttered backgrounds. Moreover, the training is performed in a self-supervised manner without the need of a large dataset with pose annotations. During test time, a codebook is generated for each object and the real image embeddings are matched against the codebook for retrieving the correct object pose, as illustrated at the bottom of Figure 2.6.

Furthermore, (Sundermeyer et al., 2020) extends this approach by training a single encoder for multiple objects with an independent decoder per object. Using this strategy, different object views share common features in the latent space without being separated. The trained encoder has been demonstrated to generalize well from synthetic images to real images, and from seen objects to unseen objects.

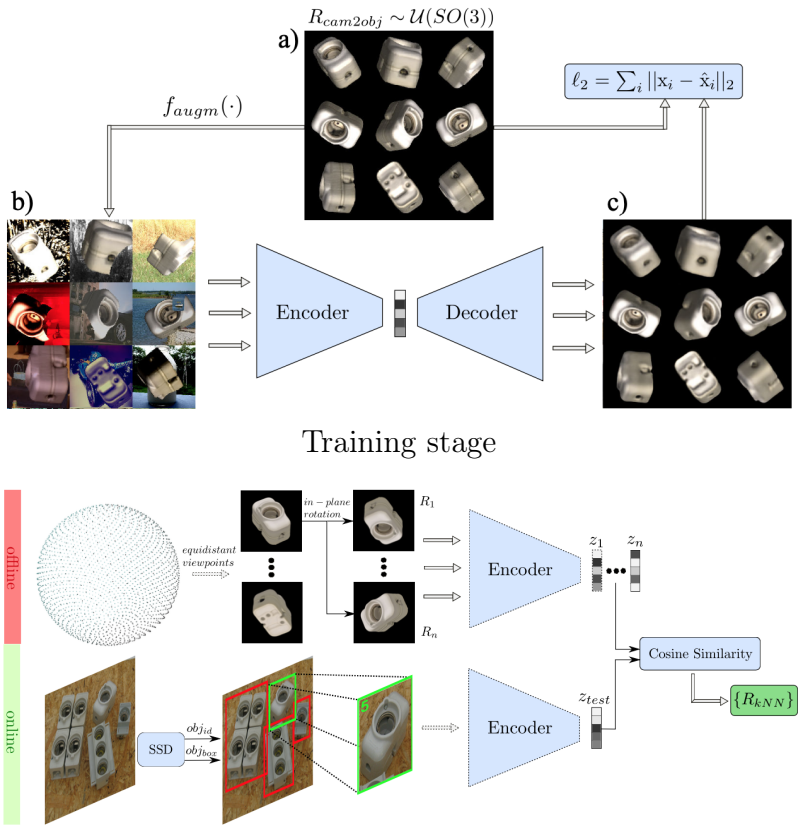


Figure 2.6: Figure from (Sundermeyer et al., 2018). By training the Autoencoder with a reconstruction loss, image encodings of clean rendered views are mapped closer with image encodings of randomly augmented images in the same pose. The learned encoder is then used to create a codebook from the encodings of discrete synthetic object views, and pose estimation is done by matching the input image encoding against the codebook.

2.2 2D-3D Correspondences

Instead of comparing observed images against a set of template images, another way to estimate the 3D object pose is to define a set of 3D points on the object model and find the corresponding 2D projection locations on the image. Based on these 2D-3D correspondences, the object pose can be retrieved by solving a PnP algorithm. We start by discussing methods relying on sparse correspondences such 3D bounding boxes or selective surface points. We then discuss methods computing dense correspondences at pixel-level. Finally, we review some particular methods aiming to estimate the generic correspondences that can be applied directly on unseen objects.

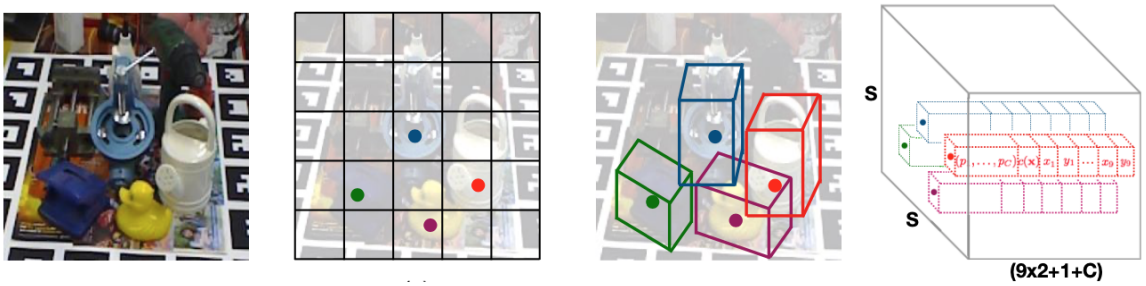


Figure 2.7: Figure from (Tekin et al., 2018). Given an input image, a grid of regular cells are generated. Each cell is responsible for predicting the 2D locations of the eight 3D bounding box corners and the object centroid in the image, together with a confidence score and the class probabilities.

While not being directly related to this group of methods, we compare with the most recent ones attempting to predict the keypoints on unseen objects, and show superior object pose estimation performances on various benchmarks.

2.2.1 Sparse correspondences

Direct regression of 3D bounding box corners. The idea of using deep learning to predict the 2D-3D correspondences for 3D object pose estimation is first introduced in BB8 (Rad and Lepetit, 2017), where a two-stage approach is proposed. In the first stage, objects in the image are segmented in a coarse-to-fine manner and the centroid of the final segmentation is used as the 2D object center. In the second stage, an image window is centered on the 2D object center and another deep network is trained to predict the 2D reprojections of the eight corners of the 3D object's bounding box. Based on these 2D-3D correspondences, the object pose is obtained with a PnP algorithm. Moreover, a pose refinement network is also introduced in BB8 to improve the pose estimation accuracy by rendering the object with the estimated pose and comparing it with the input image. The pose estimation is refined thus that the visual difference between the rendering and the input image is minimized.

In order to accelerate the whole estimation pipeline used in BB8, (Tekin et al., 2018) extend the single-stage object detection network YOLO (Redmon et al., 2016) to a single-stage 6D object pose estimation network. This network simultaneously segments the objects and estimates the 2D reprojections of the 3D bounding box corners, together with the object centroid in input images (Figure 2.7). More recently, (Hu et al., 2020b) propose an end-to-end trainable object pose estimation method

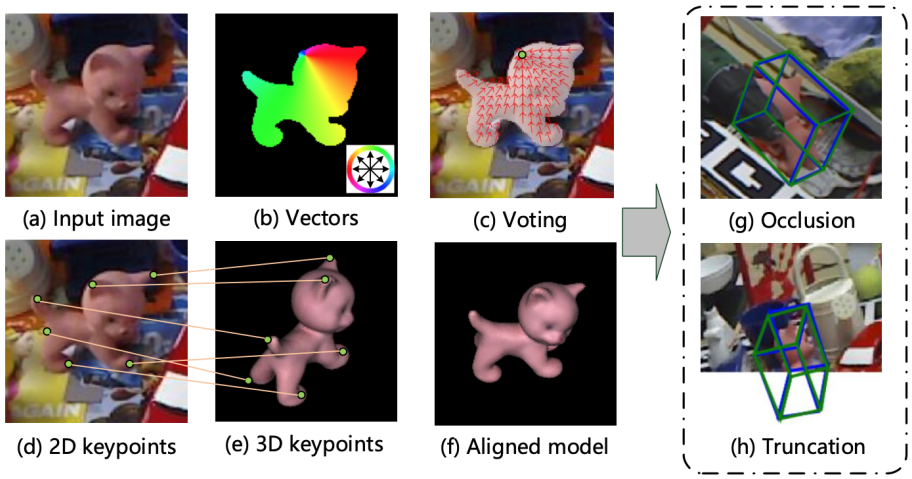


Figure 2.8: Figure from (Peng et al., 2019). Network is trained to predict an unit vector pointing to keypoints for each pixel, and the localization is done in a RANSAC based voting scheme.

by regressing directly the 3D pose from clusters of 2D reprojections in the image. This removes the need for first establishing 2D-3D correspondences through a neural network and then retrieving the final 3D pose via a non-differentiable PnP algorithm. Besides, this end-to-end estimation pipeline avoids the use of a surrogate loss that does not directly reflect the final pose estimation task.

Vector representations with voting scheme. To handle the large partial occlusions, (Hu et al., 2019) claim that treating the object as a global entity makes the estimation vulnerable, and thus propose a two-streams architecture with a segmentation stream for detecting each object in the image, and a regression stream for predicting the 2D locations of the eight corners of the 3D bounding box. Instead of directly predicting the 2D image coordinates, they predict the 2D displacements from the grid center to the 8 corner projection locations in the image plane, together with a confidence score for each displacement. A robust set of 2D-3D correspondences is then constructed by taking the most confident predictions from each visible part, and the final pose estimation is obtained with a RANSAC-based PnP algorithm.

(Peng et al., 2019) argue that addressing occlusions or truncations requires dense predictions for the 2D-3D correspondences. They thus propose a novel framework for 6D pose estimation using a Pixel-wise Voting Network (PVNet). As illustrated in Figure 2.8, PVNet predicts unit vectors that represent directions from each pixel

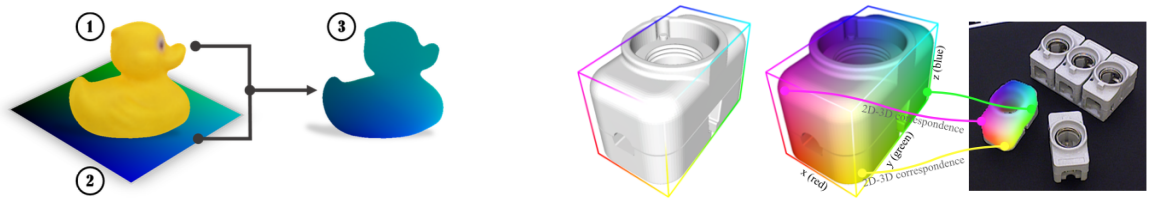


Figure 2.9: (Left) Figure from (Zakharov et al., 2019), a UV-mapping is used to get the 2-channel correspondence map with values in 0 to 255. (Right) Figure from (Park et al., 2019), normalized coordinates of each 3D vertex are directly mapped to RGB values in the color space.

of the object towards the keypoint, while the background pixels are masked out. Based on this vector-field representation, even keypoints of an invisible part can be inferred through a voting scheme based on RANSAC. In addition, they also argue that 3D bounding box corners outside the object pixels should be replaced by keypoints selected explicitly on the object surface for reducing the localization errors. In particular, the farthest point sampling (FPS) algorithm is used to select a set of keypoints for each object, and better results have been achieved in their experiments.

2.2.2 Dense correspondences

The idea of using dense 3D object coordinates for object pose estimation is first introduced in (Brachmann et al., 2014). Based on RGB-D images, they jointly predict a dense 3D object coordinate labelling and a dense class labelling for 6D object pose estimation of specific objects. (Brachmann et al., 2016) has latter extended this approach to estimate the 6D pose from single RGB image by developing a regularized regression framework which iteratively reduces uncertainty in predictions. More recently, some deep learning based methods have been proposed based to predict dense 2D-3D correspondences from a single RGB image and leverage a RANSAC based PnP algorithm for more robust and accurate pose estimations.

As illustrated in Figure 2.9 (left), DPOD (Dense Pose Object Detector) (Zakharov et al., 2019) use a UV-mapping to represent the 3D object coordinates in a textured 2-channel images with values ranging from 0 to 255. In this way, a bijective mapping between the model vertices and image pixels on the correspondence map is obtained. By predicting an identity mask and a UV map for each object, high quality 2D-3D matches are generated and used in a RANSAC based PnP algorithm for getting the final pose prediction. Similar to DPOD, Pix2Pose (Park et al., 2019) also regress dense

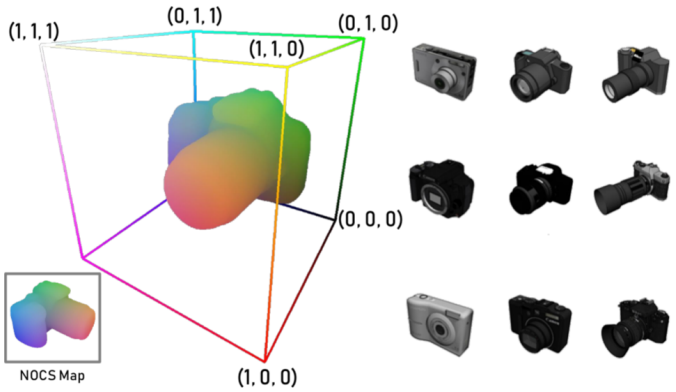


Figure 2.10: Figure from (Wang et al., 2019a). All objects from the same category are aligned and normalized into an unit cube, where 3D locations are transformed into RGB values.

2D-3D correspondences. However, instead of using a UV-mapping, they obtain the correspondence map by representing directly the normalized 3D object coordinates, encoded by the corresponding RGB values, as shown in Figure 2.9 (right). They propose an autoencoder architecture to estimate the 3D coordinates as well as the expected errors for each pixel, from which a RANSAC based PnP is used to get the final pose estimation.

(Li et al., 2019) propose CDPN (Coordinates-based Disentangled Pose Network) which also relies on dense 2D-3D correspondences for object pose estimation. Unlike DPOD and Pix2Pose that predict the whole 6D pose from these dense correspondences, they argue that the 3D rotation and the 3D translation should be treated differently by disentangling their predictions. More specifically, they estimate the 3D rotation by using predicted 2D-3D matches and a PnP algorithm in a similar way to DPOD and Pix2Pose, while they regress 3D translation from local image patches via a dynamic zoom in and a scale-invariant estimation.

2.2.3 Correspondences on unseen objects

Given the 3D model of an object, the 2D-3D correspondences can be defined on a sparse or a dense set of keypoints selected on the 3D model. However, how to predict these correspondences for unseen objects remains an under-explored problem. In this section, we review some recent keypoint based methods tackling this problem under the challenge of intra-class variation or inter-class variation.

Figure 2.11: Figure from (Zhou et al., 2018). Class-agnostic keypoints for different object classes. The number of keypoints vary for different objects.

Intra-class correspondences. In order to estimate the 3D pose of an unseen object from the same category, (Wang et al., 2019a) propose a shared canonical representation for all object instances within a category, namely NOCS (Normalized Object Coordinate Space). As illustrated in Figure 2.10, all 3D models are aligned for a given object category, and each of them is normalized in an unit cube such that the diagonal of its tight bounding box has a length of 1. Similar to Pix2Pose, each 3D location in NOCS can thus be represented by a tuple of RGB values in the color space. By training a deep network to predict the NOCS for all objects within a category, it can be applied on unseen objects from the same category during testing. And the full 6D object pose can be calculated by fitting the NOCS predictions to the depth map.

Inter-class correspondences. To predict 2D-3D correspondences for unseen object classes with unknown geometries, (Zhou et al., 2018) propose a category-agnostic keypoint representation called StarMap. Instead of defining a fixed number of keypoints for specific object classes, they use a single channel multi-peak heatmap for all keypoints of all objects, regardless of their classes. Given an input RGB image, they predict this single-channel heatmap for getting a varying number of keypoints (Figure 2.11). In addition, they also predict the 3D locations in the normalized canonical view for each keypoint, together with a depth map. The object pose can then be computed from these arbitrary 2D-3D correspondences by solving a PnP algorithm. Their intuition is that both intra-category part variations and inter-category part similarities can be captured automatically by training a class-agnostic keypoint estimation network across all classes. To the best of knowledge, they are the first to

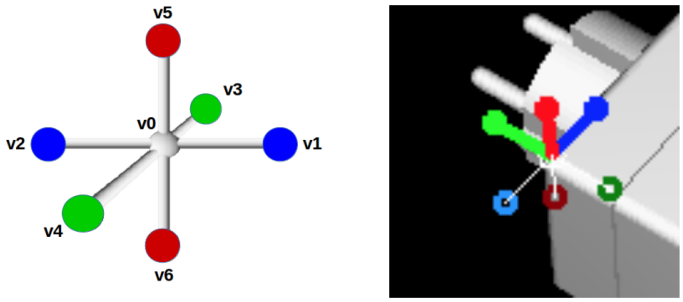


Figure 2.12: Figure from (Pitteri et al., 2019a). A set of virtual control points are selected on the corners of the object.

conduct pose estimation on unseen classes and report results for objects in the wild.

Generalization with local geometries. (Pitteri et al., 2019a) propose an approach to estimate the 3D pose of new target unseen objects whose CAD models can exist but no training image is required. In particular, they design a deep learning network for detecting the corners and determining their 3D poses through a set of predefined virtual control points, as shown in Figure 2.12. By focusing on the industrial objects that are often made of similar parts with dominant corners, the trained network can generalize naturally to unseen objects, without retraining.

(Pitteri et al., 2020) explore further into this direction by looking at the discriminative local geometries rather than only focusing at the dominant corners. By establishing dense correspondences between the image locations and the 3D object points, they show accurate pose estimation can be achieved on unseen objects without any specific selection of 3D points.

2.3 Direct Estimation

Direct pose estimation methods use large-scale dataset and learn to estimate the object pose directly from the image embeddings. We start this section by describing classification-based methods that typically divide the pose space into discrete bins and predict a classification score for each bin. We then discuss regression-based methods that predict the object pose in continuous space under different pose representations. Finally, we present mixed classification-and-regression methods that combines the advantages of both classification and regression methods by first classifying among coarse discrete bins and then regressing offset values within the selected bins.

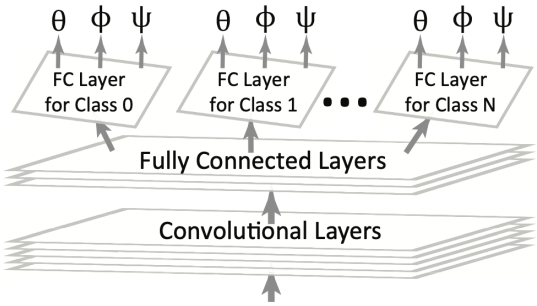


Figure 2.13: Figure from (Su et al., 2015b). Class-specific viewpoint estimation network architecture, an independent branch is used to predict the three Euler angles for each object class, and each Euler angle is predicted as the angle bin center having largest classification score.

2.3.1 Classification

Direct classification. To determine the three Euler angles from a single image, (Tulsiani and Malik, 2015) propose to formulate this task as a classification problem by discretizing each angle into disjoint angular bins. The predicted angle value is computed as the center of the angle bin having the largest classification score. By training a single deep network with separate prediction branches for different object categories, they achieve to conduct viewpoint estimation on the 12 rigid object categories in Pascal3D+ (Xiang et al., 2014). A simple cross-entropy loss is used for the angle bin classification problem.

Geometry aware fine-grained classification. While being simple and direct, (Tulsiani and Malik, 2015) can only estimate coarsely the three Euler angles as they assign an angle bin of size 15 degrees for the angle bin classification. (Su et al., 2015b) thus propose a fine-grained classification by using angular bins of size 1 degree for each Euler angle, which makes the estimation more informative and accurate. Moreover, they propose a geometric structure aware loss function to exploit the geometric constraints existed in the three Euler angles. In particular, instead of treating each angle bin independently, they add an exponential decay weight with respect to viewpoint distance in the original classification loss for the mis-classified angle bins (Figure 2.14). This explicitly encourage correlation among the viewpoint predictions of nearby views. (Massa et al., 2016a) further explore the usage of this geometry structure aware classification loss by combining it with a object category classification loss for joint object detection and pose estimation.

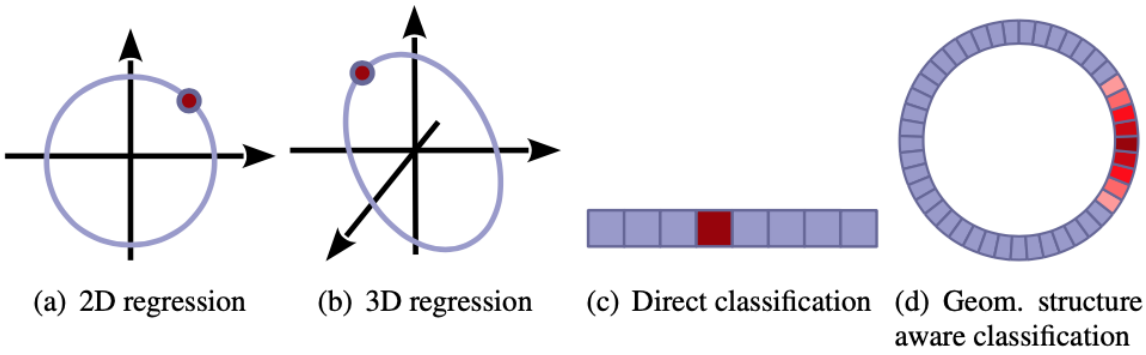


Figure 2.14: Figure from (Massa et al., 2016a). Geometry structure aware classification imposes to penalize less for angle bins close to the correct one.

2.3.2 Regression

Trigonometric representation of Euler angles. Instead of directly regressing a scalar value for an Euler angle θ, (Penedones et al., 2012) choose to regress a two-dimensional vector: $(\cos\theta, \sin\theta)$. This two-dimensional vector is also known as the trigonometric representation of an angle by choosing sine and cosine as the trigonometric functions to relate an angle to ratios of two side lengths. By doing this, the output can be any values in $\mathbb{R}^2$ to form an angle with a L2 normalization. Moreover, this makes it easier to encode the modulo property compared to a single scalar value. For instance, 0 degree is the same as 360 degrees in the trigonometric representation, while a huge distance exists between them in the scalar representation.

In order to achieve robustness against images of varying qualities, (Prokudin et al., 2018) propose a novel probabilistic deep learning model for viewpoint estimation. More specially, they train a deep network to predict the distribution over the three Euler angles, where von Mises distributions are considered in their method. As illustrated in Figure 2.15, based on the probabilistic predictions, they can model the uncertainty in the network outputs and improve the estimations using Bayesian decision theory.

Following this trigonometric representation based regression approach, (Liao et al., 2019) rethink the angle regression problem in the context of 1-spheres for each Euler angle by proposing a novel spherical exponential activation function. In particular, for the two-dimensional vector $(\cos\theta, \sin\theta)$, two regression branches are used to predict their absolute amplitudes, while one classification branch is used to predict the signs for both of them: (+-, ++, -+, –). By casting the prediction of Euler angle

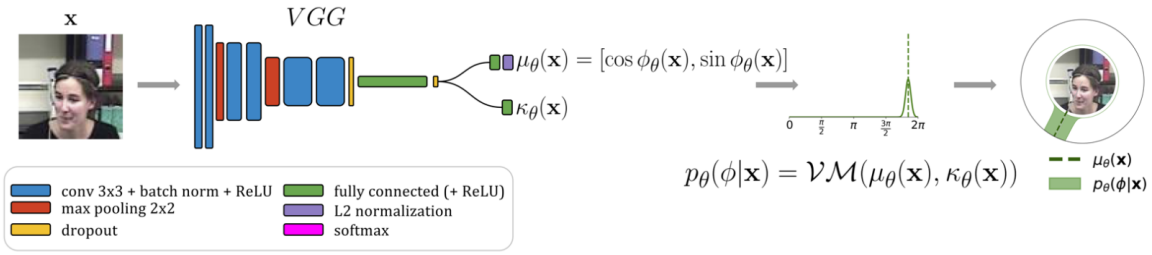


Figure 2.15: Figure from (Prokudin et al., 2018). Based on the trigonometric representation of Euler angles, a probabilistic prediction is used to capture the estimation uncertainty.

trigonometric representation as a combination of amplitude regression and vector sign classification, they achieve the best viewpoint estimation results on Pascal3D+.

Axis-angle or Quaternions. Instead of decomposing the rotation matrix into three Euler angles, (Mahendran et al., 2017) propose to represent the rotation matrix using the three-dimensional axis-angle or the four-dimensional quaternion. In the axis-angle representation $v = \theta[v_1, v_2, v_3]$, the rotation matrix captures the rotation of 3D points by an angle θ about an axis $[v_1, v_2, v_3]$, where the rotation axis is an unit vector. By restricting the rotation angle between 0 and π, they ensure an unique mapping between the rotation matrix and its corresponding axis-angle vector. On the other hand, given a three-dimensional axis-angle vector, the corresponding quaternion is represented as a four-dimensional vector: $[\cos\frac{\theta}{2}, v_1\sin\frac{\theta}{2}, v_2\sin\frac{\theta}{2}, v_3\sin\frac{\theta}{2}]$. By reconstruction, quaternions are unit-norm. Based on these representations, they also propose to compare directly the estimated rotation matrix against the ground-truth rotation matrix by optimizing the network using the geodesic loss function.

Continuous rotation representation. While many works represent the 3D rotation matrix with quaternions or Euler angles, (Zhou et al., 2019) demonstrate that, for 3D rotations, all representations are discontinuous in the real Euclidean spaces $\mathbb{R}^n$ when $n \leq 4$ (Figure 2.16). They argue that quaternions or Euler angles are thus discontinuous and difficult for neural networks to regress directly, and they propose a 6D representation as a remedy for discontinuous 3D rotation representations. They show theoretically and empirically the advantages of using the proposed continuous rotation representation for network training, when the loss function is implemented as the L2 distance between two rotation matrix.

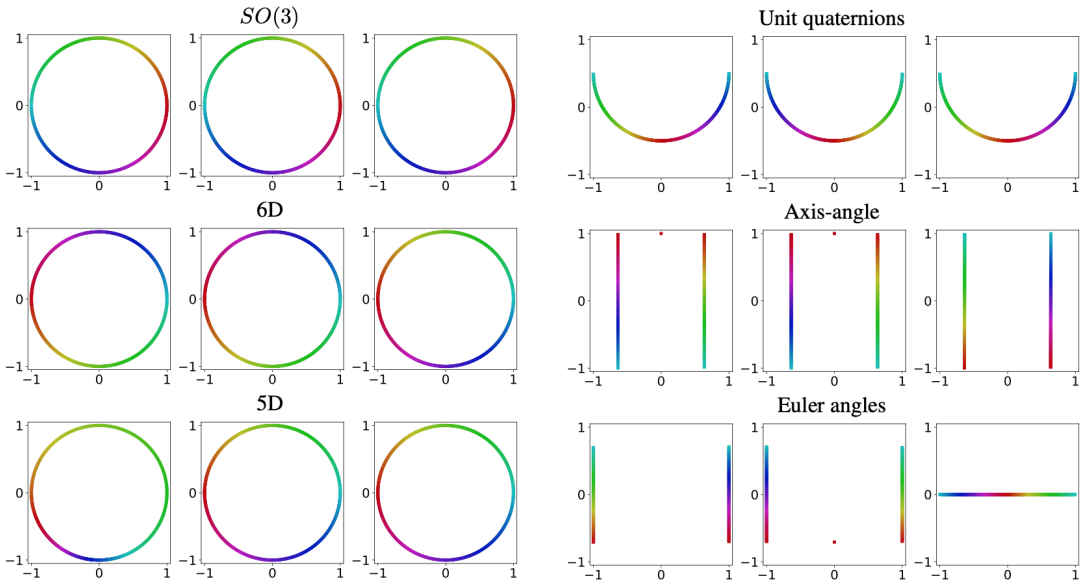


Figure 2.16: Figure from (Zhou et al., 2019). Visualization of discontinuities in 3D rotation representations. Each column represent the rotation about the corresponding axis for "X", "Y" and "Z". The curve in 2D should be homeomorphic to a circle with similar colors in nearby spatial locations if the representation is continuous.

