THE UNIVERSITY
of EDINBURGH

Unsupervised category-level viewpoint estimation

Octave Mariotti

Doctor of Philosophy

Institute of Perception, Action and Behaviour

School of Informatics

University of Edinburgh

2022

Abstract

The recent progress in deep learning techniques transformed the field of computer vision, with tasks like object classification or segmentation being almost considered solved. This however requires sufficiently many labeled samples to train the system, hence research focus has shifted towards tasks where collecting such data is challenging. Recovering camera poses is one such task, where labels are typically too costly for supervised approaches. This work explores solutions to train camera pose estimation systems without the need for external supervision.

Preliminary assessments show that it is possible to formulate this problem as a self-supervised reconstruction task. By interpreting a network output as 3D rotation, and using this output to control a differentiable rendering operation, gradient descent can be used to train the network to predict viewpoint information. However, multiple issues arise when applying such a method naively on complex data. Confounding factors of particular importance are symmetries, geometry-breaking rendering pipelines and background-induced noise. This leads to a regime where purely self-supervised training breaks, although semi-supervised approaches are still successful.

Specific solutions to the aforementioned problems are therefore studied and evaluated. For symmetries, multiple viewpoint predictions are made, and their distribution is further regulated. Two main rendering pipelines are also compared to improve over naive convolution-based reconstruction: a voxel-based one, and a more recent implicit neural representation. Experimental evidence shows that carefully crafting a system with these improvements allows recovery of poses on many everyday objects, such as cars and chairs, with performances reaching the level of supervised approaches on some categories.

In addition, this thesis underlines two potential problems in related approaches. First, an unstable pose retrieval method used in recent implicit representations, that is prohibitively expensive. Second, an insidious issue in unsupervised methods, arising from a combination of dataset biases and naive calibration. As this potentially leads to overestimated

performances, it calls for a more robust evaluation standard, as well as more careful data gathering.

Acknowledgements

Undertaking a PhD is a complex and very much nonlinear endeavour that could be described in infinitely many epigrammatic ways, but as I picture it today, it clearly appears above all else to be a learning process. From being overly enthusiastic, prone to jumping on the bandwagon and overly confident in my conception of good scientific work, to slightly less eager but infinitely more measured and hopefully mature, the lessons I take from this journey are plenty, undoubtedly humbling, and in hindsight necessary and very much welcome.

Of course, there is rarely learning without teaching, and for this reason, I would like to express my utmost gratitude to Hakan Bilen and Oisin Mac Aodha for their guidance, immeasurable help, and benevolence. With the academic ecosystem struggling to steer off a bleak publish-or-perish future, I feel particularly lucky to have found mentors that always took the time to ensure my work was headed in the correct direction. I would also like to thank William Smith and Elliot Crowley for accepting to review and evaluate this work.

While only my name is inscribed on the front page of this thesis, one of the main lessons of the time spent working on it is that research is before all collaborative and cannot be envisioned without the help of others. Hence, I would like to thank all that I have had the chance to discuss my work with and get feedback from, in particular Lucas Deecke, Boyan Gao, Wei-Hong Li, Taha Kocyigit, Konda Mopuri, Arushi Goel, Bo Zhao, Yu Yang, Simon Reinkemeier, Titas Anciukevičius, Robin Vogel and Xialei Liu. I hope I was able to give back as much help as they have given me, as their analysis of my ideas under their own prisms has been crucial in this work. Our frequent interactions reinforced my resolve to keep on seeking a life path in which my contributions would not only be towards my own scientific interests but also aimed at helping others pursue theirs.

Yet perhaps the most important teaching I received is that of the value of comradeship, especially when going through difficult times as PhDs or global pandemics can most

certainly be. First and foremost, I am immensely grateful to Arushi and Adarsh, who have offered me unwavering support. Their help and encouragements were undoubtedly the most efficient remedy to low morale, and this work would certainly not have been carried through the same way without them. I would also like to thank Divakaran, Bénédicte and Nikita for the stimulating times we spent at work and especially outside of it, Laurent for our regular climbing sessions that always were the best breathers I could get, Dorian, for being a faithful friend whose company I keep enjoying more as years go by, and Anaïs for being a part of this journey.