2.3.3 Mixed classification-and-regression

As the 3D rotation lies in a continuous space, a natural way to estimate it would be a direct regression approach, where the 3D rotation can be represented by Euler angles, axis-angles, or quaternions, etc. While being straightforward, a major disadvantage of regression based approaches is that they fail to capture the potential multimodal distributions in the pose space. To handle this issue, an alternative approach is to cast it as a classification problem by discretizing the pose space into discrete bins. By predicting a probability for each bin instead of regressing a single scalar value, the classification based approaches are able to handle ambiguous cases where two or more hypotheses are plausible. However, the accuracy of classification based approaches is limited to the discretization granularity, it requires fine-grained bins and carefully designed classification loss that considers the circular property of 3D rotation angles.

A hybrid approach combining regression and classification thus becomes as an appealing solution to object viewpoint estimation. Recent trends to learn object viewpoint estimation solve the problem by first discretizing the viewpoint space into discrete bins and then regressing the delta values within the correct bins. We first focus on approaches based on Euler angles where prediction is done independently

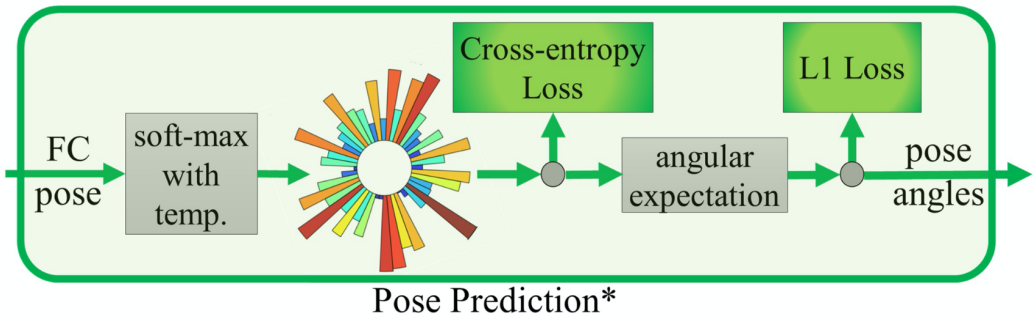


Figure 2.17: Figure from (Kundu et al., 2018). An angular expectation is used to get continuous predictions from discrete angle bins.

for each angle. We then discuss another line of work that operates on quaternions or axis-angles with discrete pose centers selected by clustering the dataset-dependent pose annotations.

In Chapter 3, we introduce a new hybrid approach that works directly on Euler angles with simple classification loss and regression loss. And this hybrid approach is used in all works presented in this thesis.

Angle bin classification and offset regression. (Mousavian et al., 2017) introduce a *MultiBin* approach by dividing the orientation angle into n overlapping bins, where a deep network is trained to estimate both a classification score that the output angle lies inside a specific bin and the residual rotation offset that needs to be applied to the orientation of the bin center in order to obtain the final output angle. These residual rotation offsets are represented by their trigonometric representations, namely the sine and the cosine of the offset angle. A cross-entropy loss is used for the angle bin classification, and a cosine distance is used to measure the error between the predictions and pose annotations. As overlapping angle bins are considered, they thus force all the bins covering the ground truth angle to estimate the correct angle.

3D-RCNN (Kundu et al., 2018) also take the best of both approaches by combining regression and classification loss. As illustrated in Figure 2.17, they first divide the prediction interval of each Euler angle into a set of discrete angle bins, and apply a cross-entropy loss for the angle bin classification problem. Different from the *MultiBin* approach of (Mousavian et al., 2017), 3D-RCNN avoids non-differentiable operations like arg max by introducing an additional temperature parameter to get the soft arg max probabilities. This enables them to get the final angle prediction through an angular expectation, where the centers of all angle bins are summed together with

weights computed as the probabilities. And they adopt a simple L1 loss for minimizing the distance between the continuous predictions and ground-truth annotations.

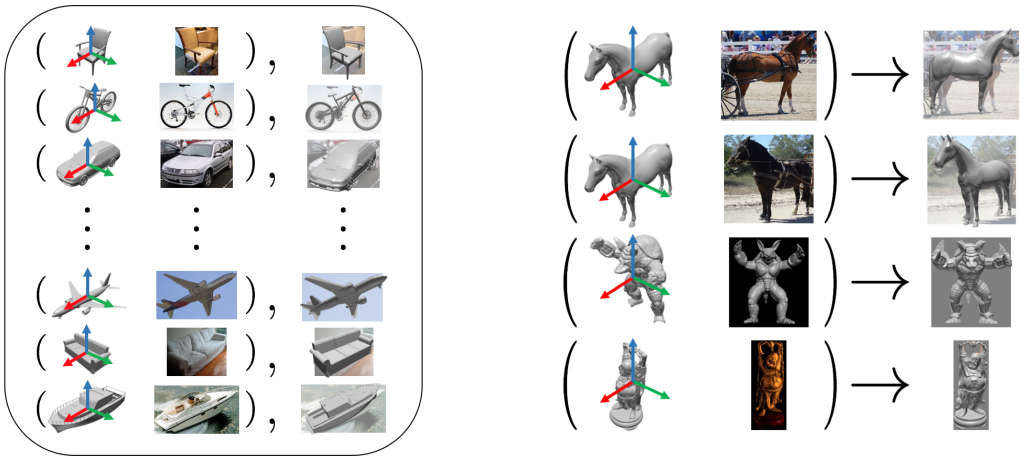
While the angular expectation used in 3D-RCNN allows end-to-end training with a Render-and-Compare layer, one of the main limitation is that this weighted-average outputs fail to capture the multi-modal distributions of certain objects. Unlike 3D-RCNN that only focuses on a specific object category such as car or motorbike, the mixed classification and regression approach used in this thesis aims to handle the viewpoint estimation problem across many different object categories, where it is important to capture the multi-modal distributions in the case of ambiguity.

Clustering based key pose selection. Except the Euler angles, other representations of 3D rotations have also been considered in recent works. (Mahendran et al., 2018) use the axis-angle representation and propose a general mixed classification-regression framework that covers a variety of different models and loss functions. They perform K-Means clustering on the ground-truth pose annotations to get a set of discrete key poses for the classification, and use a regression network to predict the residuals between the key poses and the ground-truth poses. To get a more robust clustering than K-Means, (Kuo et al., 2020) compute the set of discrete bins by K-Medoid clustering based on ground-truth pose annotations. Besides, they use quaternions to represent the 3D rotations. While (Mahendran et al., 2018) propose to implement the regression loss as a geodesic distance between the final continuous predictions and the ground-truth pose annotations, (Kuo et al., 2020) optimize directly on the residuals with a smooth-L1 loss.

Clustering on the training pose annotations would introduce a bias in the network predictions, which could hamper the generalization towards object classes with a different pose distributions. In this thesis, the proposed mixed classification-and-regression approach simply divides the three Euler angles into discrete bins with fixed bin size, without any specific assumption on the pose distributions.

Chapter 3

Deep Pose Estimation for Arbitrary 3D Objects



(a) Training with shape and pose (b) Testing on unseen objects

Figure 3.1: Illustration of our approach. (a) Training data: 3D model, input image and pose annotation for everyday man-made object; (b) At testing time, pose estimation of any arbitrary object, even an unknown category, given a RGB image and the corresponding 3D shape.

Abstract

In this chapter, we propose a completely generic deep pose estimation approach, which does not require the network to have been trained on specific categories, nor objects in a category to have a canonical pose. Our main insight is to dynamically condition pose estimation with a representation of the 3D shape of the target object. More precisely, we train a Convolutional Neural Network that takes as input both a test image and a 3D model, and outputs the relative 3D pose of the object in the input image with respect to the 3D model. We demonstrate that our method boosts performances for supervised category pose estimation on standard benchmarks, namely Pascal3D+, ObjectNet3D and Pix3D, on which we provide results superior to the state of the art. More importantly, we show that our network trained on everyday man-made objects from ShapeNet generalizes without any additional training to completely new types of 3D objects by providing results on the LINEMOD dataset as well as on natural entities such as animals from ImageNet.

The work presented in this chapter was initially presented in:

"Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects", Yang Xiao, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, Renaud Marlet, In *British Machine Vision Conference (BMVC 2019)*.

3.1 Introduction

Imagine a robot that needs to interact with a new type of object not belonging to any pre-defined category, such as a newly manufactured object in a workshop. Using existing single-view pose estimation approaches for this new object would require stopping the robot and training a specific network for this object before taking any further action. Here we propose an approach that can directly take as input a 3D model of the new object and estimate the pose of the object in images relatively to this model, without any additional training procedure. We argue that such a capability is necessary for applications such as robotics "in the wild", where new objects of unfamiliar categories can occur routinely at any time and have to be manipulated or taken into account for action. It also applies to virtual reality with similar circumstances.

To overcome the fact that deep pose estimation methods were category-specific, i.e., predicted different orientations according to object category, recent works (Grabner et al., 2018; Zhou et al., 2018) have proposed to perform category-agnostic pose estimation on rigid objects, producing a single prediction. However, (Grabner et al., 2018) only evaluated on object categories that were included in the training data, while (Zhou et al., 2018) required the testing categories to be similar to the training data. On the contrary, we want to stress that our method works on totally novel objects that can be widely different from those seen at training time. For example, we can train only on man-made objects, but still be able to estimate the pose of animals such as horses, whereas not a single animal has been seen in the training data (cf. Figure 3.1 and 3.4). Our method is similar to category-agnostic approaches in that it only produces one pose prediction and does not require additional training to produce predictions on novel categories. However, it is also instance-specific, because it takes as input a 3D model of the object of interest.

Indeed, our key idea is that viewpoint is better defined for a single object instance given its 3D shape than for whole object categories. Our work can be viewed as leveraging the recent advances in deep 3D model representations (Su et al., 2015a; Qi et al., 2017a,b) for the problem of pose estimation. We show that using 3D model information also boosts performances on known categories, even when the information is only approximate, as in the Pascal3D+ (Xiang et al., 2014) dataset.

When an exact 3D model of the object is known, as in the LINEMOD (Hinterstoißer et al., 2012b) dataset, state-of-the-art results are typically obtained by first performing a coarse viewpoint estimation and then applying a pose-refinement ap-

proach, typically matching rendered images of the 3D model to the target image. Our method is designed to perform the coarse alignment. Pose-refinement can be performed after applying our method using a classical approach based on ICP or the recent DeepIM (Li et al., 2018b) method. Note that while DeepIM only performs refinement, it is similar to our work in the sense that it is category agnostic and leverages some knowledge of the 3D model, using a view rendered in the estimated pose, to predict its pose update.

Our contributions in this chapter are as follows:

- To the best of our knowledge, we present the first deep learning approach to category-free viewpoint estimation, which can estimate the pose of any object conditioned only on its 3D model, whether or not it is similar to objects seen at training time.
- We can learn with and use "shapes in the wild", whose reference frame do not have to be consistent with a canonical orientation, simplifying pose supervision.
- We demonstrate on a large variety of datasets that adding 3D knowledge to pose estimation networks provides performance boosts when applied to objects of known categories, and meaningful performances on previously unseen objects.

All the code is available at the project webpage [1].

3.2 Related Work

In this section, we discuss pose estimation of a rigid object from a single RGB image first in the case where the 3D model of the object is known, then when the 3D model is unknown.

Pose estimation explicitly using object shape. Traditional methods to estimate the pose of a given 3D shape in an image can be roughly divided into feature-matching methods and template-matching methods. Feature-matching methods try to extract local features from the image, match them to the given object 3D model and then use a variant of PnP algorithm to recover the 6D pose based on estimated 2D-to-3D correspondences. Increasingly robust local feature descriptors (Lowe, 2004;

[1] http://imagine.enpc.fr/~xiaoy/PoseFromShape/

Tola et al., 2010; Tulsiani and Malik, 2015; Pavlakos et al., 2017) and more effective variants of PnP algorithms (Lepetit et al., 2009; Zheng et al., 2013; Li et al., 2012; Ferraz et al., 2014) have been used in this type of pipeline. Pixel-level prediction, rather than detected features, has also been proposed (Brachmann et al., 2016). Although performing well on textured objects, these methods usually struggle with poorly-textured objects. To deal with this type of objects, template-matching methods try to match the observed object to a stored template (Li et al., 2011; Lowe, 1991; Hinterstoißer et al., 2012a,b). However, they perform badly in the case of partial occlusion or truncation.

More recently, deep models have been trained for pose estimation from an image of a known or estimated 3D model. Most methods estimate the 2D position in the test image of the projections of the object 3D bounding box (Rad and Lepetit, 2017; Tekin et al., 2018; Oberweger et al., 2018; Grabner et al., 2018) or object semantic keypoints (Pavlakos et al., 2017; Georgakis et al., 2018) to find 2D-to-3D correspondences and then apply a variant of the PnP algorithm, as feature-matching methods. Once a coarse pose has been estimated, deep refinement approaches in the spirit of template-based methods have also been proposed (Manhardt et al., 2018; Li et al., 2018b).

Pose estimation not explicitly using object shape. In recent years, with the release of large-scale datasets (Geiger et al., 2012; Hinterstoißer et al., 2012b; Xiang et al., 2014, 2016; Sun et al., 2018), data-driven learning methods (on real and/or synthetic data) have been introduced which do not rely on an explicit knowledge of the 3D models. These can roughly be separated into methods that estimate the pose of any object of a training category and methods that focus on a single object or scene. For category-wise pose estimation, a canonical view is required for each category with respect to which the viewpoint is estimated. The prediction can be cast as a regression problem (Osadchy et al., 2007; Penedones et al., 2012; Massa et al., 2016a), a classification problem (Tulsiani and Malik, 2015; Su et al., 2015b; Elhoseiny et al., 2016) or a combination of both (Mousavian et al., 2017; Güler et al., 2017; Li et al., 2018a; Mahendran et al., 2018). Besides, (Zhou et al., 2018) directly regress category-agnostic 3D keypoints and obtain the final object pose using a PnP-like algorithm.

Following the same strategy, it is also possible to estimate the pose of a camera with respect to a single 3D model but without actually using the 3D model informa-

tion. Many recent works have applied this strategy to recover the full 6-DoF pose for object (Tjaden et al., 2017; Mousavian et al., 2017; Kehl et al., 2017; Xiang et al., 2018; Li et al., 2018a) and camera re-localization in the scene (Kendall et al., 2015; Kendall and Cipolla, 2017).

In this chapter, we propose to bring together the two lines of work described above. We cast pose estimation as a prediction problem, similar to deep learning methods that do not explicitly leverage viewpoint information. However, we condition our network on the 3D model of a single instance, represented either by a set of views or a point cloud, allowing our network to rely on the exact 3D model, similarly to the feature and template matching methods. To the best of our knowledge, we are the first to combine image and shape information as input to a network to estimate the relative orientation of the depicted object with respect to the shape, which does not require knowing a canonical frame of the shape.

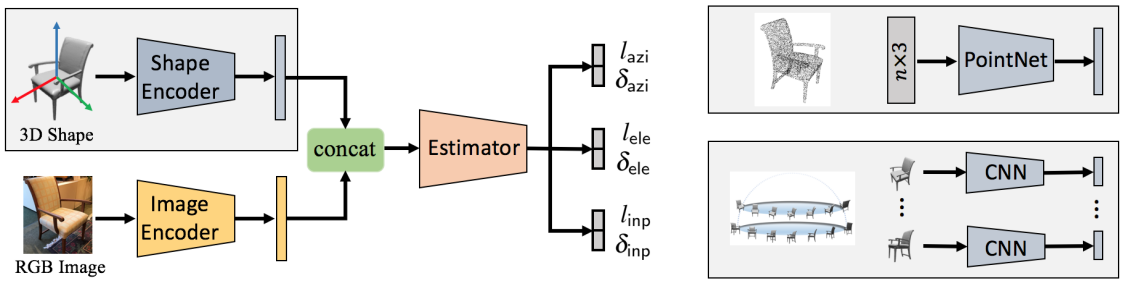
3.3 Method

Our approach consists in extracting deep features from both the image and the shape, and using them jointly to estimate a relative orientation. An overview is shown in Figure 3.2. In this section, we present in more details our architecture, our loss function and our training strategy, as well as a data augmentation scheme specifically designed for our approach.

Feature extraction. The first part of the network consists of two independent modules: (i) image feature extraction and (ii) 3D shape feature extraction. For image features, we use a standard CNN, namely ResNet-18 (He et al., 2016). For 3D shape features, we experimented with two approaches depicted in Figure 3.2(b) which are state-of-the-art 3D shape description networks.

First, we used the point set embedding network PointNet (Qi et al., 2017a) that has been successfully used as a point cloud encoder for many tasks (Engelmann et al., 2017; Groueix et al., 2018; Qi et al., 2018; Wang et al., 2019b; Xu et al., 2018).

Second, we tried to represent the shape using rendered views, similar to (Su et al., 2015a). Virtual cameras are placed around the 3D shape, pointing towards the centroid of the model; the associated rendered images are taken as input by CNNs, sharing weights for all viewpoints, which extract image descriptors; a global feature



(a) Our pose estimation approach (b) Two possible shape encoders

Figure 3.2: Overview of our method. (a) Given an RGB image of an object and its 3D shape, we use two encoders to extract features from each input, then estimate the orientation of the pictured object w.r.t. the shape using a classification-and-regression approach, predicting probabilities of angle bins l and bin offsets δ. (b) For shape encoding, we encode a point cloud sampled on the object with PointNet (top), or we rendered images around the object and use a CNN to extract the features (bottom).

vector is obtained by concatenation. We considered variants of this architecture using extra input channels for depth and/or surface normal orientation but this did not improve our results significantly. Ideally, we would consider viewpoints on the whole sphere around the object with any orientation. In practice however, many objects have a strong bias regarding verticality and are generally seen only from the side/top. In our experiments, we thus only considered viewpoints on the top hemisphere and sampled evenly a fixed number of azimuths and elevations.

Orientation estimation. The object orientation is estimated from both the image and 3D shape features by a multi-layer perceptron (MLP) with three hidden layers of size 800-400-200. Each fully connected layer is followed by a batch normalization, and a ReLU activation. As output, we estimate the three Euler angles of the camera, azimuth (azi), elevation (ele) and in-plane rotation (inp), with respect to the shape reference frame. Each of these angles $\theta \in \{\mathsf{azi}, \mathsf{ele}, \mathsf{inp}\}$ is estimated using a mixed classification-and-regression approach, which computes both angular bin classification scores and offset information within each bin.

Concretely, we split each Euler angle θ uniformly in N_θ bins with a fixed bin size ($\pi/12$ in our experiments). As azimuth and in-plane rotation angles vary from $-\pi$ to π, and elevation vary from $-\pi/2$ to $\pi/2$, we thus have 24 angle bins for azi and inp, and 12 for ele. For the i-th bin of angle θ, the network outputs a probability $p_{\theta,i} \in [0, 1]$ using a softmax non-linearity on the θ-bin classification scores, and an

offset $\delta_{\theta,i} \in [0,1]$ relatively to the beginning of the corresponding angle bin. The network thus has $2 \times (N_{\mathsf{azi}} + N_{\mathsf{ele}} + N_{\mathsf{inp}})$ outputs, where $N_{\mathsf{azi}}, N_{\mathsf{inp}} = 24$ and $N_{\mathsf{ele}} = 12$.

Loss function. As we combine classification and regression, our network has two types of outputs (probabilities and offsets), that are combined into a single loss $\mathcal{L}$ that is the sum of a cross-entropy loss for classification $\mathcal{L}_{\text{cls}}$ and smooth-L1 loss for regression $\mathcal{L}_{\text{reg}}$.

More formally, given a training sample (I, S, y) consisting of input image I, associated object shape S, and corresponding orientation $y = (y_\theta)_{\theta \in \{\mathsf{azi},\mathsf{ele},\mathsf{inp}\}}$. We convert the value of the Euler angles y_θ into a bin label bin_θ encoded as a one-hot vector and relative offsets δ_θ within the bins. The network parameters are learned by minimizing:

$$\mathcal{L} = \sum_{\theta \in \{\mathsf{azi},\mathsf{ele},\mathsf{inp}\}} \mathcal{L}_{\text{cls}}\Big(\text{bin}_\theta, p_\theta(I,S)\Big) + \mathcal{L}_{\text{reg}}\Big(\delta_\theta, \delta_{\theta,\text{bin}_\theta}(I,S)\Big), \tag{3.1}$$

where $p_\theta(I,S)$ are the probabilities predicted by the network for angle $\theta \in \{\mathsf{azi}, \mathsf{ele}, \mathsf{inp}\}$, from input image I and input shape S, and $\delta_{\theta,\text{bin}_\theta}(I,S)$ the predicted offset within the ground truth bin.

Data augmentation. We perform standard data augmentation on the input images: horizontal flip, 2D bounding box jittering, color jittering.

In addition, we introduce a new data augmentation, specific to our approach, designed to avoid the network to overfit the 3D model orientation, which has a biased distribution in training data since most images are taken near the front-view of object. On the contrary, we want our network to be category-agnostic and to always predict the pose of the object with respect to the reference 3D model. We thus add random rotations to the input shapes, and modify the orientation labels accordingly. In our experiments, we restrict our rotations to azimuth changes, again because of the strong verticality bias in the benchmarks, but could theoretically apply it to all angles. Because of objects with symmetries, typically at 90° or 180°, we also restrict azimuthal randomization to a uniform sampling in $[-45°, 45°]$, which allows to keep the 0° bias of the annotations.

Implementation details. For all our experiments, we set the batch size as 16 and trained our network using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-4} for 100 epochs then 10^{-5} for an additional 100 epochs. Compared to a

shape-less baseline method, the training of our method with the shape encoded from 12 rendered views is about 8 times slower, on a TITAN X GPU.

3.4 Results

Given an RGB image of an object and a 3D model of that object, our method estimates its 3D orientation in the image. In this section, we first give an overview of the datasets we used, and explain our baseline methods. We then evaluate our method in two test scenarios: object belonging to a category known at training time, or unknown.

Datasets. We experimented with four main datasets. Pascal3D+ (Xiang et al., 2014), ObjectNet3D (Xiang et al., 2016) and Pix3D (Sun et al., 2018) feature various objects in various environments, allowing benchmarks for object pose estimation in the wild. On the contrary, LINEMOD (Hinterstoißer et al., 2012b) focuses on few objects with little environment variations, targeting robotic manipulation. Pascal3D+ and ObjectNet3D only provide approximate models and rough alignments while Pix3D and LINEMOD offer exact models and pixelwise alignments. We also used ShapeNetCore (Chang et al., 2015) for training on synthetic data, with SUN397 backgrounds (Xiao et al., 2010), and tested on Pix3D and LINEMOD.

ShapeNetCore is a subset of ShapeNet containing 51k single clean 3D models, covering 55 common object categories of man-made artifacts. We exclude the categories containing mostly objects with rotational symmetry or small and narrow objects, which results in 30 remaining categories: *airplane, bag, bathtub, bed, birdhouse, bookshelf, bus, cabinet, camera, car, chair, clock, dishwasher, display, faucet, lamp, laptop, speaker, mailbox, microwave, motorcycle, piano, pistol, printer, rifle, sofa, table, train, watercraft* and *washer*. We randomly choose 200 models from each category and use Blender to render each model under 20 random views with various textures included in ShapeNetCore.

Evaluation Metrics. Unless otherwise stated, ground-truth bounding boxes are used in all experiments. We compute the most common metrics used with each dataset. For results on Pascal3D+, ObjectNet3D, and Pix3D, we use two common metrics: Acc30 is the percentage of estimations with rotation error less than 30°; MedErr is the median angular error (°). The pose prediction error is computed as

the geodesic distance between two rotation matrices:

$$\Delta(\mathbf{R}_{\text{pred}}, \mathbf{R}_{\text{gt}}) = \arccos\left(\frac{\text{tr}(\mathbf{R}_{\text{pred}}^T \mathbf{R}_{\text{gt}}) - 1}{2}\right) \tag{3.2}$$

where $\text{tr}(\cdot)$ means the trace of a matrix.

For results on LINEMOD, the ADD (Hinterstoißer et al., 2012b) metric is used to compute the averaged distance between points transformed using the estimated pose and the ground truth pose:

$$\textbf{ADD} = \frac{1}{m} \sum_{\mathbf{x} \in \mathcal{M}} ||(\mathbf{R}\mathbf{x} + \mathbf{t}) - (\hat{\mathbf{R}}\mathbf{x} + \hat{\mathbf{t}})|| \tag{3.3}$$

where m is the number of points on the 3D object model, $\mathcal{M}$ is the set of all 3D points of this model, $[\mathbf{R}|\mathbf{t}]$ is the ground truth pose and $[\hat{\mathbf{R}}|\hat{\mathbf{t}}]$ is the estimated pose. Following (Brachmann et al., 2016), we compute the model diameter d as the maximum distance between all pairs of points from the model. With this metric, a pose estimation is considered to be correct if the computed averaged distance is within 10% of the model diameter d. For the objects with ambiguous poses due to symmetries, (Hinterstoißer et al., 2012b) replaces this measure by ADD-S, which uses the closest point distance in computing the average distance for 6D pose evaluation:

$$\textbf{ADD-S} = \frac{1}{m} \sum_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} ||(\mathbf{R}\mathbf{x}_1 + \mathbf{t}) - (\hat{\mathbf{R}}\mathbf{x}_2 + \hat{\mathbf{t}})|| \tag{3.4}$$

Baselines. A natural baseline is to use the same architecture, data and training strategy as for our approach, but without using the 3D shape of the object. This is reported as 'Baseline' in our tables, and corresponds to the network of Figure 3.2 without the shape encoder shown in light blue. We also report a second baseline, aiming at evaluating the importance of the precision of the 3D model for our approach to work. We used exactly our approach, but at testing time we replaced the 3D shape of the object in the test image by a random 3D shape of the same category. This is reported as 'Ours (RS)' in the tables.

ObjectNet3D	bed	bcase	calc	cphone	comp	door	cabi	guit	iron	knife	micro	pen	pot	rifle	shoe	slipper	stove	toilet	tub	wchair	mean
category-specific networks/branches — test on supervised categories (Acc30)																					
(Xiang et al., 2016)*	61	85	93	60	78	90	76	75	17	23	87	33	77	33	57	22	88	81	63	50	62
category-agnostic network — test on supervised categories (Acc30)																					
(Zhou et al., 2018)	73	78	91	57	82	–	84	73	3	18	94	13	56	4	–	12	87	71	51	60	56
Baseline	70	89	90	55	87	91	88	62	29	20	93	43	76	26	58	30	91	68	51	55	64
Ours(PC)	**83**	**92**	95	58	82	87	**91**	67	43	**36**	94	53	81	39	45	35	91	80	65	56	69
Ours(MV,RS)	74	89	91	62	81	90	88	71	41	28	94	50	70	37	57	38	89	81	60	60	68
Ours(MV)	82	90	**96**	**65**	**93**	**97**	89	**75**	**52**	32	**95**	**54**	**82**	**45**	**67**	**46**	**95**	**82**	**67**	**66**	**73**
category-agnostic network — test on novel categories (Acc30)																					
(Zhou et al., 2018)	37	69	19	52	73	–	78	61	2	9	88	12	51	0	–	11	82	41	49	14	42
Baseline	56	79	26	53	77	86	83	51	4	16	90	42	65	2	34	22	86	43	50	35	50
Ours(PC)	63	85	84	51	**85**	83	83	61	**9**	**35**	92	44	**80**	8	39	20	87	56	**71**	**39**	59
Ours(MV,RS)	60	88	84	60	76	91	82	61	2	26	90	46	73	13	45	28	79	59	61	36	58
Ours(MV)	**65**	**90**	**88**	**65**	84	**93**	**84**	**67**	2	29	**94**	**47**	79	**15**	**54**	**32**	**89**	**61**	68	**39**	**62**

[images: 90,127, in the wild | objects: 201,888 | categories: 100 | 3D models: 791, approximate | alignment: rough]

Table 3.1: Pose estimation on ObjectNet3D (Xiang et al., 2016). Train and test are on the same data as (Zhou et al., 2018); for experiments on novel categories, training is on 80 categories and test is on the other 20. * Trained jointly for detection and pose estimation, tested using estimated bounding boxes.

3.4.1 Pose estimation on supervised categories

We first evaluate our method in case the categories of tested objects are covered by training data. We show that leveraging the 3D model of the object clearly improves pose estimation.

We evaluate our method on ObjectNet3D, which has the largest variety of object categories, 3D models and images. We report the results in Table 3.1 (top). First, an important result is that using the 3D model information, whether via a point cloud or rendered views, provides a very clear boost of the performance, which validates our approach. Second, results using rendered multiple views (MV) to represent the 3D model outperform the point-cloud-based (PC) representation (Qi et al., 2017a). We thus only evaluated Ours(MV) in the rest of this section. Third, testing the network with a random shape (RS) in the category instead of the ground truth shape, implicitly providing class information without providing fine-grained 3D information, leads to results better than the baseline but worst than using the ground truth model, demonstrating our method ability to exploit fine-grained 3D information. Finally, we found that even our baseline model already outperformed StarMap (Zhou et al., 2018), mainly because of five categories (iron, knife, pen, rifle, slipper) on which StarMap completely fails, likely because a keypoint-based method is not adapted for small and narrow objects.

We then evaluate our approach on the standard Pascal3D+ dataset (Xiang et al., 2014). Results are shown in Table 3.2 (top). Interestingly, while our baseline is far below state-of-the-art results, adding our shape analysis network provides again a very clear improvement, with results on par with the best category-specific approaches, and outperforming category agnostic methods. This is especially impressive considering the fact that the 3D models provided in Pascal3D+ are only extremely coarse approximations of the real 3D models. Again, as can be expected, using a random model from the same category provides intermediary results between the model-less baseline and using the actual 3D model.

Finally, we report results on Pix3D in Table 3.3 (top). Similar to the other methods, our model was purely trained on synthetic data and tested on real data, without any fine-tuning. Again, we can observe that adding 3D shape information brings a large performance boost, from 23.9% to 36% Acc30. Note that our method clearly improves even over category-specific baselines. We believe it is due to the much higher quality of the 3D models provided on Pix3D compared to ObjectNet3D

Pascal3D+	aero	bike	boat	bottle	bus	car	chair	dtable	mbike	sofa	train	tv	mean
category-specific branches, supervised categories	Acc30 (%)												
(Tulsiani and Malik, 2015)*	81	77	59	93	**98**	89	80	62	**88**	82	80	80	80.75
(Su et al., 2015b)†	74	**83**	52	91	91	88	**86**	**73**	78	**90**	**86**	**92**	82.00
(Mousavian et al., 2017)	78	**83**	57	93	94	90	80	68	86	82	82	85	81.03
(Pavlakos et al., 2017)*	81	78	44	79	96	90	80	–	–	74	79	66	–
(Grabner et al., 2018)	**83**	82	**64**	**95**	97	**94**	80	71	**88**	87	80	86	**83.92**
Category-agnostic network, supervised categories	Acc30 (%)												
(Grabner et al., 2018)	80	82	57	90	**97**	**94**	72	**67**	**90**	80	**82**	**85**	81.33
(Zhou et al., 2018)*	**82**	**86**	50	92	**97**	92	**79**	62	88	**92**	77	83	81.67
Baseline	77	74	54	91	**97**	89	74	52	85	80	79	77	77.42
Ours(MV,RS)	79	81	49	91	96	89	78	53	**90**	88	80	77	79.25
Ours(MV)	81	83	**60**	**93**	**97**	91	**79**	**67**	**90**	90	81	79	**82.66**
Category-specific branches, supervised categories	MedErr (degrees)												
(Tulsiani and Malik, 2015)*	13.8	17.7	21.3	12.9	5.8	9.1	14.8	15.2	14.7	13.7	8.7	15.4	13.6
(Su et al., 2015b)†	15.4	14.8	25.6	9.3	3.6	6.0	**9.7**	**10.8**	16.7	**9.5**	**6.1**	12.6	11.7
(Mousavian et al., 2017)	13.6	**12.5**	22.8	**8.3**	3.1	5.8	11.9	12.5	12.3	12.8	6.3	11.9	11.1
(Pavlakos et al., 2017)*	**8.0**	13.4	40.7	11.7	**2.0**	5.5	10.4	–	–	9.6	8.3	32.9	–
(Grabner et al., 2018)	10.0	15.6	**19.1**	8.6	3.3	**5.1**	13.7	11.8	**12.2**	13.5	6.7	**11.0**	**10.9**
Category-agnostic network, supervised categories	MedErr (degrees)												
(Grabner et al., 2018)	10.9	**12.2**	23.4	9.3	3.4	5.2	15.9	16.2	12.2	11.6	6.3	11.2	11.5
(Zhou et al., 2018)*	**10.1**	14.5	30.0	9.1	3.1	6.5	11.0	23.7	14.1	11.1	7.4	13.0	12.8
Baseline	13.0	18.2	27.3	11.5	6.8	8.1	15.4	20.1	14.7	13.2	10.2	14.7	14.4
Ours(MV,RS)	11.6	15.5	30.9	8.2	3.6	6.0	13.8	22.8	13.1	11.1	6.0	15.0	13.1
Ours(MV)	10.5	13.7	**21.0**	**7.7**	**3.0**	**5.0**	**10.9**	**11.9**	**11.8**	**9.1**	**5.4**	**10.3**	**10.0**

[images: 30,889, in the wild | objects: 36,292 | categories: 12 | 3D models: 79, approximate | alignment: rough]

Table 3.2: Pose estimation on Pascal3D+ (Xiang et al., 2014). * Trained using keypoints. † Not trained on ImageNet data but trained on ShapeNet renderings.

Pix3D	tool	misc	bcase	wdrobe	desk	bed	table	sofa	chair	mean
	class-specific networks — tested on seen classes (Acc30)									
(Georgakis et al., 2018)	-	-	-	-	**34.9**	**50.8**	-	-	**31.2**	-
	class-agnostic network — tested on seen classes (Acc30)									
Baseline	2.2	9.8	10.8	0.6	30.0	36.8	17.3	63.8	43.6	23.9
Ours(MV,RS)	4.1	3.6	22.8	9.5	52.8	50.1	30.8	66.3	44.5	31.6
Ours(MV)	**6.5**	**19.7**	**34.6**	**10.2**	**56.6**	**59.8**	**40.8**	**70.0**	**52.4**	**38.9**
	class-agnostic network — tested on unseen classes (Acc30)									
Baseline	2.2	**13.1**	5.4	0.6	30.3	19.6	14.9	11.9	28.0	14.0
Ours(MV,RS)	3.0	5.9	4.5	5.2	44.7	31.5	24.1	48.5	33.9	22.4
Ours(MV)	**10.9**	**13.1**	**22.3**	**6.6**	**52.0**	**55.3**	**35.6**	**64.6**	**35.8**	**32.9**

[images: 10,069, in the wild | objects: 10,069 | categories: 9 | 3D models: 395, exact | alignment: pixel]

Table 3.3: Pose estimation on Pix3D (Sun et al., 2018).

Pix3D	chair	
	class-specific, supervised	
# bins (% correct)	24 azimuth	12 elevation
(Su et al., 2015b)	40	37
(Sun et al., 2018)	49	61
Baseline	51	64
Ours(MV)	**54**	**65**

Table 3.4: Pose estimation on Pix3D (Sun et al., 2018) with comparison to (Su et al., 2015b; Sun et al., 2018), that only test bin success on 2 angles (24 azimuth bins and 12 elevation bins).

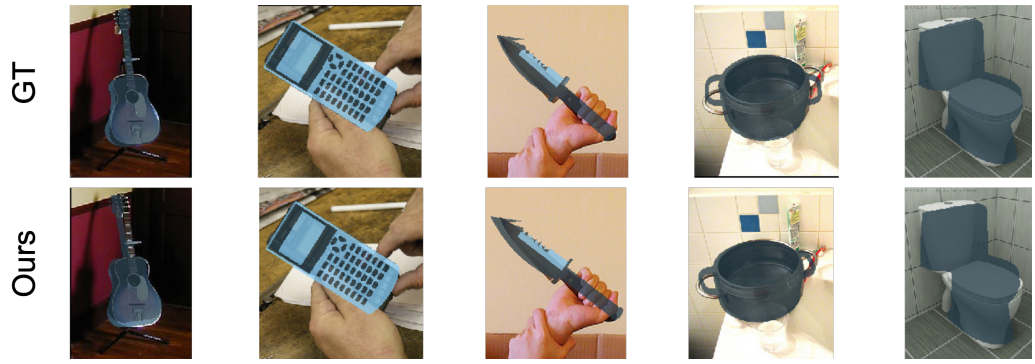


Figure 3.3: Visual results of object pose estimation for novel object classes on ObjectNet3D (Xiang et al., 2016).

and Pascal3D+. This hypothesis is supported by the fact that our results are much worse when a random model of the same category is provided.

These state-of-the-art results on the three standard datasets are thus consistent and validate (i) that using the 3D models provides a clear improvement (comparison to 'Baseline'), and (ii) that our approach is able to leverage the fine-grained 3D information from the 3D model (comparison to estimating with a random shape 'RS' in the category).

3.4.2 Pose estimation on novel categories

We now consider the generalization to unseen categories, which is the main focus of our method. We first discuss results on ObjectNet3D and Pix3D. We then show qualitative results on ImageNet horses images and quantitative results on the very different LINEMOD dataset.

Our results when testing on new categories from ObjectNet3D are shown in Table 3.1 (bottom). We use the same split between 80 training and 20 testing categories as (Zhou et al., 2018). As expected, the accuracy decreases for all methods when supervision is not provided on these test categories. The fact that the Baseline performances are still much better than chance is accounted by the presence of similar categories is the training set. The advantage of our method is however even more pronounced than in the supervised case, and our multi-view approach (MV) still outperforms the point cloud (PC) approach by a small margin. Similarly, we removed from our ShapeNet (Chang et al., 2015) synthetic training set the categories present in Pix3D, and reported in Table 3.3 (bottom) the results on Pix3D. Again, the ac-

Figure 3.4: Visual results of pose estimation on horse images from ImageNet (Deng et al., 2009) using models from Free3D. We rank the prediction for each orientation bin by the network prediction and show the first (best) results for various poses in images.

curacy drops for all methods, but the benefit from using the ground-truth 3D model increases.

In both ObjectNet and Pix3D experiments, the test categories were novel but still similar to the training ones. We now focus on evaluating our network, trained using synthetic images generated from man-made shapes from ShapeNetCore (Chang et al., 2015), on completely different object categories.

We first obtain qualitative results by using a fixed 3D model of horse from an online model repository Free3D[2] to estimate the pose of horses in ImageNet images. Indeed, compared to other animals, horses have more limited deformations. While this of course does not work for all images, the images for which the network provides the highest confidence are impressively good. On Figure 3.4, we show the most confident images for different poses. Note the very strong appearance gap between the rendered 3D models and the test images.

Finally, to further validate our network generalization ability, we evaluate it on the texture-less objects of LINEMOD (Hinterstoißer et al., 2012b), as reported in Table 3.5. This dataset focuses on very accurate alignment, and most approaches propose to first estimate a coarse alignment and then to refine it with a specific method. Our method provides a coarse alignment, and we complement it using the recent DeepIM refinement approach (Li et al., 2018b). Our method yields results below the state of the art, but they are nevertheless very impressive. Indeed, our network has never seen objects any similar the LINEMOD 3D models during training, while all the other baselines have been trained specifically for each object instance on real training images, except SSD-6D (Kehl et al., 2017) which uses the exact 3D model but no real image and for which coarse alignment performances are very low. Our method is thus very different from all the baselines in that it does not assume the

[2] https://free3d.com

LINEMOD		ape	bvise	cam	can	cat	drill	duck	ebox*	glue*	holep	iron	lamp	phone	mean
instance-specific networks/branches — tested on supervised models (ADD-0.1d)*															
w/o Refine	(Brachmann et al., 2016)	-	-	-	-	-	-	-	-	-	-	-	-	-	32.3
	(Kehl et al., 2017)‡	0	0.2	0.4	1.4	0.5	2.6	0	8.9	0	0.3	8.9	8.2	0.2	2.4
	(Rad and Lepetit, 2017)	**27.9**	62.0	40.1	48.1	45.2	58.6	32.8	40.0	27.0	42.4	67.0	39.9	35.2	43.6
	(Tekin et al., 2018)	21.6	**81.8**	36.6	68.8	41.8	63.5	27.2	69.6	80.0	42.6	**75.0**	**71.1**	47.7	56.0
	(Xiang et al., 2018)†	27.8	68.9	**47.5**	**71.4**	**56.7**	**65.4**	**42.8**	**98.3**	**95.2**	**50.9**	65.6	70.3	**54.6**	**62.7**
w/ Refine	(Brachmann et al., 2016)	33.2	64.8	38.4	62.9	42.7	61.9	30.2	49.9	31.2	52.8	80.0	67.0	38.1	50.2
	(Rad and Lepetit, 2017)	40.4	91.8	55.7	64.1	62.6	74.4	44.3	57.8	41.2	**67.2**	84.7	76.5	54.0	62.7
	(Kehl et al., 2017)‡	65	80	78	86	70	73	66	**100**	**100**	49	78	73	79	79.0
	(Xiang et al., 2018)† + DeepIM†	**76.9**	**97.4**	**93.5**	**96.6**	**82.1**	**95.0**	**77.7**	97.0	99.4	52.7	**98.3**	**97.5**	**87.8**	**88.6**
instance/category-agnostic network — tested on novel models (ADD-0.1d)*															
w/o Refine	Ours‡	**7.5**	**25.1**	**12.1**	**11.3**	**15.4**	**18.6**	**8.2**	**100**	**81.2**	**18.5**	**13.8**	**6.5**	**13.4**	**25.5**
w/ Refine	Ours‡ + DeepIM†	**59.1**	**63.8**	**40.0**	**50.8**	**54.1**	**75.3**	**48.6**	**100**	**98.7**	**49.8**	**49.5**	**55.3**	**50.4**	**61.2**

[scenes: 13, artificially arranged | images: 13407 | objects: 13 | categ.: 13 | 3D models: 13, exact | align.: pixel]

Table 3.5: Pose estimation on LINEMOD (Hinterstoißer et al., 2012b). † Training also on synthetic data. ‡ Training only on synthetic data. * ADD-S-0.1d used for symmetric objects eggbox and glue.

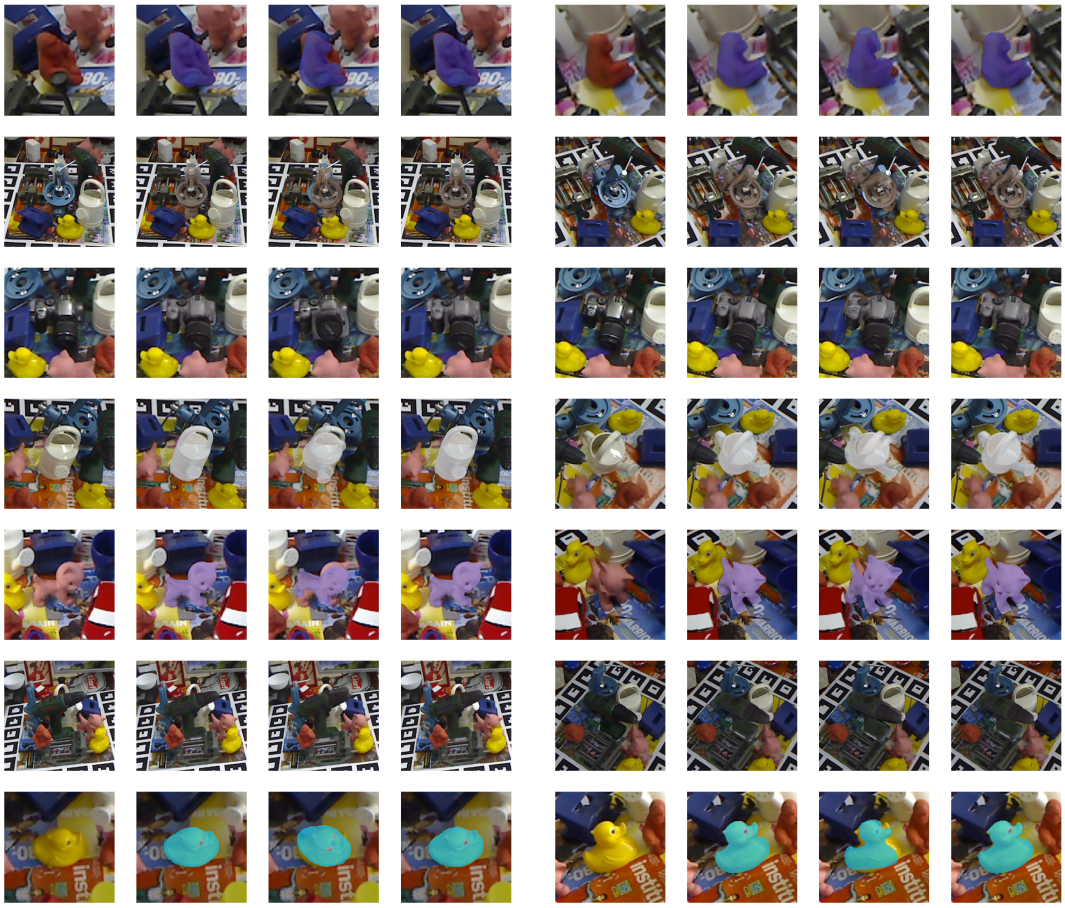


Figure 3.5: Visual results of object pose estimation on LINEMOD (Hinterstoißer et al., 2012b). For each sample, the four columns from left to right represent: the input image, the correct shape and orientation, our initial estimate and the final estimate after refining our initialization with DeepIM (Li et al., 2018b).

test object to be available at training time, which we think is a much more realistic scenario for robotics applications. We actually believe that the fact our method provides a reasonable accuracy on this benchmark is a very strong result.

Some qualitative results for the 13 LINEMOD objects are shown in Figure 3.5. Given object image and its shape, our approach gives a coarse pose estimate which is then refined by the pose refinement method DeepIM (Li et al., 2018b).

Figure 3.5: (cont.) Visual results of object pose estimation on LINEMOD (Hinterstoißer et al., 2012b). For each sample, the four columns from left to right represent: the input image, the correct shape and orientation, our initial estimate and the final estimate after refining our initialization with DeepIM (Li et al., 2018b).

$N_{\text{azi}} \times N_{\text{ele}}$	0	1×1	6×1	3×2	2×3	12×1	6×2	4×3	18×1	9×2	6×3
Acc30 ↑	50	56	59	60	58	59	**62**	58	58	60	59
MedErr ↓	50	45	44	44	51	46	**40**	46	51	43	45

Table 3.6: Ablation and parameter study on ObjectNet3D of the number and layout of rendering images at the input of the network when using multiple views to represent shape. Performance depending on the number of azimuthal and elevation samples.

3.4.3 Ablation study

Ablation and parameter study on the number of rendered images. Table 3.6 shows the experimental results of pose estimation on 20 novel categories of ObjectNet3D for different numbers and layouts of rendered images. The viewpoints are sampled evenly at N_{azi} azimuths and elevated at N_{ele} different elevations. $N_{\text{ele}} = 1, 2, 3$ represents respectively elevations at $(30°)$, $(0°, 30°)$, $(0°, 30°, 60°)$. The Acc30 metric measures the percentage of testing samples with a angular error smaller than $\frac{\pi}{6}$ and MedErr is the median angular error $(°)$ over all testing samples.

The table shows that using shape information encoded from rendered images (when $N_{\text{azi}} \times N_{\text{ele}} > 0$) can indeed help pose estimation on novel categories, i.e., that are not included in the training data. In the first column (0 rendered images) we show the performance of our baseline without using the 3D shape of the object, compared to this result, the network trained with only one rendered image has a clearly boosted accuracy.

The table also shows that more rendered images in the network input does not necessarily mean a better performance. In the table, the network trained with 12 rendered images elevated at 0°and 30°gives the best result. This may be because the ObjectNet3D dataset is highly biased towards low elevations on the hemisphere, which can be well represented without using the rendered image captured at high elevation such as 60°.

Parameter study on the azimuthal randomization strategy. Table 3.7 summarizes the parameter study on the range of azimuthal jittering applied to input shapes during network training. The poor results obtained for $[-0°, 0°]$ and $[-180°, 180°]$ are due the objects with symmetries, typically at 90°or 180°.

Randomization Range	$[-0°, 0°]$	$[-45°, 45°]$	$[-90°, 90°]$	$[-180°, 180°]$
Acc30 ↑	56	**62**	60	55
MedErr ↓	47	**40**	43	52

Table 3.7: Parameter study of azimuthal randomization used as a specific data augmentation of our approach. Performance depending on the range of azimuthal variation during training.

3.5 Conclusion

We have presented a new paradigm for deep pose estimation, taking the 3D object model as an input to the network. We demonstrated the benefits of this approach in terms of accuracy, and improved the state of the art on several standard pose estimation datasets. More importantly, we have shown that our approach holds the promise of a completely generic deep learning method for pose estimation, independent of the object category and training data, by showing encouraging results on the LINEMOD dataset without any specific training, and despite the domain gap between synthetic training data and real images for testing.

Chapter 4

Few-Shot Object Detection and Viewpoint Estimation

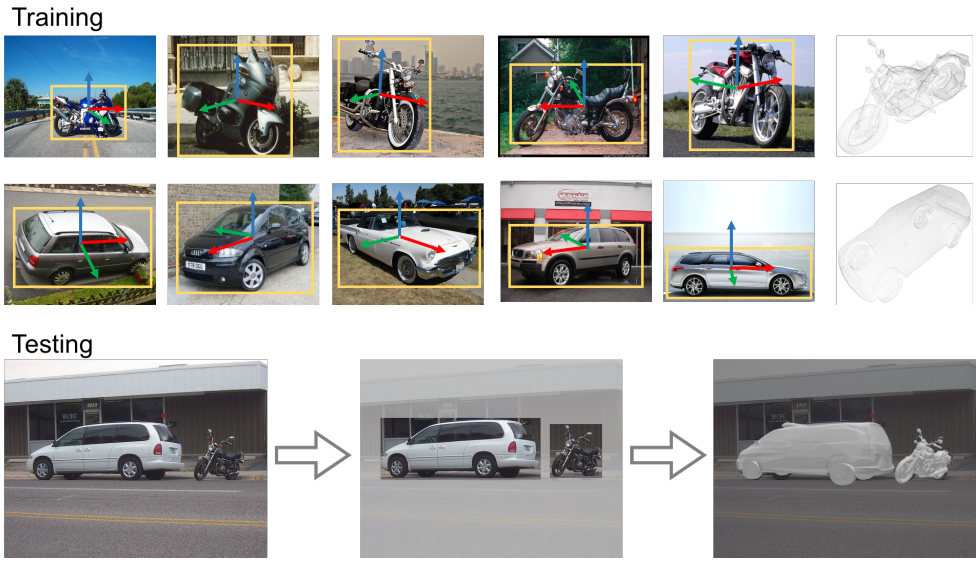


Figure 4.1: Starting with images labeled with bounding boxes and viewpoints of objects from base classes, and given only a few similarly-labeled images for new categories (top left), we predict in a query image the 2D location of objects of new categories, as well as their 3D poses (bottom), leveraging on just a few arbitrary 3D class models (top right).

Abstract

Detecting objects and estimating their viewpoint 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 propose a meta-learning framework that can be applied to both tasks, possibly including 3D data. Our models improve the results on objects of novel classes by leveraging on rich feature information originating from base classes with many samples. A simple joint feature embedding module is proposed to make the most of this feature sharing. Despite its simplicity, our method outperforms state-of-the-art methods by a large margin on a range of datasets. And for the first time, we tackle the combination of both few-shot object detection and few-shot viewpoint estimation, on ObjectNet3D, showing promising results.

The work presented in this chapter was initially presented in:

"Few-Shot Object Detection and Viewpoint Estimation for Objects in the Wild", Yang Xiao, Renaud Marlet, In *European Conference on Computer Vision (ECCV 2020)*.

4.1 Introduction

Detecting objects in 2D images and estimating their 3D pose, as shown in Figure 4.1, is extremely useful for tasks 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, that rely on rich labeled data, usually fail to generalize to *novel* object categories, when only a few annotated samples are available. Transferring the knowledge learned from large base categories with abundant annotated images to novel categories with scarce annotated samples is a *few-shot learning* problem.

To address few-shot detection, some approaches simultaneously tackle few-shot classification and few-shot localization by disentangling the learning of category-agnostic and category-specific network parameters (Wang et al., 2019c). Others attach a reweighting module to existing object detection networks (Kang et al., 2019; Yan et al., 2019). Though these methods have made significant progress, current few-shot detection evaluation protocols suffer from statistical unreliability and the prediction depends heavily on the choice of support data, which makes direct comparison difficult (Wang et al., 2020).

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 (Zhou et al., 2018; Xiao et al., 2019). However, these methods either require the testing categories to be similar to the training ones (Zhou et al., 2018), or assume the exact CAD model to be provided for each object during inference (Xiao et al., 2019). Differently, the meta-learning-based method MetaView (Tseng et al., 2019) introduces the category-level few-shot viewpoint estimation problem and addresses it by learning to estimate category-specific keypoints, requiring extra annotations. In any case, precisely annotating the 3D pose of objects in images is far more tedious than annotating their 2D bounding boxes, which makes few-shot viewpoint estimation a non-trivial, largely under-explored problem.

In this work, we propose a consistent framework to tackle both problems of few-shot object detection and few-shot viewpoint estimation. For this, we exploit, in a meta-learning setting, task-specific class information present in existing datasets, i.e., images with bounding boxes for object detection and, for viewpoint estimation, 3D

poses in images as well as a few 3D models for the different classes. Considering that these few 3D shapes are available is a realistic assumption in most scenarios. 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. Despite its simplicity, this approach leads to a significant performance improvement on novel classes under the few-shot learning regime.

Additionally, by combining our few-shot object detection with our few-shot viewpoint estimation, we address the realistic joint problem of learning to detect objects in images and to estimate their viewpoints from only a few shots. Indeed, compared to other viewpoint estimation methods, that only evaluate in the ideal case with ground-truth (GT) 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.

Our contributions in this chapter are as follows:

- We define a simple, yet effective unifying framework that tackles both few-shot object detection and few-shot viewpoint estimation.
- We show how to possibly leverage just a few arbitrary 3D models of novel classes to guide and boost few-shot viewpoint estimation.
- Our approach achieves state-of-the-art performance on various benchmarks.
- We propose a few-shot learning evaluation of the new joint task of object detection and view-point estimation, and provide promising results.

All the code is available at the project webpage [1].

4.2 Related Work

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

[1] http://imagine.enpc.fr/~xiaoy/FSDetView/

Few-shot Learning. Few-shot learning refers to learning from a few labeled training samples per class, which is an important problem in computer vision (Li et al., 2006; Hariharan and Girshick, 2017; Vinyals et al., 2016). One popular solution to this problem is meta-learning (Koch et al., 2015; Bertinetto et al., 2016; Andrychowicz et al., 2016; Wang and Hebert, 2016; Vinyals et al., 2016; Snell et al., 2017; Hu et al., 2018; Ravi and Larochelle, 2017; Ha et al., 2017; Lee et al., 2019; Hu et al., 2020a), where a meta-learner is designed to parameterize the optimization algorithm or predict the network parameters by "learning to learn". Instead of just focusing on the performance improvement on novel classes, some other work has been proposed for providing good results on both base and novel classes (Hariharan and Girshick, 2017; Gidaris and Komodakis, 2018; Qi et al., 2017c). While most existing methods tackle the problem of few-shot image classification, we find that other few-shot learning tasks such as object detection and viewpoint estimation are under-explored.

Object Detection. 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 (Girshick et al., 2014; He et al., 2015; Girshick, 2015; Ren et al., 2015; He et al., 2017) and FPN (Lin et al., 2017) fall into the former line of work, the YOLO series (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018) and SSD (Liu et al., 2016) 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. (Hao et al., 2018) propose to transfer a pre-trained detector to the few-shot task, while (Schwartz et al., 2019) exploit distance metric learning to model a multi-modal distribution of each object class.