Finally, I am forever indebted to my family, Apolline, Françoise and Catherine, whose efforts and sacrifices have made this work possible.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Octave Mariotti*)

À Jean-Marie, Huguette, Claude et Madeleine.

Table of Contents

Chapter 1

Introduction

2022 marks the 10-year anniversary of AlexNet's breakthrough results at the ILSVRC 2012 competition, with a top-5 classification error of 15.3%, more than 10 points lower than the runner-up (Krizhevsky et al., 2012; Russakovsky et al., 2015). This milestone, often considered to be the inception of modern deep learning, triggered a shift in the machine learning and computer vision communities towards the adoption of deep neural networks as the main model type. Since then, impressive progress have been made in the field of computer vision, with classification systems bringing the top-5 error down to 1.2% (Pham et al., 2021), while other tasks like object detection or action recognition in videos also saw soaring progress (Li et al., 2022; Gowda et al., 2021).

Most of this progress however comes from fully supervised settings, in which models are trained using annotated datasets, typically made of sample-target pairs. This restricts most deep learning approaches to tasks where such sets can reasonably be built, which implies i) cheap data collection, ii) cheap data annotation, and iii) clear definition of the targets per sample. Counter-examples for each of these requirements are crash events in self-driving vehicles, tracking an object in a video, and the cost of placing a placing a stone in a game of go. As a result, interest towards less restrictive settings steadily grew with recent success in self-supervised, unsupervised and reinforcement learning (Devlin

et al., 2018; Lample et al., 2018; Silver et al., 2017).

Recovering camera viewpoint is among the current challenges in computer vision, as it belongs to the second category of tasks to which fully supervised learning is not applicable, i.e. those that have an high labeling cost. The annotation process either involves manual labeling of already existing image collection, usually by asking human annotators to align a 3D model with the object as it is seen in the picture (Xiang et al., 2014), or by recording poses while taking new pictures in a calibrated environment, which limits the type and size of objects that can be treated (Georgakis et al., 2016; Hodan et al., 2017). In addition, reusing pre-existing datasets means poor quality control, which can lead to biased sample distribution.

Typically, viewpoint is defined as a set of three angles - azimuth, elevation, and in-plane rotation - that describe the relative rotation between the camera and a predefined frame of reference. This work focuses on *category-level* viewpoint estimation, where a single frame of reference is defined for a whole category of objects. This follows a natural human intuition that all objects that belong to the same category can be aligned in the same way, e.g. all cars have a ”front” and ”back” side (Fig. 1.1).

This work explores the possible methods for leveraging unlabeled samples for the task of viewpoint estimation. Chapter 3 establishes the basic principles that will be used in the rest of the thesis to learn unsupervised viewpoints. Recent analysis-by-synthesis methods are able to discover interpretable representations for the data, like keypoints on human bodies (Jakab et al., 2018) or faces (Thewlis et al., 2017b), by using an encoder-decoder architecture with a carefully crafted bottleneck to enforce desirable properties on the learned embeddings, e.g. sparsity and semantic consistency on keypoints. As previous approaches are limited to 2D cases or when rotations are known, this chapter investigates the possibility of applying similar techniques to learn viewpoints from images without labels. A simple architecture is proposed and evaluated on a variety of synthetic datasets. Experimental evidence shows it works well in a semi-supervised setting, with the intro-

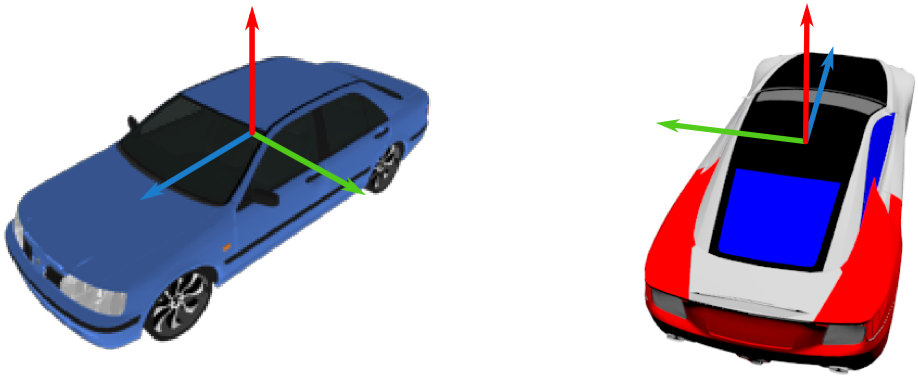


Figure 1.1: Illustration of natural frame of reference superimposed on cars. The blue axis is aligned towards the front, the green one towards the left side, and the red towards the roof. This can easily be defined, whichever specific instance of car is considered