More recently, (Wang et al., 2019c) propose specialized meta-strategies to disentangle the learning of category-agnostic and category-specific components in a detection model. Other approaches based on meta-learning learn a class-attentive vector for each class and use these vectors to reweight full-image features (Kang et al., 2019) or region-of-interest (RoI) features (Yan et al., 2019). Object detection with limited labeled samples is also addressed by approaches targeting weak supervision (Song et al., 2014; Bilen and Vedaldi, 2016; Diba et al., 2017; Shen et al., 2019) and zero-shot learning (Bansal et al., 2018; Rahman et al., 2018; Zhu et al., 2019), but these settings are different from ours.

Viewpoint Estimation. Deep-learning methods for viewpoint estimation follow roughly three different paths: direct estimation of Euler angles (Tulsiani and Malik, 2015; Su et al., 2015b; Mousavian et al., 2017; Kehl et al., 2017; Xiang et al., 2018; Xiao et al., 2019), template-based matching (Hinterstoißer et al., 2012b; Massa et al., 2016b; Sundermeyer et al., 2018), and keypoint detection relying on 3D bounding box corners (Rad and Lepetit, 2017; Tekin et al., 2018; Grabner et al., 2018; Oberweger et al., 2018; Pitteri et al., 2019a) or semantic keypoints (Pavlakos et al., 2017; Zhou et al., 2018).

Most of the existing viewpoint estimation methods are designed for known object categories or instances; very little work reports performance on unseen classes (Tulsiani et al., 2015; Zhou et al., 2018; Pitteri et al., 2019a; Tseng et al., 2019; Xiao et al., 2019). (Zhou et al., 2018) propose a category-agnostic method to learn general keypoints for both seen and unseen objects, while (Xiao et al., 2019) show that better results can be obtained when exact 3D models of the objects are additionally provided. In contrast to these category-agnostic methods, (Tseng et al., 2019) specifically address the few-shot scenario by training a category-specific viewpoint estimation network for novel classes with limited samples.

Instead of using exact 3D object models as (Xiao et al., 2019) (see Chapter 3), 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 (Tseng et al., 2019) require a separate meta-training procedure for each class and needs keypoint annotations in addition to the viewpoint.

4.3 Method

In this section, we first introduce the setup for few-shot object detection and few-shot viewpoint estimation (Section 4.3.1). Then we describe our common network architecture for these two tasks (Section 4.3.2) and the learning procedure (Section 4.3.4).

4.3.1 Few-shot Learning Setup

We have training samples $(x, y) \in (\mathcal{X}, \mathcal{Y})$ for our two tasks, and a few 3D shapes.

Figure 4.2: Example of class data for object detection (left) & viewpoint estimation (right).

- For 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.
- For 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.

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 4.2) to learn from using meta-learning:

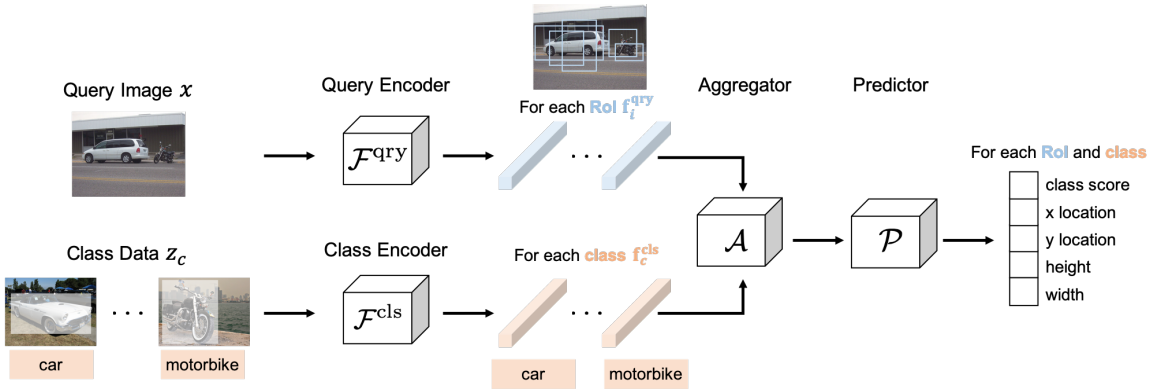
- For object detection, $Z_c = \{(x, \mathsf{mask}_i) \mid x \in \mathcal{X}, i \in \mathsf{Obj}_x\}$ is made of images x plus an extra channel with a binary mask for bounding box box_i of $i \in \mathsf{Obj}_x$.
- For viewpoint estimation, Z_c is a set of 3D models of class c.

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

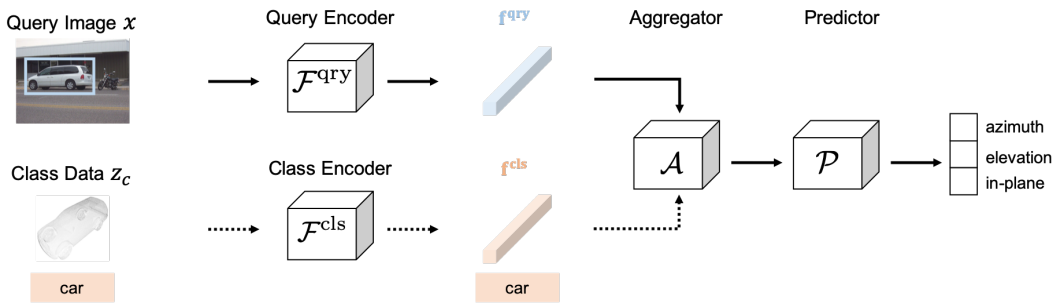
In the few-shot setting, we have a partition of the classes $C = C_{\text{base}} \cup C_{\text{novel}}$ with many samples for base classes in C_{base} and only a few samples (including shapes) for novel classes in C_{novel}. The goal is to transfer the knowledge learned on base classes with abundant samples to little-represented novel classes.

4.3.2 Network Description

Our general approach has three steps that are visualized in Figure 4.3. First, query data x and class-informative data z_c pass respectively through the query encoder $\mathcal{F}^{\text{qry}}$ and the class encoder $\mathcal{F}^{\text{cls}}$ to generate corresponding feature vectors. Next, a feature aggregation module $\mathcal{A}$ combines the query features with the class features. Finally, the output of the network is obtained by passing the aggregated features through a task-specific predictor $\mathcal{P}$:



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



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

Figure 4.3: **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, class information is extracted from a few point clouds with coordinates in normalized object canonical space, and the output of the network is the 3D pose represented by three Euler angles.

- For object detection, the predictor estimates a classification score and an object location 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.

Few-shot object detection. We adopt the widely-used Faster R-CNN (Ren et al., 2015) approach in our few-shot object detection network (see Figure 4.3(a)). The query encoder $\mathcal{F}^{\text{qry}}$ includes the backbone, the region proposal network (RPN) and the proposal-level feature alignment module. In parallel, the class encoder $\mathcal{F}^{\text{cls}}$ is here simply the backbone sharing the same weights as $\mathcal{F}^{\text{qry}}$, that extracts the class features from RGB images sampled in each class, with an extra channel for a binary mask of the object bounding box (Kang et al., 2019; Yan et al., 2019). Each extracted vector of query features $\mathrm{f}_i^{\text{qry}}$ is aggregated using operation $\{\mathcal{A}\}$ with each extracted vector of class features $\mathrm{f}_c^{\text{cls}}$ before being processed for class classification and bounding box regression:

$$\begin{aligned}(\mathsf{cls}_{i,c}, \mathsf{box}_{i,c}) &= \mathcal{P}\Big(\mathcal{A}\big(\mathrm{f}_i^{\text{qry}}, \mathrm{f}_c^{\text{cls}}\big)\Big) \\ \text{for } \ \mathrm{f}_i^{\text{qry}} &\in \mathcal{F}^{\text{qry}}(x),\ \mathrm{f}_c^{\text{cls}} = \mathcal{F}^{\text{cls}}(z_c),\ c \in C_{\text{train}}\end{aligned} \tag{4.1}$$

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 in Faster R-CNN is class-specific: the network outputs $N_{\text{train}} = |C_{\text{train}}|$ classification scores and N_{train} box regressions for each RoI. The final predictions are obtained by concatenating all the class-wise network outputs.

Few-shot viewpoint estimation. For few-shot viewpoint estimation, we rely on the recently proposed PoseFromShape architecture (Xiao et al., 2019) 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 the normalized object canonical space. The query encoder $\mathcal{F}^{\text{qry}}$ and class encoder $\mathcal{F}^{\text{cls}}$ (cf. Figure 4.3(b)) correspond respectively to the image encoder ResNet-18 (He et al., 2016) and shape encoder PointNet (Qi et al., 2017a) in PoseFromShape. By aggregating the query features and class features, we estimate the three Euler angles

using a three-layer fully-connected (FC) sub-network as the predictor:

$$\begin{aligned}(\mathsf{azi}, \mathsf{ele}, \mathsf{inp}) &= \mathcal{P}\Big(\mathcal{A}\big(\mathrm{f}^{\mathrm{qry}}, \mathrm{f}^{\mathrm{cls}}\big)\Big) \\ \text{with}\quad \mathrm{f}^{\mathrm{qry}} &= \mathcal{F}^{\mathrm{qry}}(\mathrm{crop}(\mathsf{img}(x), \mathsf{box}(x))),\ \mathrm{f}^{\mathrm{cls}} = \mathcal{F}^{\mathrm{cls}}(z_c),\ c = \mathsf{cls}(x)\end{aligned} \tag{4.2}$$

where $\mathrm{crop}(\mathsf{img}(x), \mathsf{box}(x))$ indicates that the query features are extracted from the image patch after cropping the object according to the box. Unlike the object detection making a prediction for each class and aggregating them together to obtain the final outputs, here we only make the viewpoint 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 (Xiao et al., 2019): 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). The bin size is 15° for each angle.

Feature aggregation. In recent few-shot object detection methods such as MetaYOLO (Kang et al., 2019) and Meta R-CNN (Yan et al., 2019), feature are aggregated by reweighting the query features $\mathrm{f}^{\mathrm{qry}}$ according to the output $\mathrm{f}^{\mathrm{cls}}$ of the class encoder $\mathcal{F}^{\mathrm{cls}}$:

$$\mathcal{A}(\mathrm{f}^{\mathrm{qry}}, \mathrm{f}^{\mathrm{cls}}) = \mathrm{f}^{\mathrm{qry}} \otimes \mathrm{f}^{\mathrm{cls}} \tag{4.3}$$

where $\otimes$ represents channel-wise multiplication 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 (Yan et al., 2019).)

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 (Ammirato et al., 2018; Kuo et al., 2019). The image embedding $\mathrm{f}^{\mathrm{qry}}$ itself, without any reweighting, contains relevant information too. Our aggregation thus concatenates the three forms of the query feature:

$$\mathcal{A}(\mathrm{f}^{\mathrm{qry}}, \mathrm{f}^{\mathrm{cls}}) = [\mathrm{f}^{\mathrm{qry}} \otimes \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}} - \mathrm{f}^{\mathrm{cls}}, \mathrm{f}^{\mathrm{qry}}] \tag{4.4}$$

where $[\cdot, \cdot, \cdot]$ represents channel-wise concatenation. The last part of the aggregated features in Equation (4.4) is independent of the class data. As observed experimentally

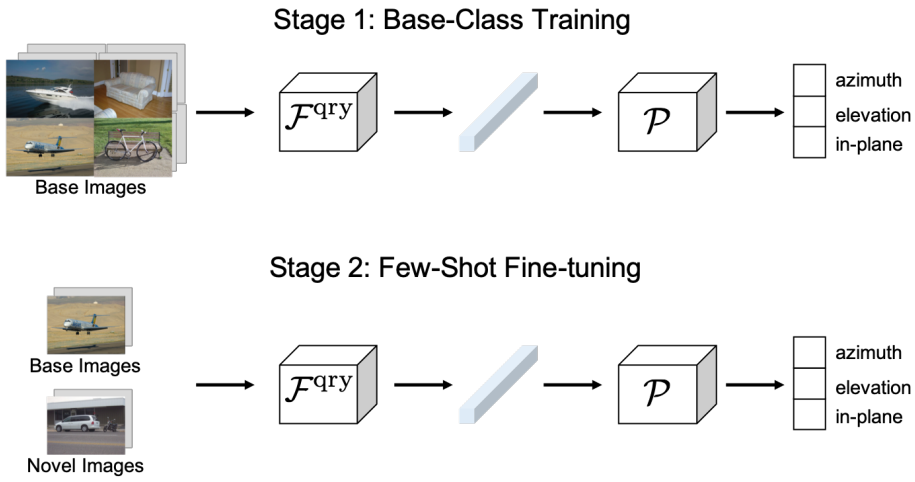


Figure 4.4: Illustration of our category-agnostic viewpoint estimation approach without using 3D models. The network is first trained on abundant labeled images of base classes (top), then fine-tuned on a balanced set of images containing both base and novel classes (bottom).

(Section 4.4.1), this partial disentanglement does not only improve few-shot detection performance, it also reduces the variation introduced by the randomness of support samples.

4.3.3 Category-agnostic Viewpoint Estimation without Shape

We also consider the case where no 3D model are 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:

$$\begin{aligned} (\mathsf{azi}, \mathsf{ele}, \mathsf{inp}) &= \mathcal{P}\left(\mathrm{f}^{\mathrm{qry}}\right) \\ \text{with} \quad \mathrm{f}^{\mathrm{qry}} &= \mathcal{F}^{\mathrm{qry}}(\mathrm{crop}(\mathsf{img}(x), \mathsf{box}(x))) \,. \end{aligned} \tag{4.5}$$

The feature extraction module is category-agnostic and all object classes share the same prediction module. Therefore, the viewpoint estimation network can fully leverage the similarities between related categories such as *bicycle* and *motorbike*.

As shown in Figure 4.4, by first training on a large base-class dataset and then

fine-tuning on a balanced dataset consisting of base and novel classes, this simple yet effective fine-tuning viewpoint estimation approach already outperforms previous methods on few-shot viewpoint estimation—we provide thorough comparison with state-of-the-art methods in Section 4.4.2.

4.3.4 Learning Procedure

The learning 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 ($C_{\text{train}} = C_{\text{base}} \cup C_{\text{novel}}$). In both phases, we optimize the network using the same loss function.

Detection loss function. Following the training protocol in Meta R-CNN (Yan et al., 2019), the query images are re-scaled such that their shorter side is of length 600 pixels, while the spatial size of support object images is normalized to 224×224 to ease the computation burden. For the objective function, we adopt the same loss function as MetaRCNN (Yan et al., 2019):

$$\mathcal{L} = \mathcal{L}_{\text{rpn}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}} + \mathcal{L}_{\text{meta}} \tag{4.6}$$

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 object classification, $\mathcal{L}_{\text{loc}}$ is a smoothed-L1 loss for object location regression, and $\mathcal{L}_{\text{meta}}$ is a cross-entropy loss encouraging class features to be diverse for different classes (Yan et al., 2019).

Viewpoint loss function. For viewpoint estimation, consistently with the PoseFromShape approach (Chapter 3), the size of the input query images is adjusted to 224×224, and the number of points included in the point cloud of each object is 2500. To predict the three Euler angles that compose the viewpoint, we adopt the same mixed classification-and-regression approaches as (Xiao et al., 2019) to compute both angular bin classification scores and offset information within each angle bin by optimizing a classification loss and a regression loss for each angle:

$$\mathcal{L} = \sum_{\theta \in \{\mathsf{azi},\mathsf{ele},\mathsf{inp}\}} \mathcal{L}^{\theta}_{\text{cls}} + \mathcal{L}^{\theta}_{\text{reg}} \tag{4.7}$$

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 smoothed-L1 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.

Class data construction. For viewpoint estimation, we make use of all the 3D models available for each class (typically less than 10) during both training stages. By contrast, the class data used in object detection requires the label of object class and location, which is limited by the number of annotated samples for novel classes. Therefore, we use large number of class data for base classes in the base training stage (typically $|Z_c| = 200$, as in Meta R-CNN (Yan et al., 2019)) and limit its size to the number of shots for both base and novel classes in the K-shot fine-tuning stage ($|Z_c| = K$).

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

$$\mathrm{f}_c^{\text{cls}} = \frac{1}{|Z_c|} \sum_{z_c \in Z_c} \mathcal{F}^{\text{cls}}(z_c). \tag{4.8}$$

This corresponds to the offline computation of all red feature vectors in Figure 4.3(a).

4.4 Results

In this section, we first evaluate on few-shot object detection benchmarks (Section 4.4.1) and few-shot viewpoint estimation benchmarks (Section 4.4.2) 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 (Kang et al., 2019; Tseng et al., 2019) and report the performance averaged over multiple experimental runs with different groups of few-shot training examples to obtain a sensible accuracy estimation (Tseng et al., 2019; Wang et al., 2020). Furthermore, we conduct a full evaluation of the joint task of few-shot object detection and viewpoint estimation on ObjectNet3D to demonstrate the generalization capacity of our method for both tasks in the few-shot regime (Section 4.4.3). For all the experiments, we run 10 trials with random support data and report the average performance.

Table 4.1: **Few-shot object detection evaluation on PASCAL VOC.** We report the mAP with IoU threshold 0.5 (AP50) under 3 different splits for 5 novel classes with a small number of shots. *Results averaged over multiple random runs.

	Novel Set 1					Novel Set 2					Novel Set 3				
Method \ Shots (K)	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
(Hao et al., 2018)	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
(Kang et al., 2019)	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
(Wang et al., 2019c)*	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
(Yan et al., 2019)*	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
(Wang et al., 2020)*	**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*	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**

4.4.1 Few-Shot Object Detection

We adopt a well-established evaluation protocol for few-shot object detection (Kang et al., 2019; Wang et al., 2019c; Yan et al., 2019) and report performance on PASCAL VOC (Everingham et al., 2012) and MS-COCO (Lin et al., 2014).

Datasets. PASCAL VOC (Everingham et al., 2012) is a small-scale object detection dataset containing 20 object categories. Following the common protocol (Redmon and Farhadi, 2017; Girshick, 2015; Ren et al., 2015), we use the test set of VOC 2007 for testing while use the train-val set of VOC 2007 and VOC 2012 for training, which results in 16,551 training images and 4,952 testing images. Among the 20 object categories, (Kang et al., 2019) introduce 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 (Lin et al., 2014) is a large-scale object detection dataset containing 80 object categories. We follow (Kang et al., 2019; Yan et al., 2019; Wang et al., 2019c) 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 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,

Table 4.2: **Few-shot object detection evaluation on MS-COCO.** We report the mean Averaged Precision and mean Averaged Recall on the 20 novel classes of COCO. *Results averaged over multiple random runs.

		Average Precision						Average Recall					
Shots	Method	0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
10	(Hao et al., 2018)	3.2	8.1	2.1	0.9	2.0	6.5	7.8	10.4	10.4	1.1	5.6	19.6
	(Kang et al., 2019)	5.6	12.3	4.6	0.9	3.5	10.5	10.1	14.3	14.4	1.5	8.4	28.2
	(Wang et al., 2019c)*	7.1	14.6	6.1	1.0	4.1	12.2	11.9	15.1	15.5	1.7	9.7	30.1
	(Yan et al., 2019)*	8.7	19.1	6.6	2.3	7.7	14.0	12.6	17.8	17.9	**7.8**	15.6	27.2
	(Wang et al., 2020)*	9.1	17.1	8.8	–	–	–	–	–	–	–	–	–
	Ours*	**12.5**	**27.3**	**9.8**	**2.5**	**13.8**	**19.9**	**20.0**	**25.5**	**25.7**	7.5	**27.6**	**38.9**
30	(Hao et al., 2018)	6.7	15.8	5.1	0.4	2.9	12.3	10.9	14.3	14.3	0.9	7.1	27.0
	(Kang et al., 2019)	9.1	19.0	7.6	0.8	4.9	16.8	13.2	17.7	17.8	1.5	10.4	33.5
	(Wang et al., 2019c)*	11.3	21.7	8.1	1.1	6.2	17.3	14.5	18.9	19.2	1.8	11.1	34.4
	(Yan et al., 2019)*	12.4	25.3	10.8	2.8	11.6	19.0	15.0	21.4	21.7	**8.6**	20.0	32.1
	(Wang et al., 2020)*	12.1	22.0	12.0	–	–	–	–	–	–	–	–	–
	Ours*	**14.7**	**30.6**	**12.2**	**3.2**	**15.2**	**23.8**	**22.0**	**28.2**	**28.4**	8.3	**30.3**	**42.1**

we classically report $AP^{0.5}$, that computes AP with a single minimum Intersection over Union (IoU) threshold at 0.5. For evaluation on MS-COCO, we use the standard MS-COCO evaluation metrics (Redmon and Farhadi, 2017; Ren et al., 2015): 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: area $< 32^2$ square pixels), M (medium: $32^2 \leq$ area $< 96^2$) and L (large: $96^2 \leq$ area).

Training details. We employ the same learning scheme as (Yan et al., 2019), 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 learning rate of 10^{-3} and another 4 epochs with a learning rate of 10^{-4}. During both training and testing, we re-scale the images such that their shorter side has 600 pixels (Ren et al., 2015). 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. In all learning stages, we generate 256 proposals for RPN training and 128 RoIs for prediction head training. Horizontal flipping is used as a standard data

Table 4.3: **Ablation study on the feature aggregation scheme.** Using the same class splits of PASCAL VOC as in Table 4.1, we measure the performance of few-shot object detection on the novel classes. We report the average and standard deviation of the AP50 metric over 10 runs. f^{qry} is the query features and f^{cls} is the class features.

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

augmentation.

Quantitative results. The results are summarized in Table 4.1 and 4.2. Our method outperforms state-of-the-art methods in most cases for the 3 different dataset splits of PASCAL VOC, and it achieves the best results on the 20 novel classes of MS-COCO, which validates the efficacy and generality of our approach. Moreover, our improvements on the difficult COCO dataset (around 3 points in mAP) is much larger than the gap among previous methods. This demonstrates that our approach can generalize well to novel classes even in complex scenarios with ambiguities and occluded objects. By comparing results on objects of different sizes contained in COCO, we find that our approach obtains a much better improvement on medium and large objects while it struggles on small objects.

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 $K \in \{3, 10\}$. The results are reported in Table 4.3. We can see that our feature aggregation scheme $[f^{qry} \otimes f^{cls}, f^{qry} - f^{cls}, f^{qry}]$ yields the best AP. In particular, although the difference $[f^{qry} - f^{cls}]$ could in theory be learned from the individual feature vectors $[f^{qry}, f^{cls}]$, 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 one of the main issues in few-shot learning.

4.4.2 Few-Shot Viewpoint Estimation

Following the few-shot viewpoint estimation protocol proposed in (Tseng et al., 2019), we evaluate our method under two settings: *intra*-dataset on ObjectNet3D (Xiang et al., 2016) (reported in Table 4.4) and *inter*-dataset between ObjectNet3D and Pascal3D+ (Xiang et al., 2014) (reported in Table 4.5).

Datasets. Pascal3D+ (Xiang et al., 2014) is a standard evaluation benchmark used in 3D pose estimation. Unlike 6D pose estimation datasets, e.g., LINEMOD (Hinterstoißer et al., 2012b), T-LESS (Hodan et al., 2017), YCB-Video (Xiang et al., 2018), that focus on dozens of objects with limited environment variations, Pascal3D+ contains 12 man-made object categories with 2k to 4k images per category capturing 36,292 objects in various uncontrolled environments, allowing the benchmarking of object pose estimation in the wild. ObjectNet3D (Xiang et al., 2016), an extension of Pascal3D+, features 100 object categories. This dataset contains 90,127 images and 201,888 objects in total. In both datasets, the number of available 3D models for each category varies from 2 to 16, which means the models only approximate the pictured objects.

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 $\Delta(R_{\text{pred}}, R_{\text{gt}}) = \| \log(R_{\text{pred}}^{\top}, R_{\text{gt}}) \|_F / \sqrt{2}$, where $\| \cdot \|_F$ is the Frobenius norm. Following previous work (Zhou et al., 2018; Tseng et al., 2019), 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. The case of predicted bounding boxes is studied in Section 4.4.3.

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

Table 4.4: **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 evaluated on ObjectNet3D.

Method	bed	bookshelf	calculator	cellphone	computer	door	f_cabinet
StarMap+F	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	0.32 / 42.2	0.76 / 15.7	0.58 / 26.8	0.59 / 22.2	0.69 / 19.2	- / -	0.76 / 15.5
(Tseng et al., 2019)	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	0.54 / 27.9	0.00 / 128	0.05 / 120	0.82 / 19.0	- / -	0.51 / 29.9	0.02 / 100
StarMap+M	0.59 / 21.5	0.00 / 136	0.08 / 117	0.82 / 17.3	- / -	0.51 / 28.2	0.01 / 100
(Tseng et al., 2019)	0.63 / 24.0	0.20 / 76.9	0.05 / **97.9**	0.77 / 17.9	- / -	0.49 / 31.6	0.21 / **80.9**
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 / 91.2
Ours w/ 3D	**0.68** / **19.4**	**0.34** / **60.1**	**0.27** / 137	**0.93** / **7.4**	**0.47** / **36.4**	**0.76** / **11.8**	**0.28** / 87.1

Method	shoe	slipper	stove	toilet	tub	wheelchair	All
StarMap+F	- / -	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	- / -	0.15 / 128	0.83 / 15.6	0.39 / 35.5	0.41 / 38.5	0.24 / 71.5	0.46 / 33.9
(Tseng et al., 2019)	- / -	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** / **92.7**	**0.91** / **9.5**	**0.69** / **17.8**	**0.65** / **16.4**	0.35 / 61.2	**0.65** / **15.6**

with a learning rate of 10^{-5}. Standard data augmentation is applied during training, such as random rotation, random flipping and color jittering.

Compared methods. For few-shot viewpoint estimation, we compare our method to MetaView (Tseng et al., 2019) and to two adaptations of StarMap (Zhou et al., 2018). More precisely, the authors of MetaView (Tseng et al., 2019) re-implemented StarMap with one stage of ResNet-18 as the backbone, and trained the network with MAML (Finn et al., 2017) 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.

Intra-dataset evaluation. We follow the protocol of (Tseng et al., 2019; Xiao et al., 2019) to split the 100 categories of ObjectNet3D into 80 base classes and 20 novel classes. As shown in Table 4.4, our model outperforms the recently proposed meta-learning-based method MetaView (Tseng et al., 2019) 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

Table 4.5: **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	0.03 / 102	0.05 / 98.8	0.07 / 98.9	0.48 / 31.9	0.46 / 33.0	0.18 / 80.8	0.22 / **74.6**
StarMap+M	0.03 / 99.2	0.08 / 88.4	0.11 / 92.2	0.55 / 28.0	0.49 / 31.0	0.21 / 81.4	0.21 / 80.2
(Tseng et al., 2019)	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 / 88.2	0.30 / 67.8	0.20 / 83.4	0.81 / 12.1	0.73 / 9.6	0.43 / 53.8	0.30 / 78.8
Ours w/ 3D	**0.21** / **72.8**	**0.33** / **64.7**	**0.25** / **78.0**	**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	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	0.29 / 36.8	0.11 / 83.5	0.44 / 42.9	0.42 / 33.9	0.64 / 25.3	0.28 / 60.5
(Tseng et al., 2019)	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**

of keypoint-based approaches (Zhou et al., 2018; Tseng et al., 2019) as they require a set of manually labeled keypoints for network training. By 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 classes and another target dataset for novel (disjoint) classes. Following the same split as MetaView (Tseng et al., 2019), 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 4.5, 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

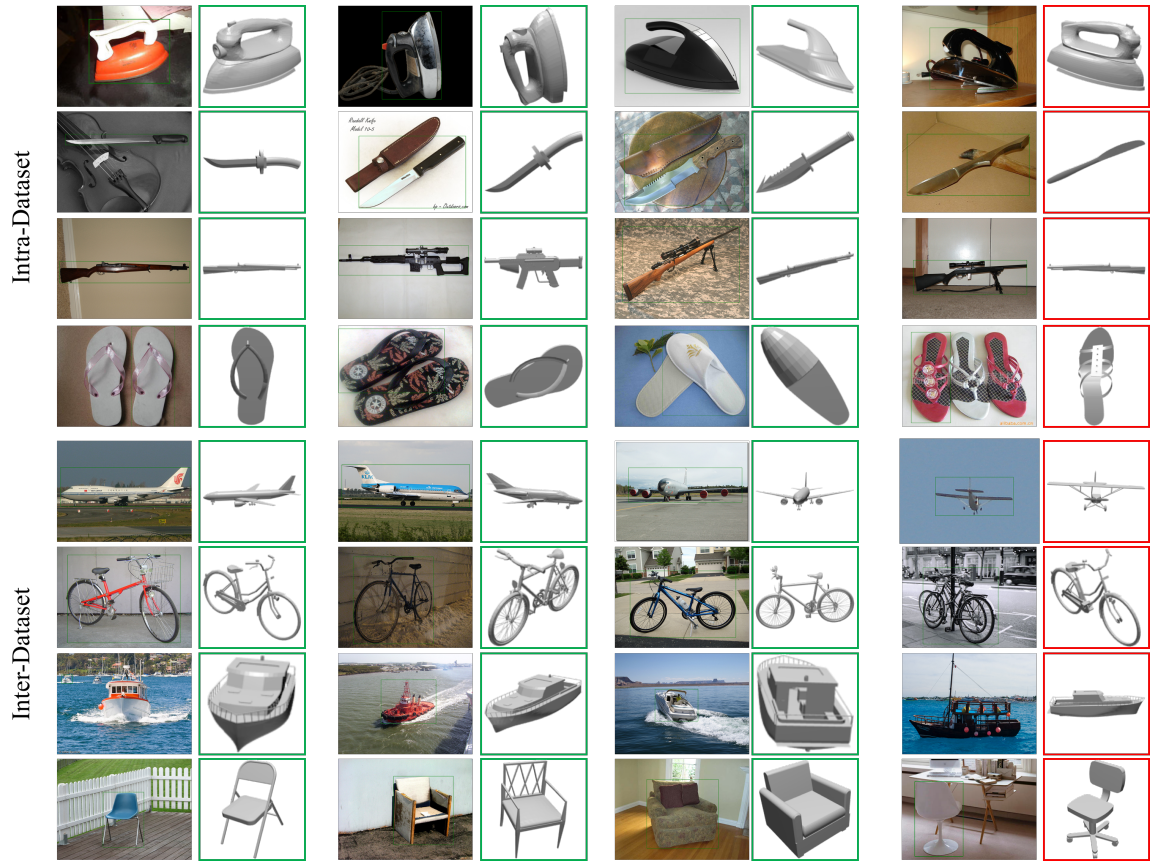


Figure 4.5: **Qualitative results of few-shot viewpoint estimation.** We visualize results on ObjectNet3D and Pascal3D+. For each category, we show three success cases (the first six columns) and one failure case (the last two columns). CAD models are shown here only for the purpose of illustrating the estimated viewpoint.

and evaluated on different datasets.

Visual results. We provide in Figure 4.5 visualizations of viewpoint estimation for novel objects on 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+. Other failure cases include heavy clutter cases (bicycle) and large shape variations between training objects and testing objects (chair).

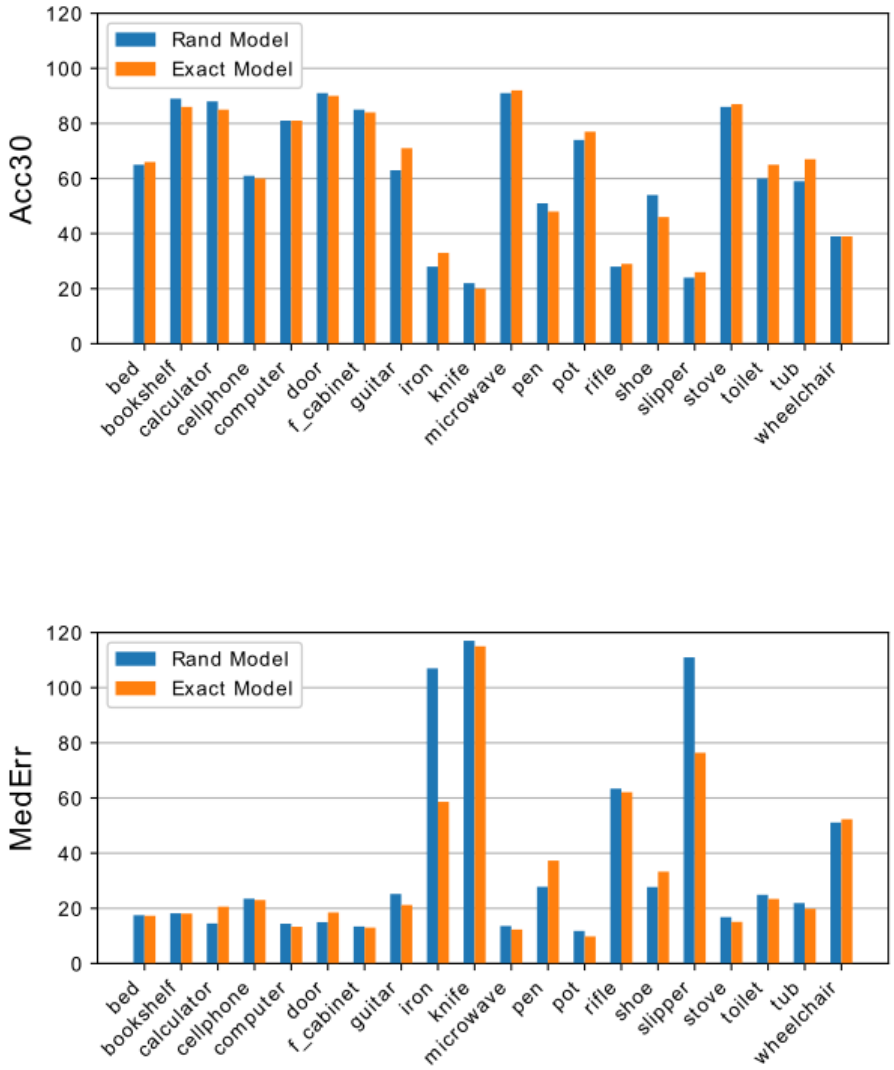


Figure 4.6: 10-shot intra-dataset viewpoint estimation performance on the 20 novel classes of ObjectNet3D: Acc30(↑) on the top and MedErr(↓) on the bottom. Training with (more or less) exact object models helps reducing the median errors for iron and slipper classes, while the overall accuracy remains similar compared to training with random models.

Robustness of shape encoding (exact vs random shape). To validate the effectiveness of our method, we report in Figure 4.6 an ablation study of training with random or (more or less) exact CAD models. By using exact CAD models during network training, we can obtain smaller median errors for predictions on irons and slippers, while the overall accuracy remains similar. This indicates that our 3D canonical shape encoding module, regardless of being fed with random or matched CAD models, is able to learn meaningful class-specific feature vectors from very few labeled samples for predicting object viewpoint on novel classes.

Different 3D model representations, if any. In Table 4.6, 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 at a set of camera locations defined on the upper hemisphere. We also use the normalized, canonical object space (Brachmann et al., 2014; Wang et al., 2019a; Grabner et al., 2019b) 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 embedding the 3D category-level shape representation 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 4.6), we note again that the 3D models can indeed help improve the viewpoint estimation 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 4.7. 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. The first thing to note is that all variants of our viewpoint estimation approach can achieve better results when the number of available annotated training samples increase from 1 to 30,

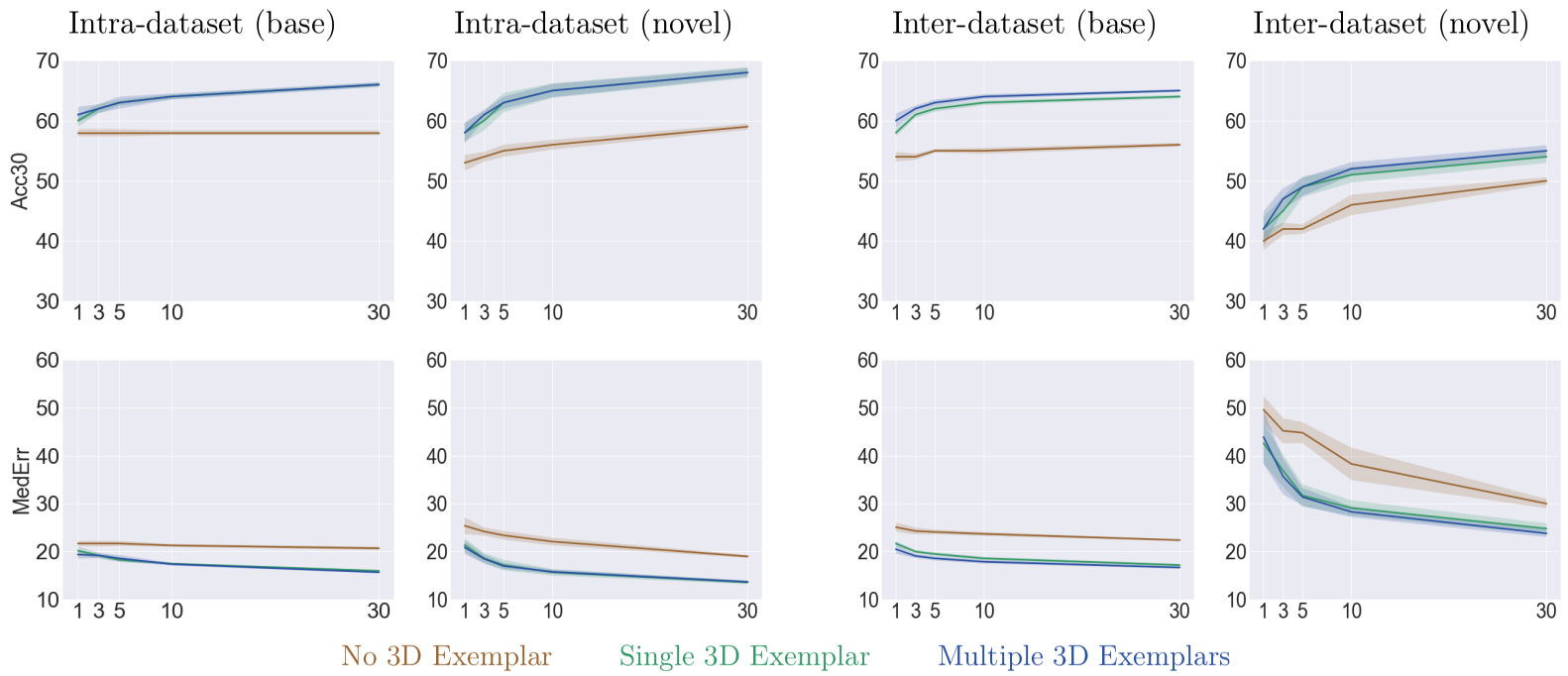
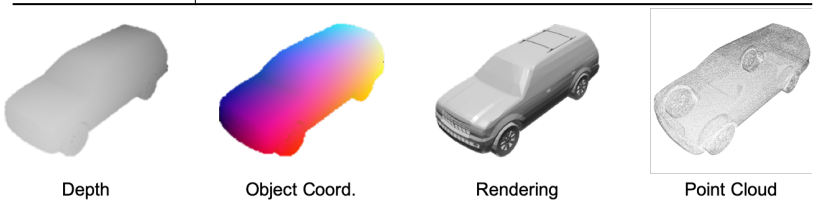


Figure 4.7: Few-shot viewpoint estimation evaluation using different number of shots. For each metric, we report the average (dark line) and standard deviation (light region) computed over 10 random experiments.

Table 4.6: Efficacy of different 3D representations, compared to using none. 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	$\mathbf{0.64} \pm 0.01$ / $\mathbf{17.5} \pm 0.18$	$\mathbf{0.65} \pm 0.01$ / $\mathbf{15.6} \pm 0.38$



especially for the objects of novel classes. 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. On the other hand, we note that the performance gap between our approach with a single 3D model per class or multiple models per class is negligible compared to the gap between using or not using 3D models. This demonstrates that even a single 3D model is sufficient to obtain a good class embedding for viewpoint estimation and adding more 3D models only leads to a minor improvement.

4.4.3 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 (Zhou et al., 2018; Tseng et al., 2019), that is not representative of actual needs. On the contrary, estimating viewpoints based on predicted detection is much more realistic and challenging.

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 by (Grabner et al., 2019a), 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 (Tseng et al., 2019) and StarMap+M, which is the best performing adaptation of StarMap (Zhou et al., 2018) (cf. Tables 4.4-4.5). 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 the viewpoint estimation variants of our's that exploits exemplar 3D models (Ours w/ 3D), as the method variant without 3D information (Ours w/o 3D) is category-agnostic.

Intra-dataset evaluation on ObjectNet3D. To experiment with this scenario, we split ObjectNet3D into 80 base classes and 20 novel classes as done in StarMap (Zhou et al., 2018), and train the object detector and viewpoint estimator using 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 top part of Table 4.7, 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, or not well centered because of imperfect predicted bounding boxes, especially for tiny objects (shoes) and ambiguous objects with similar appearances in different poses (knifes, 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

Table 4.7: 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 4.4-4.5). 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																				
Method	bed	bshelf	calc	cphone	comp	door	fcabin	guit	iron	knife	micro	pen	pot	rifle	shoe	slipper	stove	toilet	tub	wchair	All
Evaluated using ground-truth classes and ground-truth bounding boxes (viewpoint estimation)																					
StarMap+M	32	76	58	59	69	–	76	59	0	8	82	–	51	1	–	15	83	39	41	24	46
(Tseng et al., 2019)	36	76	92	58	70	–	66	63	20	5	77	–	49	21	–	7	74	50	29	27	48
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
Ours w/ 3D	64	90	90	61	86	90	86	68	34	27	93	47	76	28	49	28	91	69	65	35	65
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
Ours w/ 3D	56	75	70	47	53	64	65	75	39	8	57	22	57	15	36	24	82	64	58	24	50

	Inter-dataset evaluation on Pascal3D+												
Method	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	3	8	11	55	49	21	21	29	11	44	42	64	28
(Tseng et al., 2019)	12	8	9	71	64	22	20	39	14	29	61	58	33
Ours w/o 3D	14	30	20	81	73	43	30	51	36	49	62	77	46
Ours w/ 3D	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	14	14	10	12	73	34	19	0	20	41	64	74	31
Ours w/ 3D	15	22	15	15	74	42	16	0	30	54	70	74	35

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, 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. 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 bottom part of Table 4.7. 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 4.8 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+.

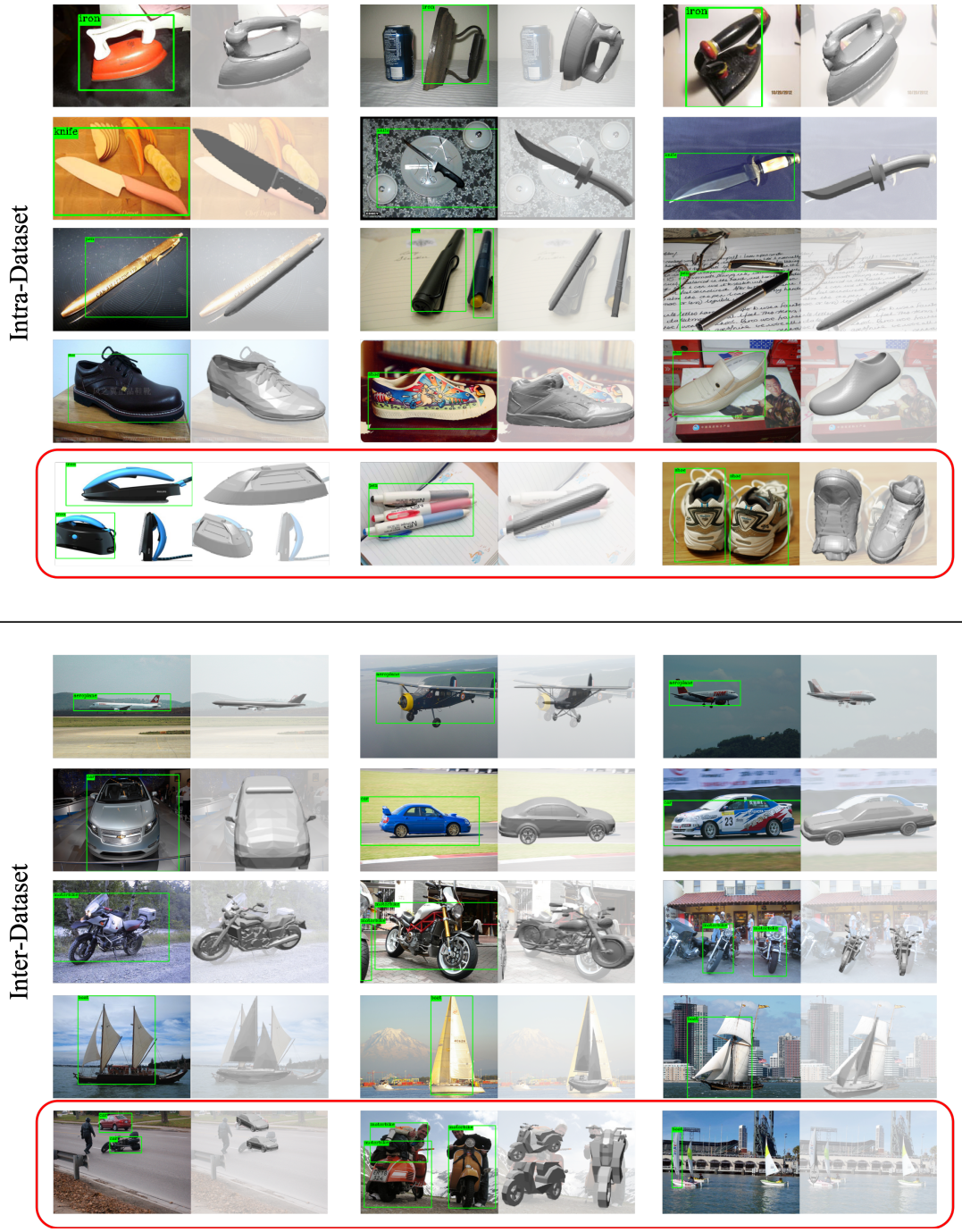


Figure 4.8: Qualitative results of joint few-shot object detection and viewpoint estimation using the predicted 2D bounding boxes given by our object detection model. We visualize results on ObjectNet3D and Pascal3D+. For each setting, we show some success cases (the first four rows) and some failure cases (the last row). 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 a single one (pen); successful detection but failed viewpoint estimation (shoe and airplane).

4.5 Conclusion

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 to the existing methods tested with ground-truth bounding boxes.

In Chapter 3 and Chapter 4, we have introduced two novel methods to estimate the 3D pose of unseen objects, with instance-level shapes or class-level shape exemplars. However, in some situations, it is hard to get the exact 3D model, or an exemplar 3D model representing the class for an arbitrary object in the wild. To handle this issue, we propose a model-free 3D pose estimation method in the next Chapter that leverages the contrastive learning in a pose-aware manner.

Chapter 5

Pose-Aware Contrastive Learning

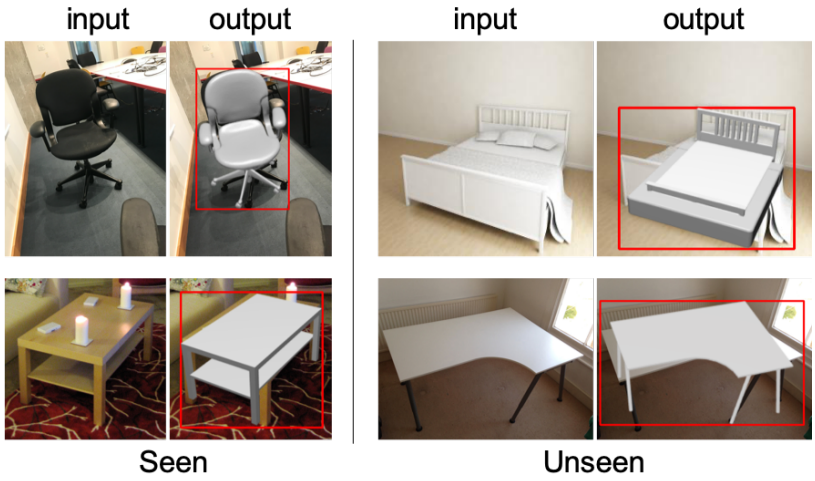


Figure 5.1: Given an RGB image picturing an object, we aim to estimate its 3D pose (viewpoint) without knowing its class or shape. It is made possible by training a model for all objects in a class-agnostic way and applying it to objects of unseen classes having similar geometries as training objects, with similar canonical frames, e.g., an unseen desk being similar to a seen table. (Red boxes are detections of a class-agnostic Mask R-CNN and the 3D models here are only used to visualize the pose.)

Abstract

Motivated by the need of estimating the 3D pose of arbitrary objects in the wild, we consider the challenging problem of class-agnostic object viewpoint estimation from images only, without CAD model knowledge. The idea is to leverage features learned on seen classes to estimate the pose for classes that are unseen, yet that share similar geometries and canonical frames with seen classes. For this, we train a direct pose estimator in a class-agnostic way by sharing weights across all object classes, and we introduce a contrastive learning method that has three main ingredients: (i) the use of pre-trained, self-supervised, contrast-based features; (ii) pose-aware data augmentations; (iii) a pose-aware contrastive loss. We experimented on Pascal3D+ and ObjectNet3D, as well as Pix3D in a cross-dataset fashion, with both seen and unseen classes. We report state-of-the-art results, including against methods that additionally use CAD models as input.

The work presented in this chapter was initially presented in:

"PoseContrast: Class-Agnostic Object Viewpoint Estimation in the Wild with Pose-Aware Contrastive Learning", Yang Xiao, Yuming Du, Renaud Marlet, In *ArXiv-2021*.

5.1 Introduction

Object 3D pose (viewpoint) estimation aims at predicting the 3D rotation of objects in images with respect to the camera. Deep learning, as well as datasets containing a variety of pictured objects annotated with 3D pose, have led to great advances in this task Tulsiani and Malik (2015); Su et al. (2015b); Mousavian et al. (2017); Wang et al. (2019a); Liao et al. (2019).

However, they mainly focus on class-specific estimation for few categories, and they mostly evaluate on ground-truth bounding boxes. It is an issue when encountering objects of unseen classes, for which no training data was available and no bounding box is given, which is a likely circumstance for a number of robot applications in uncontrolled settings.

Our goal is to address this issue. Given training data for some known classes (images with bounding boxes of multiple objects, class labels and 3D pose annotations), we want to detect and estimate the 3D pose of objects of unknown classes, given only an RGB image as input (Fig. 5.1), vs also using CAD models of objects as some methods do Xiao et al. (2019); Pitteri et al. (2020).

This new task relies on two assumptions. First, it applies to unseen classes that share similarities with seen classes. For example, one may expect to orient an unseen bed when trained on seen chairs and sofas, but not a wrench.

The other assumption is that similar classes have a consistent canonical pose, i.e., have aligned similarities (Figs. 5.2 and 5.5). It is somehow a weak assumption, satisfied by all datasets we know of, probably because many objects are used consistently w.r.t. verticality, and feature a notion of left- and right-hand sides, or at least a main vertical symmetry plane, which is enough to define a "natural" canonical frame, possibly up to symmetry. Besides, if similar classes in a training set have inconsistent canonical poses, they can be normalized by a systematic rotation; no 3D shape is needed for that. In this first work, we only consider the general case, disregarding the different forms of symmetry.

Overview. To detect arbitrary objects and estimate their pose, although not in training data, we use a class-agnostic approach for both object detection and pose estimation.

Approaches like Grabner et al. (2018); Zhou et al. (2018); Pitteri et al. (2019a) have already demonstrated the effectiveness of this setting. They detect 2D keypoints regardless of the class of the object, estimate 2D-3D keypoint correspondences, and

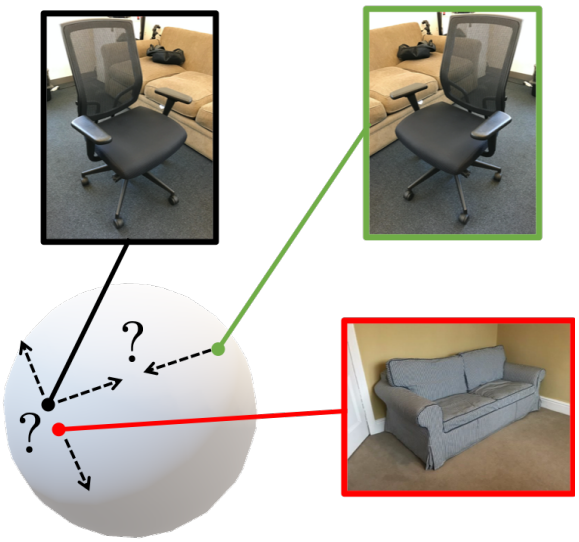


Figure 5.2: **Pose-Aware Contrastive Loss.** In usual self-supervised contrastive training, the network learns to pull together in feature space the **query** (e.g., chair) and a **positive** variant (e.g., flipped image), while pushing apart the **query** from **negatives** (different objects, e.g., sofa), ignoring pose information. Instead, we exclude flipped **positives**, whose pose actually differ from the **query**, and do not push apart **negatives** with similar poses (e.g., sofa).

use a PnP algorithm Lepetit et al. (2009) to compute the pose. But besides being indirect, these methods need a suitable design of class-agnostic keypoints on various object geometries. In contrast, our approach estimates the 3D pose directly from the image embeddings, without any intermediate representation.

Others assume a 3D model of the object is given at test time (sometimes also at training time) Xiao et al. (2019); Park et al. (2020); Pitteri et al. (2020); Dani et al. (2021), either provided by a human or retrieved automatically by an algorithm, which is hard due to the image-shape domain gap and to the number of classes to discriminate Su et al. (2015a); Massa et al. (2016b); Xiang et al. (2016), and it is limited by the database of possible 3D models to handle. In comparison, our method relies only on RGB images both at train and test time, without any CAD model as input.

To that end, we train a class-agnostic pose estimator by sharing weights across all object classes. And we propose a contrastive-learning approach to learn geometry-aware image embeddings that are optimized for pose estimation.

Recent contrastive-learning approaches create discriminative image features by learning to distinguish pairs of *identical objects* with different appearances thanks

to a synthetic transformation (positives), from pairs of *different objects* (negatives). Inspired by image-level discrimination He et al. (2020); Chen et al. (2020a); Xiao et al. (2021), we adapt the common contrastive loss InfoNCE Oord et al. (2018) so that it discriminates poses rather than categories: we propose PoseNCE, a pose-aware contrastive loss that pushes away in latent space the image features of objects having different poses, ignoring the class of these objects as we aim at class-agnostic pose estimation (see Fig. 5.2). Besides, departing from the binary separation between positives and negatives in classical InfoNCE, PoseNCE takes into account the level of pose difference between two objects as a weighting term to reduce or stress the negativeness of a pair, regardless of the class (see Fig. 5.5).

Concretely, we use both an angle loss and a contrastive loss. We also curate the contrastive learning transformations to distinguish pose-variant data augmentations, e.g., horizontal flip, and pose-invariant data augmentations, e.g., color jittering. The former is used to actually augment the dataset, while the latter is used to create similar variants to construct positive and negative pairs. And rather than training from scratch on available datasets, that are relatively small, we initialize our network with a contrastive model trained on a large dataset in a self-supervised way.

Last, we propose a class-agnostic approach for both object detection *and* pose estimation. For this, we train a Mask R-CNN in a class-agnostic way for generic object detection, and pipeline it with our pose estimator, thus addressing the coupled problem of generic object detection and pose estimation for unseen objects. It is a more realistic setting w.r.t. existing class-agnostic pose estimation methods, that only evaluate in the ideal case of ground-truth bounding boxes.

Our contributions in this chapter as as follows:

- We define a new task suited for uncontrolled settings: class-agnostic object 3D pose estimation, possibly coupled and preceded by class-agnostic detection.

- We propose a contrastive-learning approach for class-agnostic pose estimation, which includes a pose-aware contrastive loss and pose-aware data augmentations.

- We report state-of-the-art results on 3 datasets, including against methods that also require shape knowledge.

5.2 Related Work

Class-Specific Object Pose Estimation. While instance-level 3D object pose estimation has long been studied in both robotic and vision communities (Hinterstoißer et al., 2012b; Brachmann et al., 2014; Kehl et al., 2017; Rad and Lepetit, 2017; Tekin et al., 2018; Rad et al., 2018; Xiang et al., 2018; Sundermeyer et al., 2018; Oberweger et al., 2018; Labbé et al., 2020), class-level pose estimation has developed more recently thanks to learning-based methods (Su et al., 2015b; Tulsiani and Malik, 2015; Tulsiani et al., 2015; Mousavian et al., 2017; Kundu et al., 2018; Wang et al., 2018; Grabner et al., 2018, 2019a; Wang et al., 2019a; Zhou et al., 2018; Tseng et al., 2019). These methods can be roughly divided into two categories: direct pose estimation methods that regress 3D orientations directly (Tulsiani and Malik, 2015; Su et al., 2015b; Mousavian et al., 2017; Wang et al., 2018; Xiao et al., 2019), and keypoint-based methods that predict 2D locations of 3D keypoints (Grabner et al., 2018, 2019a; Wang et al., 2019a; Zhou et al., 2018; Tseng et al., 2019).

Still, annotating 3D poses for objects in the wild is a tedious process of searching best-matching CAD models and aligning them to images (Xiang et al., 2014, 2016). This cannot easily scale to large numbers of objects from many classes. While class-specific methods achieve high accuracies on supervised classes, how to generalize beyond training data remains an important, yet under-explored problem.

Class-Agnostic Object Pose Estimation. To circumvent the problem of limited labeled object classes, a few class-agnostic pose estimation methods have been proposed in recent years (Grabner et al., 2018; Zhou et al., 2018; Xiao et al., 2019; Dani et al., 2021; Pitteri et al., 2019a). In contrast to class-specific methods that build an independent prediction branch for each object class, agnostic methods estimate the object pose without knowing its class *a priori*, which is enabled by sharing model weights across all object classes during training.

Ge et al. (2020) trains on multiple views of the same object instance on a turntable. (Grabner et al., 2018; Pitteri et al., 2019a) use the 3D bounding box corners as generic keypoints for class-agnostic object pose estimation. However, (Grabner et al., 2018) only reports performance on seen classes and (Pitteri et al., 2019a) focuses on cubic objects with simple geometric shape. Instead of using a fixed set of keypoints for all objects, (Zhou et al., 2018) propose a class-agnostic keypoint-based approach combining a 2D keypoint heatmap and 3D keypoint locations in the object canonical

frame. These methods are robust on textured objects but fail with heavy occlusions and tiny or textureless objects. In contrast, our method ignores keypoints, it directly infers a pose and is less sensitive to texturelessness.

Rather than only relying on RGB images, another group of class-agnostic pose estimation methods (Xiao et al., 2019; Dani et al., 2021) makes use of 3D models at test time to adapt to objects unseen at training time. (Xiao et al., 2019) aggregates 3D shape and 2D image information for arbitrary objects, representing 3D shapes as multi-view renderings or point clouds. (Dani et al., 2021) propose a lighter version of (Xiao et al., 2019) encoding the 3D models into graphs using node embeddings (Grover and Leskovec, 2016). Pitteri et al. (2020) matches local images embeddings with local 3D embeddings, then use RANSAC and PnP algorithms to recover an object from a database of CAD models, and a pose. In contrast, we need no 3D shape, neither at training nor at testing time.

Pose Loss. 3D pose dissimilarity has been measured indirectly, e.g., with a distance on reprojected features such as keypoints (see above), or directly on pose parameters. In the latter case, the chosen representation and penalty may yield more or less artefacts due to, e.g., discontinuities in the parameterization (Euler angles, quaternions Zhou et al. (2019)), gimbal lock Grassia (1998), anti-podal symmetry (quaternions), non-uniform parameter distributions, classification discretization Tulsiani and Malik (2015); Su et al. (2015b); Elhoseiny et al. (2016), single-mode analysis as with regression Osadchy et al. (2007); Penedones et al. (2012); Massa et al. (2016a), or parameter-space biases when penalizing with the L2-norm of the difference of pose parameters, including with the exponential twist representation Zhu et al. (2017). We use a combination of classification and regression Mousavian et al. (2017); Güler et al. (2017); Mahendran et al. (2018); Li et al. (2018a) of Euler angles similar to Xiao et al. (2019) (offset regression from bin center), which better separates modes in case of pose ambiguities, but we penalize a geodesic distance on the unit sphere rather than the Euclidean distance of parameters, which does not have dimensional biases.

Contrastive Learning. Instead of designing pretext tasks for unsupervised learning (Doersch et al., 2015; Noroozi and Favaro, 2016; Zhang et al., 2016; Gidaris et al., 2018), powerful image features can be learned by contrasting positive and negative pairs (Wu et al., 2018; Oord et al., 2018; Tian et al., 2020; Misra and Maaten, 2020;

Caron et al., 2020; He et al., 2020; Chen et al., 2020c,a,b; Khosla et al., 2020). Among the various forms of the contrastive loss function (Hadsell et al., 2006; Wang and Gupta, 2015; Hjelm et al., 2019; Wu et al., 2018; Oord et al., 2018), InfoNCE (Oord et al., 2018) has become a standard pick in many methods.

While most contrastive learning approaches work in the unsupervised setting, (Khosla et al., 2020) extends the approach to full supervision by leveraging label information. Considering the class label of training examples, features belonging to the same class are pulled together while, simultaneously, features from different classes are pushed apart.

Similar to (Khosla et al., 2020), we also propose a contrastive loss that works in the fully-supervised setting. However, instead of focusing on semantic label information, we design it for our geometric task, taking into account the pose distance between different examples. Moreover, we also curate data augmentations as advocated in (Xiao et al., 2021), leaving out those that would be harmful for our pose estimation task.

Besides requiring 3D shapes at training time and operating on RGB-D data, Balntas et al. (2017) is not pose-aware: in the InfoNCE spirit, it creates positive pairs from the same known shape model and negative pairs from known different shapes, ignoring pose. Besides, it favors features whose L2-distance is *equal to* their pose L2-distance, which is a heavy burden for feature learning, especially for objects with large shape variations. In comparison, we simply contrast features w.r.t. pose dissimilarity. Wohlhart and Lepetit (2015), which operates in a class-specific way and also requires known 3D shapes or at least multiple views or renderings of each object, uses a triplet loss whose formulation can be related to our more general PoseNCE loss, but it does not take into account the level of pose dissimilarity nor pose-aware data augmentation.

Coupled Detection and Pose Estimation. Very few works consider the realistic scenario of detecting unknown objects in images *and* inferring their pose. Pitteri et al. (2020) trains a class-agnostic Mask R-CNN and pipelines it with a pose estimator, as we do, but it applies to industrial objects and requires knowing the 3D shapes, including for novel instances. Ge et al. (2020), which trains with objects on a turntable, does not do any detection but somehow also applies to ImageNet, i.e., with well-centered, single-object images. None of these methods is thus applicable to objects in the wild. And although Grabner et al. (2018) predicts a 3D box size

(not location) for PnP reprojection, it operates on ground-truth 2D bounding boxes. We can only compare in the class-specific detection and pose estimation setting Wang et al. (2018); Grabner et al. (2019a) and, in the class-agnostic setting, against methods also requiring an input 3D shape Xiao et al. (2019).

5.3 Method

Given an RGB image I containing an object at a given (known or detected) image location, we aim to estimate the 3D pose $\mathbf{R}$ of the object with no prior knowledge of its class or shape. To that end, we crop the image region containing the object, encode it to produce class-agnostic features, from which the object 3D pose is directly predicted (Fig. 5.3).

3D Pose Parameterization. To predict the 3D rotation matrix $\mathbf{R}$ of the pictured object, we decompose it into three Euler angles as in (Su et al., 2015b; Xiao et al., 2019): azimuth azi, elevation ele, and inplane rotation inp, with $\mathsf{azi}, \mathsf{inp} \in [-\pi, \pi)$ and $\mathsf{ele} \in [-\pi/2, \pi/2]$.

Recent work on pose estimation shows that a higher performance can be achieved with a formulation mixing both angular bin classification and within-bin offset regression (cf. Chapter 3 and Chapter 4). Concretely, we split each Euler angle $\theta \in \{\mathsf{azi}, \mathsf{ele}, \mathsf{inp}\}$ uniformly into discrete bins i of size B ($= \pi/12$ in our experiments). The network outputs bin classification scores $p_{\theta,i} \in [0, 1]$ and offsets $\delta_{\theta,i} \in [0, 1]$ within the bin.

Angle Loss. We use a cross-entropy loss for angle bin classification and a smooth-L1 loss for bin offset regression:

$$\mathcal{L}_{\text{ang}} = \sum_{\theta \in \{\mathsf{azi},\mathsf{ele},\mathsf{inp}\}} \mathcal{L}_{\text{cls}}(\text{bin}_\theta, p_\theta) + \lambda \, \mathcal{L}_{\text{reg}}(\text{offset}_\theta, \delta_\theta) \tag{5.1}$$

where bin_θ is the ground-truth bin and offset_θ is the offset for angle θ. The relative weight λ is set to 1 in our experiments. The final prediction for angle θ is obtained as:

$$\widehat{\theta} = (j + \delta_{\theta,j})B \quad \text{with} \quad j = \arg\max_i p_{\theta,i} \tag{5.2}$$

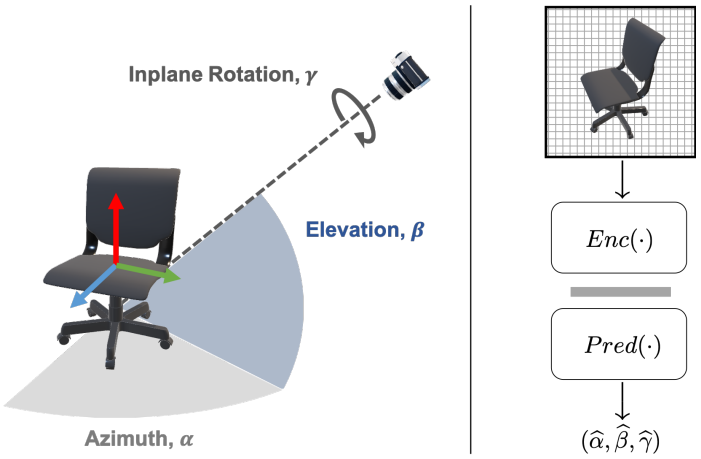


Figure 5.3: **Pose Parameters** (left). **Network Architecture** (right): from an image crop, the encoder *Enc* produces an embedding, which is given to the predictor *Pred* to produce pose angles.

where $i \in [-12..11]$ for azi, inp, and $i \in [-6..5]$ for ele. The angle loss is complemented by a contrastive loss (cf. Section 5.3).

Network Architecture. The architecture of our network is depicted in Figure 5.3 (right). It consists of two modules: an image encoder $Enc(\cdot)$ and a pose predictor $Pred(\cdot)$.

For feature extraction, we use a standard CNN, namely ResNet-50. We crop the input image to the targeted object and pass it through the encoder network until the max-pooling layer. It provides a 2048-dimension feature vector.

We then pass the image embedding through the pose predictor, which is a multi-layer perceptron (MLP) with 3 hidden layers of size 800-400-200, each followed by batch normalization and ReLU activation. Contrary to class-specific methods (Su et al., 2015b; Tulsiani and Malik, 2015; Liao et al., 2019; Mousavian et al., 2017) that use one prediction branch per class, we use a single prediction branch for all objects.

Contrastive Features. Datasets of images with pose annotations are scarce and small. (One of the reasons is probably that pose is much harder to annotate than class, especially for images in the wild.) It makes it difficult to learn a high-quality pose estimator. Rather than learning a network from scratch, as most other methods do, or from an initial ImageNet classifier, whose bias is not particularly suited for pose estimation, we initialize our predictor using a pre-trained contrast-based network. We

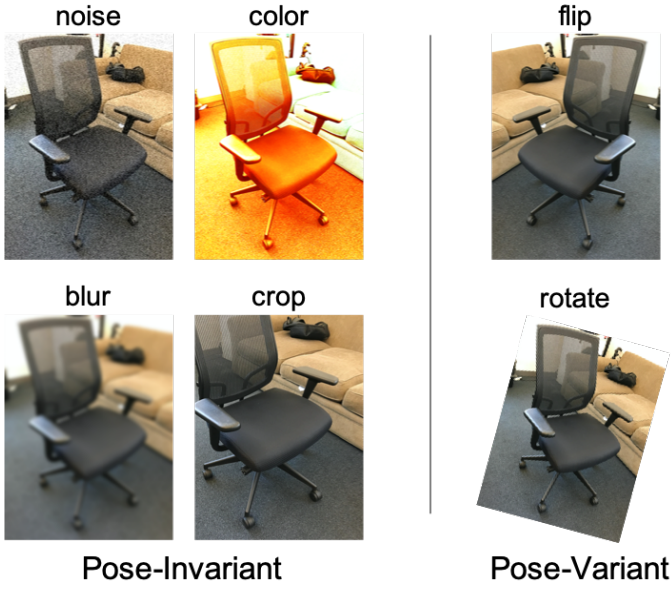


Figure 5.4: **Pose-Aware Data Augmentations:** while *pose-invariant* data augmentations do not alter the pose, *pose-variant* augmentations modify it and cannot be used as positives.

show that it plays a significant role in our high performance (cf. Section 5.4.3).

Self-Supervised Contrastive Loss. In self-supervised contrastive learning (Chen et al., 2020a; He et al., 2020), the contrastive loss serves as an unsupervised objective for training an image encoder that maximizes agreement between different transformations of the same sample, while minimizing the agreement with other samples. Concretely, we consider a batch $(I_k)_{k\in[1..N]}$ of training samples, transformed into $(\tilde{I}_k)_{k\in[1..N]}$ by data augmentation, and encoded as $\mathrm{f}_k = Enc(\tilde{I}_k)$. For any index $k^+ \in [1..N]$, we consider an alternative augmentation $\tilde{I}_q$ of I_{k^+} (the query), with embedding $\mathrm{f}_q = Enc(\tilde{I}_q)$, and we separate the positive pair (q, k^+) from the negative pairs $(q, k^-)_{k^-\in[1..N]\setminus\{k^+\}}$ with the following infoNCE loss:

$$\mathcal{L}_{\text{infoNCE}} = -\log \frac{\exp(\mathrm{f}_q \cdot \mathrm{f}_{k^+}/\tau)}{\sum_{k\in[1..N]} \exp(\mathrm{f}_q \cdot \mathrm{f}_k/\tau)} \tag{5.3}$$

where τ is a temperature parameter (Chen et al., 2020a) (0.5 in experiments).

Pose-Aware Data Augmentations. As illustrated in Figure 5.4, we divide data augmentations into two categories. *Pose-invariant* augmentations transform the im-

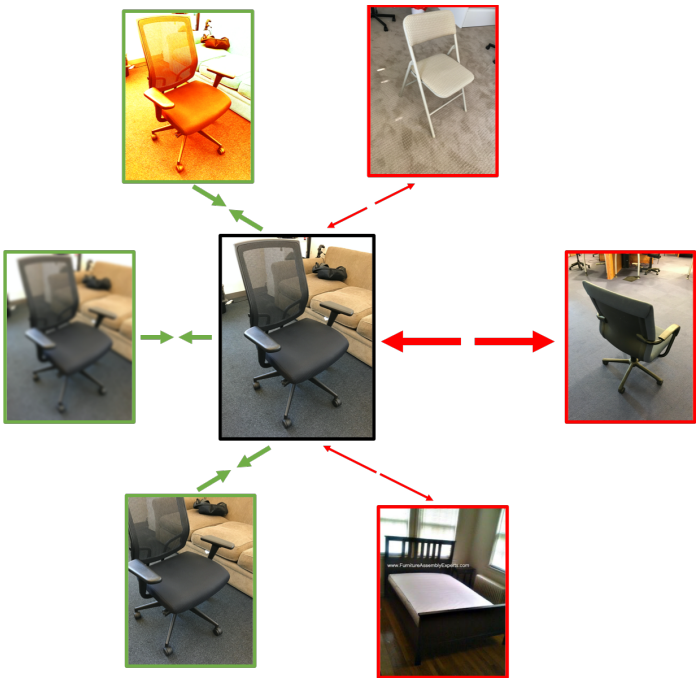


Figure 5.5: **Pose-Aware Contrast.** Instead of treating all negatives (right images) equally, as for positives (left images), we give more weight to **negatives** with a large rotation (rightmost) and less to those with a small rotation, *regardless of their semantic class.*

age without changing the 3D pose of objects: color jittering, blur, crop, etc. On the contrary, *pose-variant* augmentations change the 3D pose at the same time: rotation and horizontal flip. More precisely, an image rotation of angle ϕ corresponds to an in-plane rotation of angle $\mathsf{inp} + \phi$ for the object, and a horizontal image flip corresponds to a change of sign of azimuth azi and in-plane rotation inp. (We assume mirror-imaged objects are realistic objects with identical canonical frame.) In our experiments, ϕ varies in $[-15°, 15°]$.

Unlike self-supervised learning methods that make use of all data augmentation techniques at the same time, we distinguish to augmentation times: at batch creation time and at contrast time. At *batch creation time*, we only apply pose-variant data augmentations, i.e., a small rotation or a horizontal flip, and update the object pose information accordingly. At *contrast time*, i.e., when creating *positive* and *negative* pairs, we only apply pose-invariant data augmentations. The latter is motivated by (Xiao et al., 2021): a blind use of any data augmentation could be harmful.

Pose-Aware Contrastive Loss. The contrastive loss in Eq. (5.3) is designed for unsupervised learning with no annotation involved. While efficient for learning instance-discriminative image embeddings, it is not particularly suited for contrasting geometric cues towards pose estimation: the query embedding is contrasted away from the embeddings of negative samples even if their pose is identical or similar to the pose of the query object. While the case of different views of identical or similar instances can be disregarded in usual contrastive learning because of its practical rarity, similar and even identical poses are common in a single batch. What we want, instead of contrasting the object semantics, is to learn pose-variant image features.

We thus introduce a new pose-aware contrastive loss, illustrated in Figure 5.5. It takes into account the level of sample negativeness: the larger the pose difference, the higher the weight in the loss. Concretely, for each pair (q, k), we compute a normalized distance $\mathrm{d}(\mathbf{R}_q, \mathbf{R}_k) \in [0, 1]$ between the associated 3D pose rotations $\mathbf{R}_q, \mathbf{R}_k$ and we use it as a weight in our *pose-aware contrastive loss*:

$$\mathcal{L}_{\text{poseNCE}} = -\log \frac{\exp(\mathrm{f}_q \cdot \mathrm{f}_{k^+}/\tau)}{\sum_{k \in [1..N]} \mathrm{d}(\mathbf{R}_q, \mathbf{R}_k) \exp(\mathrm{f}_q \cdot \mathrm{f}_k/\tau)} \tag{5.4}$$

with $\mathrm{d}(\mathbf{R}_q, \mathbf{R}_{k^+}) = 0$ as $\mathbf{R}_q = \mathbf{R}_{k^+}$. Our distance is defined as the normalized angle difference between the rotations, which is akin to a geodesic distance on the unit sphere:

$$\begin{aligned} \mathrm{d}(\mathbf{R}_q, \mathbf{R}_k) &= \Delta(\mathbf{R}_q, \mathbf{R}_k)/\pi \quad \text{with} \\ \Delta(\mathbf{R}_q, \mathbf{R}_k) &= \arccos\Big(\frac{\mathrm{tr}(\mathbf{R}_q^{\mathrm{T}} \mathbf{R}_k) - 1}{2}\Big) \end{aligned} \tag{5.5}$$

The total loss is the sum of the angle loss and the contrastive loss:

$$\mathcal{L} = \mathcal{L}_{\text{ang}} + \kappa \, \mathcal{L}_{\text{poseNCE}} \tag{5.6}$$

The relative weight κ is set to 1 in our experiments.

5.4 Results

In this section, we introduce our experimental setup, and then present results on three commonly used benchmarks: we report performance on supervised (seen) classes of Pascal3D+ (Xiang et al., 2014) and novel (unseen) classes of Pix3D (Sun et al., 2018), as well as a few-shot evaluation on ObjectNet3D (Xiang et al., 2016). Moreover, we

Dataset	year	# classes	# img train / val*	annot.
Pascal3D+ (Xiang et al., 2014)	2014	12	28,648 / 2,113	+
ObjectNet3D (Xiang et al., 2016)	2016	100	52,048 / 34,375	++
Pix3D (Sun et al., 2018)	2018	9	0 / 5,818	+++

Table 5.1: **Experimented Datasets:** images of objects in the wild, with different qualities of pose annotation due to aligned shapes. *Only non-occluded and non-truncated objects, as done usually.

provide an ablation study of individual components of our method. Last, we discuss the limitations.

5.4.1 Experimental Setup

Implementation Details. Our experiments are coded using PyTorch. We use parameters $\lambda = 1$, $\kappa = 1$, $\tau = 0.5$, and the transformation rotation ϕ varies in $[-15°, 15°]$, i.e., $[-\frac{\pi}{24}, \frac{\pi}{24}]$. We train our networks end-to-end using Adam optimizer with a batch size of 32 and an initial learning rate of 1e-4, which we divide by 10 at 80% of the training phase. Unless otherwise stated, we train for 15 epochs, which takes less than 2 hours on a single V100-16G GPU.

Training Details of Detection Network. We use a class-agnostic Mask R-CNN (He et al., 2017) with a ResNet-50-FPN backbone (Lin et al., 2017) as our instance segmentation network. The Mask R-CNN is trained on COCO dataset (Lin et al., 2014), which contains 80 classes and 115k training images. We use the open source repo of Mask R-CNN (Massa and Girshick, 2018) and follow the training setting of (He et al., 2017), except that we adopt the class-agnostic architecture, where all 80 classes are merged into a single "object" category. Our backbone network is initialized with weights pre-trained on ImageNet (Deng et al., 2009). During training, the shorter edge of images are resized to 800 pixels. Each GPU has 4 images and each image has 512 sampled RoIs, with a ratio of 1:3 of positives to negatives. We train our Mask R-CNN for 90k iterations. The learning rate is set to 0.02 at the beginning and is decreased by 10 at the 60k-th and 80k-th iteration. We use a weight decay of 0.0001 and momentum of 0.9. The entire training is carried out on 4 Nvidia RTX 2080Ti GPUs. During the training, mixed precision training is used to reduce memory consumption and accelerate training.

Datasets. We experimented with three commonly used datasets for benchmarking object pose estimation in the wild (see Table 5.1). While they all feature a variety of objects and environments, Pascal3D+ (Xiang et al., 2014) contains only the 12 rigid classes of PASCAL VOC 2012 (Everingham et al., 2012), with approximate poses due to coarsely aligned 3D models at annotation time. ObjectNet3D distinguishes 100 classes in a subset of ImageNet (Deng et al., 2009), with more accurate poses as more and finer shapes were used for annotation. Recently, Pix3D (Sun et al., 2018) proposes a smaller but even more accurate dataset with pixel-level 2D-3D alignment using exact shapes; although it only features 9 classes, two of them ('tool' and 'misc') do not appear in Pascal3D+ nor ObjectNet3D. All methods are tested only non-occluded and non-truncated objects, as done in the other publications.

Evaluation Metrics. Unless otherwise stated, ground-truth object bounding boxes are used in all experiments. We compute the most common metrics (Tulsiani and Malik, 2015; Su et al., 2015b): Acc30 is the percentage of estimations with rotation error less than 30 degrees; MedErr is the median angular error in degrees.

5.4.2 Main Results

Upper Bound: Performance on Seen Classes. To check our performance before considering unseen classes, we first evaluate on seen classes. We follow the common protocol (Grabner et al., 2018; Zhou et al., 2018) to train our model on the train split of Pascal3D+ (Xiang et al., 2014) and test it on the val split. Both train and val splits share the same 12 object classes. In Table 5.2 we compare with state-of-the-art class-agnostic object pose estimation methods, using ground-truth bounding boxes. As we leverage on contrast-based features, we use the available MOCOv2 pre-trained ResNet-50 model Chen et al. (2020c). But no MOCO pre-trained ResNet-18 is available for comparison (see also Table 5.7).

For most categories, our class-agnostic approach consistently outperforms other class-agnostic methods (Grabner et al., 2018; Zhou et al., 2018), including those that leverage a 3D shape as additional input (Xiao et al., 2019; Dani et al., 2021). In particular, our direct pose estimation method achieves a clear improvement for the class 'chair', which features a higher variety and geometric complexity than other classes. It suggests that keypoint-based methods as (Grabner et al., 2018; Zhou et al., 2018) may fail to capture detailed shape information for accurate 2D-3D correspondence

	Method	w/ 3D	PnP	Backbone	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
MedErr ↓	Grabner et al. Grabner et al. (2018)		✓	ResNet-50	10.9	**12.2**	23.4	9.3	3.4	5.2	15.9	16.2	12.2	11.6	6.3	11.2	11.5
	StarMap Zhou et al. (2018)		✓†	ResNet-18 *	10.1	14.5	30.3	9.1	3.1	6.5	11.0	23.7	14.1	11.1	7.4	13.0	12.8
	3DPoseLite Dani et al. (2021)	✓		ResNet-18	–	–	–	–	–	–	–	–	–	–	–	–	13.4
	PoseFromShape (Chapter 3)	✓		ResNet-18	11.1	14.4	22.3	7.8	3.2	5.1	12.4	13.8	11.8	**8.9**	**5.4**	**8.8**	10.4
	PoseFromShape (Chapter 3)	✓		ResNet-50	10.9	14.5	21.5	7.5	3.3	5.0	11.2	14.2	11.6	9.2	5.5	9.0	10.3
	PoseContrast (ours)			ResNet-50	**10.0**	13.6	**18.3**	**7.2**	**2.8**	**4.6**	**9.8**	**9.2**	**11.5**	11.0	5.6	11.6	**9.6**
Acc30 ↑	Grabner et al. Grabner et al. (2018)		✓	ResNet-50	0.80	0.82	0.57	0.90	**0.97**	0.94	0.72	0.67	0.90	0.80	**0.82**	**0.85**	0.81
	StarMap Zhou et al. (2018)		✓†	ResNet-18 *	0.82	**0.86**	0.50	0.92	**0.97**	0.92	0.79	0.62	0.88	**0.92**	0.77	0.83	0.82
	3DPoseLite Dani et al. (2021)	✓		ResNet-18	0.80	0.82	0.58	0.93	0.96	0.92	0.77	0.57	0.88	0.82	0.80	0.79	0.80
	PoseFromShape (Chapter 3)	✓		ResNet-18	0.83	**0.86**	0.60	**0.95**	0.96	0.91	0.79	0.67	0.85	0.85	**0.82**	0.82	0.83
	PoseFromShape (Chapter 3)	✓		ResNet-50	0.83	**0.86**	0.61	**0.95**	0.96	0.92	0.80	0.67	0.84	0.82	**0.82**	0.83	0.83
	PoseContrast (ours)			ResNet-50	**0.85**	0.84	**0.64**	0.94	**0.97**	**0.95**	**0.86**	**0.71**	**0.91**	0.90	**0.82**	**0.85**	**0.85**

Table 5.2: **3D Pose Estimation of Class-Agnostic Methods on Pascal3D+ Xiang et al. (2014) (all classes seen).** All methods are evaluated with ground-truth bounding boxes. *The authors observe similar or worse performance with ResNet-50 Zhou et al. (2018). †StarMap actually obtains the rotation by solving for a similarity transformation between the image frame and world frame, weighting keypoint distances by the heatmap value.

	Method	w/ 3D	2D Bbox	46 tool	61 misc	130 b-case	166 w-drobe	297 desk	394 bed	739 table	1092 sofa	2894 chair	mean	5818 global
Acc30 ↑	(Dani et al., 2021)	✓	GT	**0.09**	0.10	0.62	0.57	0.66	0.58	0.40	0.94	0.50	0.50	0.58
	PoseFromShape*	✓	GT	0.07	**0.28**	0.71	0.65	0.71	0.54	0.53	0.94	0.79	0.58	0.75
	PoseContrast (ours)		GT	**0.09**	0.18	**0.81**	**0.68**	**0.78**	**0.68**	**0.54**	**0.97**	**0.86**	**0.62**	**0.80**
	PoseFromShape	✓	pred	0.07	**0.23**	0.68	0.55	0.71	0.51	**0.53**	0.93	0.77	0.55	0.73
	PoseContrast (ours)		pred	**0.09**	0.16	**0.72**	**0.58**	**0.77**	**0.65**	**0.53**	**0.97**	**0.85**	**0.59**	**0.79**

Table 5.3: **Cross-dataset 3D Pose Estimation of Class-Agnostic Methods on Pix3D (Sun et al., 2018).** The methods are trained on the Pascal3D+ (Xiang et al., 2014) train set and tested on Pix3D, where 6 classes are unseen (novel) and 3 classes are already seen (table, sofa, chair). As the classes are heavily unbalanced, we also report the global average (instance-wise rather than class-wise). We consider two kinds of input: ground-truth (GT) 2D object bounding box, and predicted (pred) bounding box by a class-agnostic Mask R-CNN detector. *3DPoseLite (Dani et al., 2021) reports much worse figures for PoseFromShape (Xiao et al., 2019) than what we got here with our own runs, probably due to a wrong experimental setting.

prediction, while model-based methods as (Xiao et al., 2019; Dani et al., 2021) do not construct powerful-enough embeddings despite their access to an actual 3D shape.

Overall, we achieve the best average performance in both metrics. In fact, we even outperform class-specific methods (Mahendran et al., 2017; Tulsiani and Malik, 2015; Mousavian et al., 2017; Su et al., 2015b; Prokudin et al., 2018; Grabner et al., 2018; Mahendran et al., 2018) except one (Liao et al., 2019), that reaches MedErr 9.2° and Acc30 88%, while we get 9.6° and 85%.

Stressing Class Agnosticism: Cross-Dataset Evaluation. To show our generalization ability, we follow the recent protocol proposed in Dani et al. (2021) and conduct a cross-dataset object pose estimation. We train on the 12 classes of Pascal3D+ (that has approximate pose annotations) and test on the 9 classes of Pix3D (with accurate poses), where only 3 classes coincide with Pascal3D+. Hence, 6 classes are totally unseen while 3 are already seen. Besides, methods that report cross-dataset results on Pix3D usually assume that ground-truth bounding boxes and 3D object models are given for testing (Xiao et al., 2019; Dani et al., 2021). We compare here in that same setting (see below for using detected objects). Results are in Table 5.3.

For the three seen classes ('table', 'sofa', 'chair'), our method outperforms all compared methods. It is consistent with results on Pascal3D+ (see Table 5.2), including for the difficult class 'chair'. More interestingly, we note that we can achieve a significantly better performance for certain unseen classes, even though there is no prior

Method	Setting	w/ 3D	Acc30 ↑	MedErr ↓
StarMap (Zhou et al., 2018)	no-shot		0.44	55.8
PoseContrast (ours)	no-shot		**0.55**	**42.6**
PoseFromShape (Chapter 3)	no-shot	✓	0.62	42.0
MetaView (Tseng et al., 2019)	10-shot		0.48	43.4
PoseContrast (ours)	10-shot		**0.60**	**38.7**
FSDetView ((Chapter 4))	10-shot	✓	0.63	32.1

Table 5.4: **Few-Shot Object Pose Estimation on ObjectNet3D (Xiang et al., 2016).** We report results on the 20 novel classes of ObjectNet3D as defined by (Zhou et al., 2018; Tseng et al., 2019). We compare both in no-shot and 10-shot settings, including with methods that additionally use 3D shapes.

knowledge of the testing objects for our network. As expected, it applies in particular to unseen classes that share a similar shape and canonical frame as seen classes, e.g., 'desk' and 'table'. By sharing weights across different classes during training, our class-agnostic pose estimation network learns a direct mapping from image embeddings to 3D poses and can easily apply to unseen objects when they have a similar shape as the training objects. But when the target objects possess a geometry widely differing from the training ones, such as 'tool' and 'misc', our purely image-based method usually fails; PoseFromShape does a bit better because it leverages a shape model, but accuracy remains poor. Some failure cases of our method can be seen in Figure 5.11.

We present in Figure 5.6 some visual results of our class-agnostic method on the cross-dataset 3D pose estimation, training on Pascal3D+ and testing on Pix3D (Sun et al., 2018).

Few-Shot Regime on ObjectNet3D. As detailed in Table 5.5, we first follow the no-shot setting proposed in (Zhou et al., 2018): we train on the 80 seen classes and test on the 20 unseen (novel) classes, cf. Table 5.4 (top). Compared to PoseFromShape (Xiao et al., 2019), both our approach and StarMap (Zhou et al., 2018) do not rely on 3D object models at test time, but exploit geometric similarities shared between seen and unseen classes. However, while StarMap struggles to predict precise 3D object coordinates and depth values, we can achieve a higher performance by using a simple network with good features.

We then evaluate in the 10-shot setting as used by (Tseng et al., 2019; Xiao and Marlet, 2020): the networks are first trained on the 80 seen classes, and then fine-

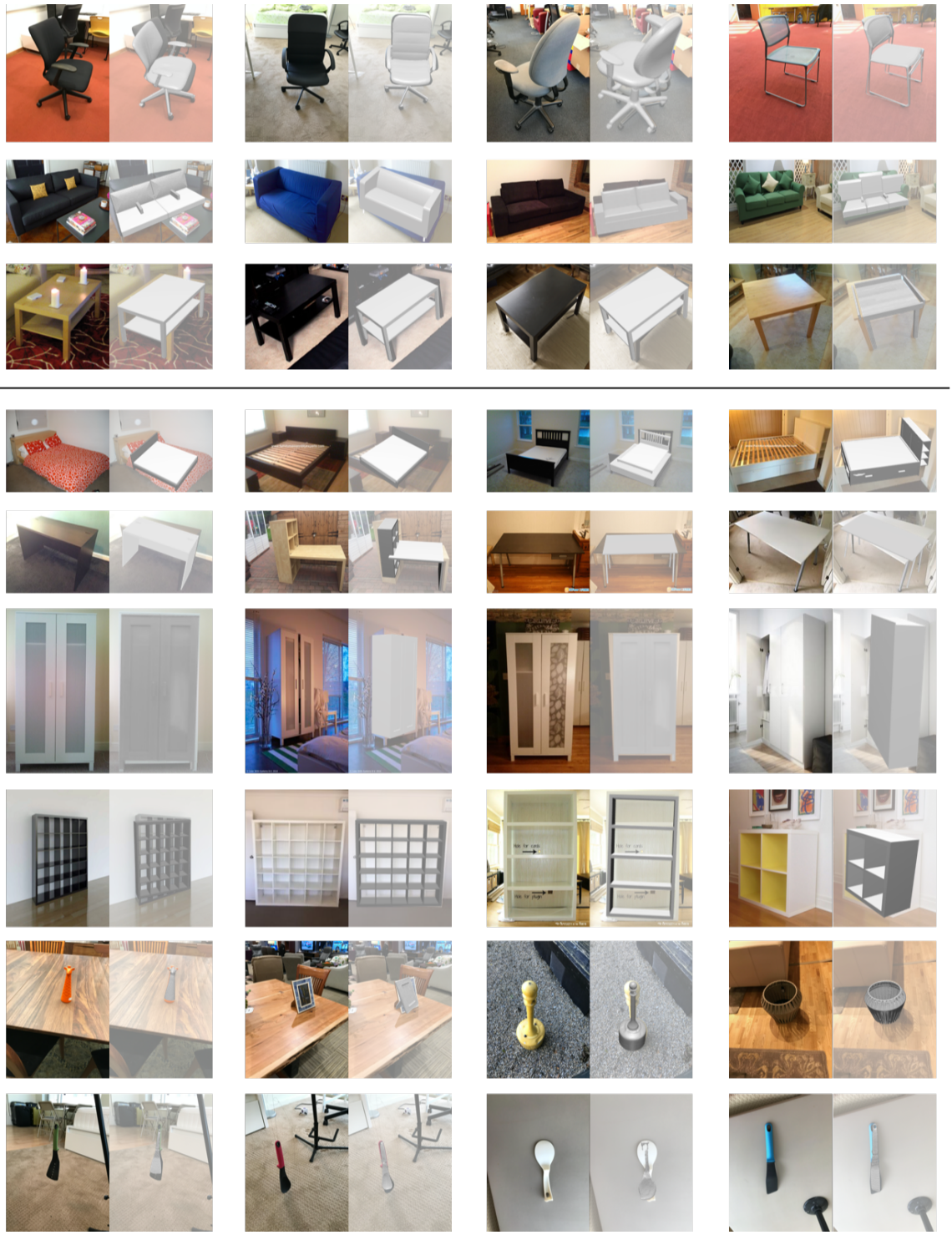


Figure 5.6: **Qualitative results on the 9 object classes of Pix3D (Sun et al., 2018).** The network is trained on the 12 object classes of Pascal3D+ and directly tested on Pix3D. From top to bottom: 'chair', 'sofa', 'table', 'bed', 'desk', 'wardrobe', 'bookcase', 'misc', and 'tool'. The 3 seen classes and the 6 unseen classes are separated by the black line. Here, 3D object models are only used to visualize the pose.

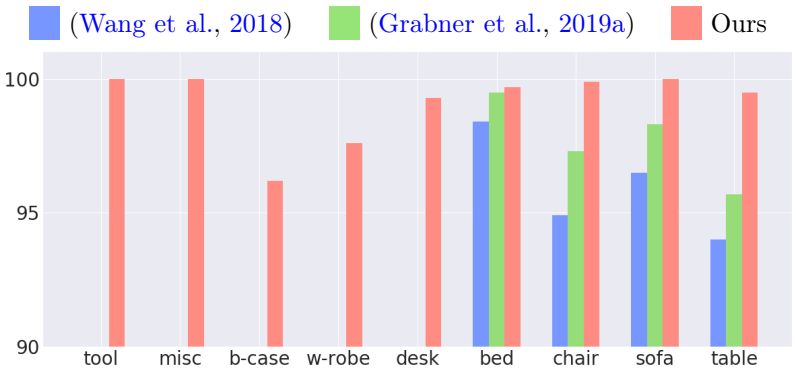


Figure 5.7: **Object Detection on Pix3D.** Results are given in $Acc_{D_{0.5}}$ as defined in (Grabner et al., 2019a). We compare with two methods (Wang et al., 2018; Grabner et al., 2019a) that train a class-specific Mask R-CNN on COCO, then fine-tune on a subset of Pix3D containing the same classes as COCO. In contrast, our agnostic Mask R-CNN is only trained on COCO and can generalize to classes not included in COCO.

tuned with a few labeled images from the 20 novel classes. Results are shown in Table 5.4 (bottom). Compared to MetaView (Tseng et al., 2019), that relies on class-specific keypoint prediction, we again find that our approach can obtain a better performance by sharing weights across all object classes.

In both settings, the best performing methods additionally use 3D object models (Xiao et al., 2019; Xiao and Marlet, 2020). Such a prior knowledge of the geometry makes sense, especially for unseen objects with shapes widely different shapes from training classes. Yet, our method nonetheless achieves promising results on these unseen classes, even compared to using a 3D model.

We also show class-wise results in Table 5.6. And we notice that our method can outperform CAD-model-based methods on a few classes, e.g., 'filling_cabinet', 'guitar' and 'wheelchair'. Besides, the relative gap between our method and these CAD-model-based methods is mostly due to a few classes, such as 'rifle', 'iron' or 'shoe', for which base classes offer limited help in terms of geometrical cue or canonical frame. In fact, if we put aside 'iron' and 'shoe', our method is on par with FSDetView (Chapter 4) on 10-shot viewpoint estimation, despite not using any extra shape information. Moreover, our class-agnostic network directly estimates the viewpoint from image embeddings, without relying on any keypoint prediction. This direct estimation network thus can predict the viewpoint for all classes, while keypoint-based methods struggle to get a reasonable prediction for certain classes, e.g., 'door', 'pen', and 'shoe'.

Base classes								Novel classes	
aeroplane	ashtray	backpack	basket	bench	bicycle	blackboard	boat	bed	bookshelf
bottle	bucket	bus	cabinet	camera	can	cap	car	calculator	cellphone
chair	clock	coffee_maker	comb	cup	desk_lamp	diningtable	dishwasher	computer	door
eraser	eyeglass	fan	faucet	file_extinguisher	fish_tank	flashlight	fork	filling_cabinet	guitar
hair_dryer	hammer	headphone	helmet	jar	kettle	key	keyboard	iron	knife
laptop	lighter	mailbox	microphone	motorbike	mouse	paintbrush	pan	microwave	pen
pencil	piano	pillow	plate	printer	racket	refrigerator	remote_control	pot	rifle
road_pole	satellite_dish	scissors	screwdriver	shovel	sign	skate	skateboard	shoe	slipper
sofa	speaker	spoon	stapler	suitcase	teapot	telephone	toaster	stove	toilet
toothbrush	train	train_bin	trophy	tvmonitor	vending_machine	washing_machine	watch	tub	wheelchair

Table 5.5: **Dataset split of ObjectNet3D Xiang et al. (2016):** 80 base classes (left) and 20 novel classes (right). Some novel classes share similar geometries and canonical frames as base classes, e.g., 'door'/'black_board', 'filling_cabinet'/'cabinet', 'wheelchair'/'chair'.

	Method *Acc30(↑) / MedErr(↓)*	bed	bookshelf	calculator	cellphone	computer	door	filling_cabinet
no-shot	StarMap Zhou et al. (2018)	0.37 / 45.1	0.69 / 18.5	0.19 / 61.8	0.51 / 29.8	0.74 / 15.6	– / –	0.78 / 14.1
	PoseContrast (ours)	0.62 / 17.4	0.89 / 6.7	0.65 / 17.7	0.57 / 15.8	0.85 / 14.5	0.91 / 2.7	0.88 / 10.4
	PoseFromShape	0.65 / 15.7	0.90 / 6.9	0.88 / 12.0	0.65 / 10.5	0.84 / 11.2	0.93 / 2.3	0.84 / 12.7
10-shot	StarMap* Zhou et al. (2018)	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 Tseng et al. (2019)	0.36 / 37.5	0.76 / 17.2	0.92 / 12.3	0.58 / 25.1	0.70 / 22.2	– / –	0.66 / 22.9
	PoseContrast (ours)	0.67 / 13.9	0.90 / 7.0	0.85 / 11.0	0.58 / 15.2	0.85 / 10.9	0.91 / 2.5	0.89 / 8.4
	FSDetView	0.64 / 14.7	0.89 / 8.3	0.90 / 8.3	0.63 / 12.7	0.84 / 10.5	0.90 / 0.9	0.84 / 10.5

	Method *Acc30(↑) / MedErr(↓)*	guitar	iron	knife	microwave	pen	pot	rifle
no-shot	StarMap Zhou et al. (2018)	0.64 / 20.4	0.02 / 142	0.08 / 136	0.89 / 12.2	– / –	0.50 / 30.0	0.00 / 104
	PoseContrast (ours)	0.73 / 14.4	0.03 / 124	0.25 / 122	0.93 / 7.5	0.45 / 39.8	0.76 / 9.2	0.00 / 102
	PoseFromShape	0.67 / 20.8	0.02 / 145	0.29 / 138	0.94 / 7.7	0.46 / 37.3	0.79 / 13.2	0.15 / 110
10-shot	StarMap* Zhou et al. (2018)	0.59 / 21.5	0.00 / 136	0.08 / 117	0.82 / 17.3	– / –	0.51 / 28.2	0.01 / 100
	MetaView Tseng et al. (2019)	0.63 / 24.0	0.20 / 76.9	0.05 / 97.9	0.77 / 17.9	– / –	0.49 / 31.6	0.21 / 80.9
	PoseContrast (ours)	0.73 / 14.7	0.03 / 126	0.23 / 116	0.94 / 6.9	0.45 / 41.3	0.78 / 10.6	0.04 / 90.4
	FSDetView	0.72 / 17.1	0.37 / 57.7	0.26 / 139	0.94 / 7.3	0.45 / 44.0	0.74 / 12.3	0.29 / 88.4

	Method *Acc30(↑) / MedErr(↓)*	shoe	slipper	stove	toilet	tub	wheelchair	TOTAL
no-shot	StarMap Zhou et al. (2018)	– / –	0.11 / 146	0.82 / 12.0	0.43 / 35.8	0.49 / 31.8	0.14 / 93.8	0.44 / 55.8
	PoseContrast (ours)	0.23 / 58.9	0.25 / 138	0.91 / 12.0	0.43 / 30.8	0.53 / 24.0	0.42 / 43.4	0.56 / 42.6
	PoseFromShape	0.54 / 28.2	0.32 / 158	0.89 / 10.1	0.61 / 21.8	0.68 / 17.8	0.39 / 57.4	0.62 / 42.0
10-shot	StarMap* Zhou et al. (2018)	– / –	0.15 / 128	0.83 / 15.6	0.39 / 35.5	0.41 / 38.5	0.24 / 71.5	0.46 / 50.0
	MetaView Tseng et al. (2019)	– / –	0.07 / 115	0.74 / 21.7	0.50 / 32.0	0.29 / 46.5	0.27 / 55.8	0.48 / 43.4
	PoseContrast (ours)	0.24 / 56.7	0.23 / 155	0.92 / 8.1	0.64 / 22.2	0.55 / 18.6	0.45 / 36.7	0.60 / 38.7
	FSDetView	0.51 / 29.4	0.25 / 96.4	0.92 / 9.4	0.69 / 17.4	0.66 / 15.1	0.36 / 64.3	0.63 / 32.1

Table 5.6: **Few-shot viewpoint estimation on ObjectNet3D Xiang et al. (2016).** All models are trained and evaluated on ObjectNet3D. For each method, we report Acc30(↑) / MedErr(↓) on the same 20 novel classes of ObjectNet3D, while the remaining 80 classes are used as base classes. *StarMap network trained with MAML Finn et al. (2017) for few-shot viewpoint estimation, with numbers reported in Tseng et al. (2019). Methods additionally using 3D object models are shown in gray.

Class-Agnostic Object Detection and Pose Estimation. To evaluate the coupling of generic object detection *and* generic pose estimation, we train a Mask R-CNN with backbone ResNet-50 on COCO in a class-agnostic way, merging all classes into a single one, then apply it directly on Pix3D without fine-tuning. All 9 Pix3D classes

Figure 5.8: **Qualitative Results on Pix3D.** For each sample, we first plot the original image, then we visualize the pose prediction obtained from the ground-truth bounding box and the detected bounding box, respectively. The top two rows show results for seen classes that intersect with training data in Pascal3D+ ('chair', 'sofa', 'table'), while the bottom two rows show results for novel classes. Note that the 3D CAD object models are only used here for pose visualization purpose; our approach does not rely on them for object pose prediction.

can thus be detected by our network, whether they are in COCO or not. To compare with other methods, we adopt the metric $Acc_{D_{0.5}}$ in Grabner et al. (2019a), which computes the percentage of objects for which the Intersection-over-Union (IoU) between the ground-truth box and the predicted box is larger than 50%.

Compared to class-specific methods performing object localization together with classification (Wang et al., 2018; Grabner et al., 2019a), our class-agnostic detection merges all classes into a single one. That is, we localize objects without classifying them into categories, not relying on semantic information for prediction. As shown in Figure 5.7, it provides a better detection accuracy and, more importantly, it enables the efficient detection of objects that are not included in COCO classes.

Qualitative results are illustrated in Figure 5.8. We find that both our object detector and our pose estimator can generalize to unseen objects (two bottom rows). Quantitative results are given in Table 5.3. We observe that our object pose estimation, evaluated using ground-truth boxes or predicted boxes, can outperform existing methods evaluated using ground-truth boxes only. This promising results suggests it is possible to develop autonomous systems that perform class-agnostic object detec-

tion and pose estimation on unknown objects in the wild.

5.4.3 Ablation Study

Initialization	Method	Pre-train data	Epochs	Acc30 ↑	MedErr ↓
from scratch	random	—	15*	0.76	12.8
from scratch	random	—	75	0.81	11.9
supervised	classification	ImageNet	15	0.83	10.7
unsupervised	SimCLR (Chen et al., 2020a)	ImageNet	15	0.83	11.0
unsupervised	SWAV (Caron et al., 2020)	ImageNet	15	0.84	10.2
unsupervised	MOCOv1 (He et al., 2020)	ImageNet	15	0.84	10.3
unsupervised	MOCOv2 (Chen et al., 2020c)	ImageNet	15	**0.85**	**9.6**

Table 5.7: **Network Initializations Evaluated on Pascal3D+.** We compare different initializations, training until convergence (*except for the first line), showing the number of epochs required.

Pre-trained Features. We initialize our image encoder network with MOCOv2 Chen et al. (2020c) to transfer rich features to the down-stream task of object pose estimation. Yet, other pre-trained features could be used He et al. (2020); Chen et al. (2020a); Caron et al. (2020), or learning from scratch. Table 5.7 reports results with various initializations.

Learning from scratch is suboptimal, probably due to the small dataset size, hence the relevance of using a pre-trained network. Convergence is also 5 times faster. Also, contrast-based pre-trained networks tend to perform best. In comparison, Grabner et al. (2018) also pre-trains on ImageNet while PoseFromShape has similar results with or without ImageNet pre-training. Zhou et al. (2018) uses a ResNet-18 trained from scratch for its keypoint-based 2-stack hourglass network. Pre-training is not known for Dani et al. (2021).

Loss	Eqs.	$\mathrm{d}(\mathbf{R}_i, \mathbf{R}_{k^-})$	Acc30 ↑	MedErr ↓
$\mathcal{L}_{\text{ang}}$	(5.1)	N/A	0.83	10.2
$\mathcal{L}_{\text{ang}} + \mathcal{L}_{\text{infoNCE}}$	(5.1), (5.3)	1	0.83	10.0
$\mathcal{L}_{\text{ang}} + \mathcal{L}_{\text{poseNCE}}$	(5.1), (5.4)	$(\Delta(\mathbf{R}_i, \mathbf{R}_{k^-})/\pi)^{1/2}$	0.84	9.8
$\mathcal{L}_{\text{ang}} + \mathcal{L}_{\text{poseNCE}}$	(5.1), (5.4)	$(\Delta(\mathbf{R}_i, \mathbf{R}_{k^-})/\pi)^{2}$	0.85	10.0
$\mathcal{L}_{\text{ang}} + \mathcal{L}_{\text{poseNCE}}$	(5.1), (5.4)	$\Delta(\mathbf{R}_i, \mathbf{R}_{k^-})/\pi$	**0.85**	**9.6**

Table 5.8: **Adding a Contrastive Loss, Alternative Pose Distances.** A contrastive loss with suited distance impacts Pascal3D+ results.

Adding a Contrastive Loss. Table 5.8 shows the relevance of adding a contrastive loss to the angle loss for pose estimation. However, adding the original InfoNCE loss only brings a minor improvement. A larger performance gap is obtained with our pose-aware contrastive loss of Eq. (5.4).

Alternative Pose Distances. Our contrastive loss relies on a distance between two poses $\mathrm{d}(\mathbf{R}_q, \mathbf{R}_k)$, defined as the normalized rotation difference $\Delta(\mathbf{R}_q, \mathbf{R}_k)/\pi$. Table 5.8 compares this definition to two variants: square and square root of this distance. All three perform better than the InfoNCE loss of Eq. (5.3), but the linear distance performs best.

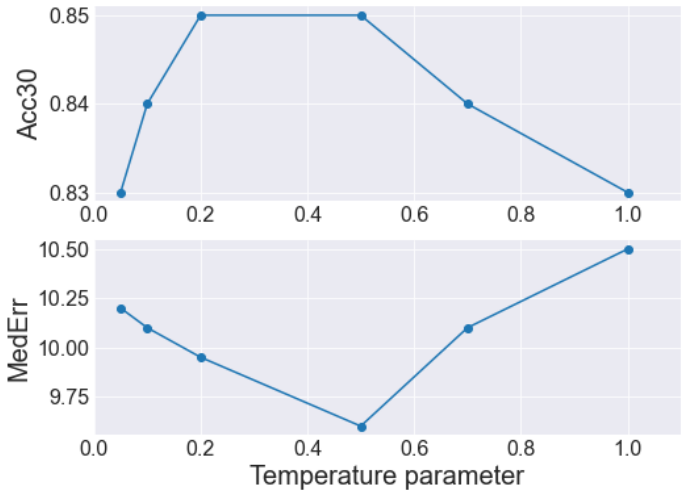
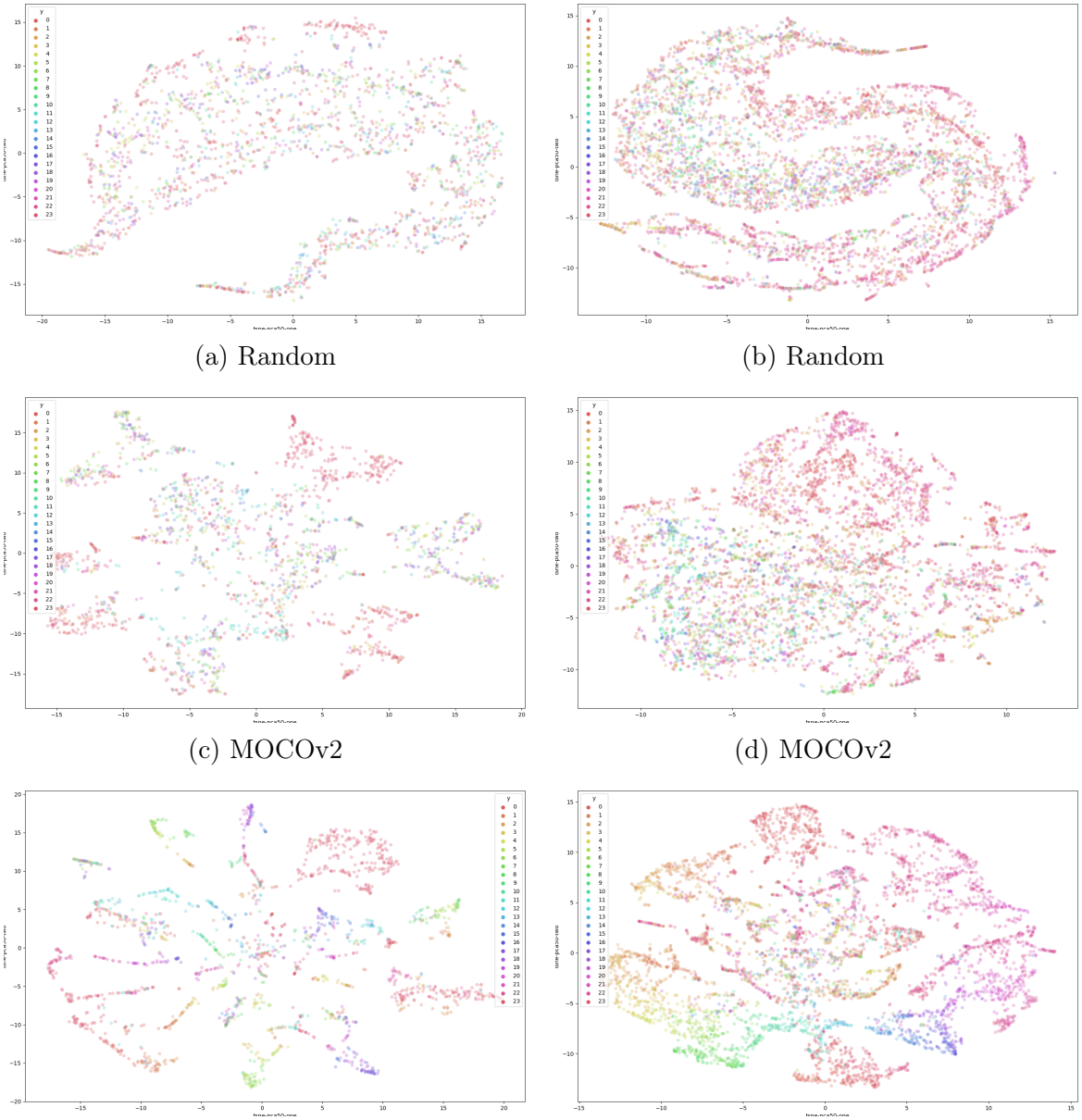


Figure 5.9: **Ablation on τ of $\mathcal{L}_{\text{poseNCE}}$.** We report the performance on the dataset Pascal3D+ (Xiang et al., 2014) with 30-degree accuracy (Acc30 ↑) and median error (MedErr ↓).

Temperature Parameter. Figure 5.9 shows the influence of temperature parameter τ in the proposed pose-aware contrastive loss $\mathcal{L}_{\text{poseNCE}}$. By varying this parameter from 0.05 to 1.0, we obtain the best performance on Pascal3D+ when $\tau = 0.5$. While training without this pose-aware contrastive loss can still reach an overall accuracy at 0.83 and an overall median error at 10.2, we note that the performance can be improved using the propose loss $\mathcal{L}_{\text{poseNCE}}$ with a temperature parameter between 0.2 and 0.6, which is quite robust.

Visualization of the Latent Space. To better understand the effect of contrastive learning, we use t-SNE to visualize the features obtained by different backbones.

Pascal3D Xiang et al. (2014) | Pix3D Sun et al. (2018)



(a) Random (b) Random

(c) MOCOv2 (d) MOCOv2

(e) PoseContrast (f) PoseContrast

Figure 5.10: **Feature Visualization.** We visualize image features from the val set of Pascal3D+ Xiang et al. (2014) (left) and Pix3D Sun et al. (2018) (right) by t-SNE (preceded by PCA) for three different ResNet-50 backbones: (a,b) randomly initialized network (top); (c,d) network pre-trained on ImageNet by MOCOv2 Chen et al. (2020c) (middle); and (e,f) network trained on Pascal3D+ with PoseContrast (bottom). We divide the 360 degrees of azimuth angle into 24 bins of 15° and use one color for each bin. The figure is better viewed in color with zoom-in.

Results are presented in Figure 5.10.

The features are extracted from val images of Pascal3D+. Considering the fact that the distributions of elevation and inplane-rotation are highly centered around a specific value compared to that of azimuth, we split the visualized features into different clusters, with each cluster corresponding to objects with similar azimuth angles. More specifically, we divide the 360 degrees of azimuth angle into 24 bins, and objects within the same azimuth bin are shown by the same color.

As seen in Figure 5.10 (left), the features extracted using a randomly-initialized network are more or less uniformly distributed across different locations in the latent space, and regardless of their 3D poses. On the contrary, the features extracted using networks trained with contrastive learning (MOCOv2 and PoseContrast) tend to form clusters, where each cluster groups objects with similar azimuth angles. Arguably, feature clusters are less spread with PoseContrast, compared to MOCOv2, and actual azimuths are more consistent within clusters.

When doing the same kind of visualization on Pix3D, as shown in Figure 5.10 (right), we observe more or less the same kind of distribution for the random initialization. However, MOCOv2 has a harder time clustering the features of Pix3D images, including regarding pose. Yet, PoseContrast manages to produce clusters, and with a better pose consistency.

5.4.4 Discussions

To understand where most prediction errors come from, we visualize some common failure cases in Figure 5.11. We notice that the current evaluation metrics do not consider the symmetries of certain objects, e.g., tables, that should be defined by 180°. The pose annotation is then arbitrary and the model has no way to know which orientation to choose.

In fact, a few other works specifically treat symmetries Hodaň et al. (2016); Drost et al. (2017); Balntas et al. (2017); Corona et al. (2018); Pitteri et al. (2019b). It is largely orthogonal to our proposal and left for future work. Note that it concerns only about 10-15% of the classes (e.g., table, bottle in Pascal3D+) and has little impact here as annotations generally assume the orientation with the smallest angle(s) w.r.t. the viewpoint.

Our approach also fails on unseen objects with shapes differing completely from training ones, e.g., 'tool' and 'misc' of Pix3D. But it actually is a problem to all

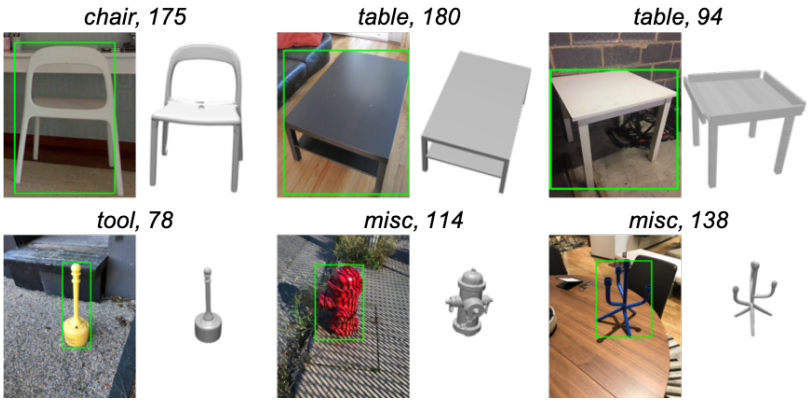


Figure 5.11: **Visualization of Failure Cases.** We show input image crops and predicted object poses, with class name and prediction error displayed at the top. Common failures come from ambiguous appearances of symmetrical objects, or shapes out of distribution.

the RGB-only class-agnostic methods (Zhou et al., 2018; Grabner et al., 2018), not specifically to ours, as generalizing towards unseen objects mainly relies on similarities. Even shape-based methods (Xiao et al., 2019; Dani et al., 2021), that exploit extra shape knowledge, nevertheless also struggle to get a good performance on these two classes.

5.5 Conclusion

We presented a new class-agnostic object pose estimation approach based on a pose-aware contrastive learning. Our network is trained end-to-end, leveraging on existing unsupervised contrastive features. We empirically show on various benchmarks that our method constitutes a strong baseline for class-agnostic object pose estimation. We also pave the way to more practical applications by successfully combining it with a class-agnostic object detector.

Chapter 6

Conclusions

In this chapter, we summarize the main contributions of this thesis and outline some research directions for future work.

6.1 Summary of Contributions

In this thesis, we have focused on 3D object pose estimation from single image with the aim to generalize towards unseen objects that are not included in the training data. This presented several challenges, such as designing a suitable model that can be easily applied to novel objects and finding a suitable learning scheme. We have investigated several methods for solving this problem in different settings. The contributions of each chapter are detailed below.

In Chapter 3, we proposed a shape-based deep pose estimation method for arbitrary 3D objects that can be unseen for the deep model during training. The network architecture follows the direct pose estimation pipeline where the predictions are directly obtained from abstract features via a *pose estimation* module. By contrast to traditional direct estimation methods, we have proposed to combine image features with shape features that are extracted from the corresponding 3D objects, and to estimate the relative 3D pose of the object in the input image with respect to the 3D model. This enables the model to estimate the pose of any object conditioned only on its 3D model, whether or not it is similar to objects seen at training time. In order to train this model, we have proposed a mixed classification-and-regression loss that works with the three Euler angles, namely azimuth, elevation, and inplane rotation. This pose parameterization was adopted in all the subsequent chapters of this thesis. Finally, we have developed a specific data augmentation technique that randomizes the object canonical frame during training, which further improved the model generalization properties. We have shown that this approach holds the promise of a completely generic deep learning method for pose estimation, independent of the object category and training data, by showing encouraging results on the unseen objects without any specific training, and despite the domain gap between synthetic training data and real images for testing.

In Chapter 4, we have extended the previously proposed model under the few-shot setting, which does not assume an aligned shape for each object but a few exemplar shapes for each category. It thus allows pose estimation of any object class conditioned on a few labeled samples with exemplar 3D models rather than instance-level 3D

models. Based on this estimation conditioned on a class-level representation, we have proposed an unified framework to tackle both few-shot object detection and few-shot viewpoint estimation. Our approach showed state-of-the-art results for both tasks on various benchmarks. Instead of conducting viewpoint estimation on ground-truth bounding boxes as previous works, we have proposed a full evaluation protocol that combines object detection and viewpoint estimation and shown encouraging results for objects in the wild.

In Chapter 5, we have proposed a new class-agnostic model for generic object viewpoint estimation. Contrary to the model from previous chapters which relies on the aggregated features extracted from both images and shapes, this model can estimate the viewpoint directly from image embeddings, bypassing the requirement of any 3D model. This is achieved by optimizing the image embedding space with a pose-aware contrastive loss that enhances the contrast between positive and negative pairs, depending on pose difference. Furthermore, we have separated the application of pose-variant and pose-invariant data augmentations in different stages: pose-variant augmentations are applied at *mini-batch creation time* for getting more training samples with different annotations; pose-invariant augmentations are applied at *contrast time* for getting positives with the same pose. We have also used a class-agnostic object detection network for detecting unseen objects, and then estimated their poses using the proposed model. Without any specific finetuning or post-processing, promising detection and viewpoint estimation results have been achieved on unseen objects with similar geometries as the training objects.

6.2 Future Work

In this section, we analyze some future research directions that could extend the works presented in this thesis.

End-to-end object detection and pose estimation. Most viewpoint estimation methods rely on the ground-truth classes and on the detected bounding boxes to obtain image crops centered on objects of interest and then estimate their viewpoints based on features extracted from the image crops, which alleviates the influence of other objects in the same image. This two-step pipeline usually requires the training of two independent models and generate noisy intermediate detection results with overlaps. However, for 3D scene understanding in the industrial settings, we want

to have an end-to-end model to obtain clean detection results per object as well as the corresponding 3D attributes such as 3D pose or 3D shape. Different approaches have been proposed to address the object detection and viewpoint estimation jointly in an end-to-end network architecture. (Massa et al., 2016a) presented the benefits of joint training for object detection and viewpoint estimation. (Divon and Tal, 2018) provided further insights into this direction with detailed analysis on the network architecture and loss function. (Kundu et al., 2018) trained a deep convolutional network that learns to map image regions to the full 3D shape and pose of all object instances in the image. However, despite these efforts, none of these works have tried to generalize towards unseen object classes that are not included in the training data. In Chapter 5, a class-agnostic viewpoint estimation model was trained separately with a class-agnostic object detection model for the joint detection and viewpoint estimation on unseen objects. We believe that this approach has the potential of enabling class-agnostic training of an end-to-end model for generic object detection together with viewpoint estimation, and therefore constitutes a possible future research direction.

3D translation estimation for unseen objects. All methods presented in this thesis focus on the estimation of object viewpoint while ignoring the 3D translation vectors, which means that the actual location of objects in the 3D scenes are unknown to the system. An important future direction would be to estimate the 3D object orientations together with the 3D translation vectors for various objects in the wild without relying on any prior knowledge on the objects or the environment. It would be a significant improvement, and it would allow us to apply the method to new objects in any situation without restriction. This is extremely important for real-world applications where new objects could come into play at any time. The computer vision research community has been studied this problem for years under the table-top setting with typically tens of objects (Hinterstoißer et al., 2012b; Hodan et al., 2017). Recently, (Wang et al., 2019a) proposed a new dataset for category-level object pose estimation, where the full 6-DoF poses and object scales need to be predicted for unseen objects within the seen categories. However, how to estimate the 3D translation of objects from unseen categories remains unclear, especially for images captured in the wild with depth ambiguity.

For this purpose, we could work on two potential directions. One possible solution is to learn a model that can estimate the depth value and the object's 3D bounding

box center from the image, from which the 3D translation vectors can be retrieved using the camera projection matrix. Another way to estimate the translation is via analysis-by-synthesis, where we adjust the translation vectors until the rendering of object model under the estimated pose fits well the observation in the image. This approach requires the actual object scales and the camera parameters to be known.

Pose refinement with 3D model. A natural extension of our work proposed in Chapter 3 would be implementing a pose refinement model to improve the accuracy of the coarse pose estimation. This pose refinement module should work on color images, which updates the pose estimation by rendering the 3D model under the estimated pose and comparing it with the input images. While most pose refinement methods proposed in previous works only work with seen objects, DeepIM (Li et al., 2018b) has shown some preliminary results of pose refinement on unseen objects when training and testing on synthetic images. Therefore, how to perform pose refinement on real images containing unseen objects remains an open research direction. Moreover, based on the available large-scale databases of 3D models, it would also be interesting to conduct 3D model retrieval to get an aligned model for the target object. This retrieved 3D model could be used for the pose refinement, and could even be refined together with the pose estimation for a better 2D-3D alignment.

Object pose estimation with interactions. Generating pose estimation for humans interacting with various objects is one of the next critical challenges in 3D scene understanding. Works presented in this thesis mainly focus on object pose estimation in uncontrolled environments, where the objects are usually static or moving by themselves in the scene. However, typical scenarios in many realistic applications include the interactions between a human and the objects. In these scenarios, the joint estimation of object pose and human pose becomes important for the full understanding of the properties and motions in the 3D scene. The interactions could also be extended to robots for robotic manipulations. So, moving forward, we would like estimate the poses of objects and their manipulators in uncontrolled environments, where the manipulators could be human hands, the whole human body, or a robot arm. Prior approaches have been proposed to solve the pose estimation of objects, humans and robots independently. Very few works attempted to solve the joint estimation problem. (Hampali et al., 2020) proposed a novel dataset for hand-object pose estimation, covering 68 sequences with 10 different persons manipulating 10 dif-

ferent objects taken from YCB dataset (Xiang et al., 2018). Recently, (Chao et al., 2021) proposed a larger hand-object dataset covering 20 objects from YCB captured in rich, real RGB-D sequences. (Zhang et al., 2020) has gone further beyond human hands by proposing a method that infers spatial arrangements and shapes of both the humans and objects in a globally consistent 3D scene, all from a single image captured in an uncontrolled environment. In the evaluation on images taken from COCO dataset (Lin et al., 2014), they have shown encouraging results where the spatial arrangements are physically plausible. While all these previous works show interesting results on hand-object or human-object interactions, they are all limited to a small number of objects or very few object classes. Thus, developing a method that can be directly applied on objects from unseen classes in varying interactions could be an important step for the community.

Bibliography

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

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 *International Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T.-K. Kim. Pose Guided RGBD Feature Learning for 3D Object Pose Estimation. In *International Conference on Computer Vision (ICCV)*, 2017.

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

L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

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

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.

E. Brachmann, F. Michel, A. Krull, M. Yang, S. Gumhold, and C. Rother. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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. Technical Report arXiv:1512.03012 [cs.GR], Stanford University – Princeton University – Toyota Technological Institute at Chicago, 2015.

Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, J. Kautz, and D. Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020a.

T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

X. Chen, H. Fan, R. Girshick, and K. He. Improved Baselines with Momentum Contrastive Learning. *arXiv preprint arXiv:2003.04297*, 2020c.

E. Corona, K. Kundu, and S. Fidler. Pose estimation for objects with rotational symmetry. In *International Conference on Intelligent Robots and Systems*, 2018.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

M. Dani, K. Narain, and R. Hebbalaguppe. 3DPoseLite: A Compact 3D Pose Estimation Using Node Embeddings. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021.

J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

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.

G. Divon and A. Tal. Viewpoint Estimation—Insights & Model. In *European Conference on Computer Vision (ECCV)*, 2018.

C. Doersch, A. Gupta, and A. A. Efros. Unsupervised Visual Representation Learning by Context Prediction. *IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

B. Drost, M. Ulrich, P. Bergmann, P. Härtinger, and C. Steger. Introducing MVTec ITODD — a dataset for 3D object recognition in industry. In *International Conference on Computer Vision Workshops (ICCVW)*, 2017.

M. Elhoseiny, T. El-Gaaly, A. Bakry, and A. M. Elgammal. A comparative analysis and study of multiview CNN models for joint object categorization and pose estimation. In *International Conference on Machine Learning (ICML)*, 2016.

F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe. Exploring spatial context for 3D semantic segmentation of point clouds. In *International Conference on Computer Vision (ICCV)*, 2017.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

L. Ferraz, X. Binefa, and F. Moreno-Noguer. Very fast solution to the PnP problem with algebraic outlier rejection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

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.

D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. *International Conference on Computer Vision (ICCV)*, 1999.

Y. Ge, J. Zhao, and L. Itti. Pose augmentation: Class-agnostic object pose transformation for object recognition. In *European Conference on Computer Vision (ECCV)*, 2020.

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

G. Georgakis, S. Karanam, Z. Wu, and J. Kosecka. Matching RGB images to CAD models for object pose estimation. *CoRR*, abs/1811.07249, 2018.

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

S. Gidaris, P. Singh, and N. Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

R. Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

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.

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)*, 2019a.

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)*, 2019b.

F. S. Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 1998.

T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. AtlasNet: A papier-mâché approach to learning 3D surface generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

A. Grover and J. Leskovec. node2vec: Scalable Feature Learning for Networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

R. A. Güler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos. DenseReg: Fully Convolutional Dense Shape Regression In-the-Wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

D. Ha, A. Dai, and Q. V. Le. HyperNetworks. In *International Conference on Learning Representations (ICLR)*, 2017.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742, 2006.

S. Hampali, M. Rad, M. Oberweger, and V. Lepetit. HOnnotate: A method for 3D Annotation of Hand and Object Poses. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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.

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

K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.

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

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

K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020.

S. Hinterstoißer, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

S. Hinterstoißer, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. *International Conference on Computer Vision (ICCV)*, 2011.

S. Hinterstoißer, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012a.

S. Hinterstoißer, V. Lepetit, S. Ilic, S. Holzer, G. 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)*, 2012b.

R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019.

T. Hodaň, J. Matas, and Š. Obdržálek. On evaluation of 6d object pose estimation. In *ECCV Workshops (ECCVw)*, 2016.

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. *Winter Conference on Applications of Computer Vision (WACV)*, 2017.

S. Holzer, S. Hinterstoißer, S. Ilic, and N. Navab. Distance transform templates for object detection and pose estimation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

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.

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)*, 2020a.

Y. Hu, J. Hugonot, P. Fua, and M. Salzmann. Segmentation-Driven 6D Object Pose Estimation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Y. Hu, P. Fua, W. Wang, and M. Salzmann. Single-Stage 6D Object Pose Estimation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020b.

D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1993.

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.

W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab. Deep Learning of Local RGB-D Patches for 3D Object Detection and 6D Pose Estimation. *European Conference on Computer Vision (ECCV)*, 2016.

W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

A. Kendall, M. K. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *International Conference on Computer Vision (ICCV)*, 2015.

P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning Workshops*, 2015.

A. Kundu, Y. Li, and J. M. Rehg. 3D-RCNN: Instance-level 3D object reconstruction via render-and-compare. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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.

W. Kuo, A. Angelova, T.-Y. Lin, and A. Dai. Mask2CAD: 3D Shape Prediction by Learning to Segment and Retrieve. In *European Conference on Computer Vision (ECCV)*, 2020.

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.

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.

V. Lepetit. Recent Advances in 3D Object and Hand Pose Estimation. *arXiv*, 2020.

V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision (IJCV)*, 2009.

C. Li, J. Bai, and G. D. Hager. A unified framework for multi-view multi-class object pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018a.

D. Li, H. Wang, Y. Yin, and X. Wang. Deformable registration using edge-preserving scale space for adaptive image-guided radiation therapy. *Journal of Applied Clinical Medical Physics (JACMP)*, 2011.

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

S. Li, C. Xu, and M. Xie. A robust O(n) solution to the perspective-n-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.

Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. DeepIM: Deep iterative matching for 6D pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018b.

Z. Li, G. Wang, and X. Ji. CDPN: Coordinates-Based Disentangled Pose Network for Real-Time RGB-Based 6-DoF Object Pose Estimation. *International Conference on Computer Vision (ICCV)*, 2019.

S. Liao, E. Gavves, and C. G. M. Snoek. Spherical Regression: Learning Viewpoints, Surface Normals and 3D Rotations on N-Spheres. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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.

T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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.

D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1991.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 2004.

S. Mahendran, H. Ali, and R. Vidal. 3D pose regression using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

S. Mahendran, H. Ali, and R. Vidal. 3D Pose Regression Using Convolutional Neural Networks. In *International Conference on Computer Vision Workshops (ICCV Workshops)*, 2017.

S. Mahendran, H. Ali, and R. Vidal. A mixed classification-regression framework for 3D pose estimation from 2D images. In *British Machine Vision Conference (BMVC)*, 2018.

F. Manhardt, W. Kehl, N. Navab, and F. Tombari. Deep model-based 6D pose refinement in RGB. In *European Conference on Computer Vision (ECCV)*, 2018.

F. Massa and R. Girshick. Maskrcnn-Benchmark: Fast, Modular Reference Implementation of Instance Segmentation and Object Detection Algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018.

F. Massa, R. Marlet, and M. Aubry. Crafting a multi-task CNN for viewpoint estimation. In *British Machine Vision Conference (BMVC)*, 2016a.

F. Massa, B. Russell, and M. Aubry. Deep Exemplar 2D-3D Detection by Adapting from Real to Rendered Views. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016b.

I. Misra and L. V. D. Maaten. Self-Supervised Learning of Pretext-Invariant Representations. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6706–6716, 2020.

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

M. Noroozi and P. Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European Conference on Computer Vision (ECCV)*, 2016.

M. Oberweger, M. Rad, and V. Lepetit. Making deep heatmaps robust to partial occlusions for 3D object pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

A. Oord, Y. Li, and O. Vinyals. Representation Learning with Contrastive Predictive Coding. *ArXiv*, abs/1807.03748, 2018.

M. Osadchy, Y. L. Cun, and M. L. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research (JMLR)*, 2007.

K. Park, T. Patten, and M. Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. *International Conference on Computer Vision (ICCV)*, 2019.

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.

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.

H. Penedones, R. Collobert, F. Fleuret, and D. Grangier. Improving object classification using pose information. Technical report, Idiap Research Institute, 2012.

S. Peng, Y. Liu, Q. Huang, H. Bao, and X. Zhou. PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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)*, 2019a.

G. Pitteri, M. Ramamonjisoa, S. Ilic, and V. Lepetit. On object symmetries and 6d pose estimation from images. In *International Conference on 3D Vision (3DV)*, 2019b.

G. Pitteri, A. Bugeau, S. Ilic, and V. Lepetit. 3d object detection and pose estimation of unseen objects in color images with local surface embeddings. In *Asian Conference on Computer Vision (ACCV)*, 2020.

S. Prokudin, P. Gehler, and S. Nowozin. Deep directional statistics: Pose estimation with uncertainty quantification. In *European Conference on Computer Vision (ECCV)*, 2018.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a.

C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Conference on Neural Information Processing Systems (NIPS)*, 2017b.

C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3D object detection from RGB-D data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

H. Qi, M. Brown, and D. G. Lowe. Low-shot learning with imprinted weights. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017c.

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 *International Conference on Computer Vision (ICCV)*, 2017.

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.

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.

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

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

J. Redmon and A. Farhadi. Yolov3: An incremental improvement, 2018. arXiv preprint arXiv:1804.02767.

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.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *International Conference on Neural Information Processing Systems (NeuIPS)*, 2015.

R. Rios-Cabrera and T. Tuytelaars. Discriminatively Trained Templates for 3D Object Detection: A Real Time Scalable Approach. *International Conference on Computer Vision (ICCV)*, 2013.

C. Sahin, G. Garcia-Hernando, J. Sock, and T.-K. Kim. Instance- and Category-level 6D Object Pose Estimation. *Chapter of "RGB-D Image Analysis and Processing"*, 2019.

C. Sahin, G. Garcia-Hernando, J. Sock, and T.-K. Kim. A Review on Object Pose Recovery: from 3D Bounding Box Detectors to Full 6D Pose Estimators. *Journal of Image and Vision Computing*, 2020.

E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, S. Pankanti, R. S. Feris, A. Kumar, R. Giryes, and A. M. Bronstein. RepMet: Representative-based metric learning for classification and few-shot object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

X. Shen, A. A. Efros, and M. Aubry. Discovering visual patterns in art collections with spatially-consistent feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell. Weakly-supervised discovery of visual pattern configurations. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

C. Steger. Occlusion, clutter, and illumination invariant object recognition. In *International Archives of Photogrammetry and Remote Sensing (ISPRS)*, 2002.

H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view Convolutional Neural Networks for 3D Shape Recognition. *IEEE International Conference on Computer Vision (ICCV)*, 2015a.

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)*, 2015b.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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.

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

B. Tekin, S. N. Sinha, and P. Fua. Real-time seamless single shot 6D object pose prediction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Y. Tian, D. Krishnan, and P. Isola. Contrastive Multiview Coding. In *European Conference on Computer Vision (ECCV)*, 2020.

H. Tjaden, U. Schwanecke, and E. Schömer. Real-Time Monocular Pose Estimation of 3D Objects Using Temporally Consistent Local Color Histograms. In *International Conference on Computer Vision (ICCV)*, 2017.

E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010.

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.

S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

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

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010.

O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

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)*, 2019a.

X. Wang and A. Gupta. Unsupervised Learning of Visual Representations Using Videos. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2794–2802, 2015.

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.

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.

Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019b.

Y.-X. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision (ECCV)*, 2016.

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

P. Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3D pose estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Z. Wu, Y. Xiong, S. Yu, and D. Lin. Unsupervised Feature Learning via Non-parametric Instance Discrimination. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3733–3742, 2018.

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

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 Computer Vision (ECCV)*, 2016.

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.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

T. Xiao, X. Wang, A. A. Efros, and T. Darrell. What Should Not Be Contrastive in Contrastive Learning. In *International Conference on Learning Representations (ICLR)*, 2021.

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.

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.

D. Xu, D. Anguelov, and A. Jain. PointFusion: Deep sensor fusion for 3D bounding box estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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.

S. Zakharov, W. Kehl, B. Planche, A. Hutter, and S. Ilic. 3D object instance recognition and pose estimation using triplet loss with dynamic margin. *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

S. Zakharov, I. S. Shugurov, and S. Ilic. Dpod: 6d pose object detector and refiner. *International Conference on Computer Vision (ICCV)*, 2019.

J. Y. Zhang, S. Pepose, H. Joo, D. Ramanan, J. Malik, and A. Kanazawa. Perceiving 3D Human-Object Spatial Arrangements from a Single Image in the Wild. In *European Conference on Computer Vision (ECCV)*, 2020.

R. Zhang, P. Isola, and A. A. Efros. Colorful Image Colorization. In *European Conference on Computer Vision (ECCV)*, 2016.

Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi. Revisiting the PnP problem: A fast, general and optimal solution. In *International Conference on Computer Vision (ICCV)*, 2013.

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.

Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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

R. Zhu, H. K. Galoogahi, C. Wang, and S. Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.