duction of new unlabeled samples boosting performances beyond supervised baselines. It also outperforms off-the-shelf semi-supervised methods thanks to its task-specific design. Unsupervised experiments demonstrate some limited ability on toy examples, but fail on more complex categories like cars. This is caused by the reconstruction pipeline finding shortcuts to minimize reconstructions without having to learn viewpoints. Chapter 4 and 5 focus on this flaw by introducing different reconstruction mechanisms.

To prevent the shortcuts from occurring and leading the predictions to uninterpretable viewpoints, the CNN-based decoder is replaced in Chapter 4 with a voxel-based reconstruction pipeline coupled with a pseudo-rendering operation to guarantee geometric consistency between different decoded views. Concurrently, the viewpoint estimator is fitted with multihead predictor in order to overcome confusions created by object symmetries, allowing it to test multiple hypothesis and select the best one.

This allows the proposed system, called ViewNet, to be trained in a fully unsupervised way, and to be evaluated on PASCAL3D+ (Xiang et al., 2014), a dataset comprising real

images of 12 object categories annotated with viewpoints. A severe limitation of PASCAL3D+ as a viewpoint estimation benchmark is further illustrated, as careful analysis of the predictions shows that unsupervised and even completely untrained systems can outperform supervised approaches in some categories by exploiting viewpoint distribution biases.

One of the limitations of ViewNet is its relatively simplistic voxel-based reconstruction pipeline, which is enough to reconstruct synthetic objects, but struggles on real data. Therefore, Chapter 5 extends ViewNet by replacing its voxel predictor with a more powerful implicit representation inspired by recent developments in 3D reconstruction. The resulting system, called ViewNeRF, contains a simplified conditional Neural Radiance Field (Mildenhall et al., 2020), coupled with a novel viewpoint regularization term to help prevent prediction collapse. In particular, performance in real settings in significantly boosted compared with ViewNet, thanks to NeRF's ability to model complex light patterns like reflections on metallic surfaces.

Additionally, the approach of using a network to predict viewpoint is compared with the gradient-based direct pose recovery, a technique widespread in the implicit representation literature (Yen-Chen et al., 2021). Results show that on top of being much slower and computationally expensive, it is unable to correctly estimate viewpoints due to its inability to properly disentangle it from object appearance, even when the system is trained in a fully-supervised manner.

Chapters 3 to 5 are based on the following publications:

Chapter 3: Semi-supervised Viewpoint Estimation with Geometry-Aware Conditional Generation. Octave Mariotti and Hakan Bilen. European Conference on Computer Vision workshop on Recovering 6D Object Pose. 2020.

Chapter 4: ViewNet: Unsupervised Viewpoint Estimation From Conditional Generation. Octave Mariotti, Oisin Mac Aodha and Hakan Bilen. International Conference on Computer Vision. 2021.

Chapter 5: ViewNeRF: Unsupervised Viewpoint Estimation Using Category-Level Neural Radiance Fields. Octave Mariotti, Oisin Mac Aodha and Hakan Bilen. British Machine Vision Conference 2022.

Chapter 2

General definitions and background

This chapter introduces and motivates the core concepts that are going to be discussed throughout the thesis.

2.1 Problem definition

Generally speaking, *pose estimation* refers to the task of predicting the position of a particular object in a scene with respect to the camera, or conversely, the position of the camera with respect to a frame of reference. A great number of works have been published claiming to tackle this problem, however, this term covers a wide array of specific tasks, which can be only vaguely related. As a result, the methods used to address them differ significantly, in their formulation, implementation and evaluation.

We can broadly identify three major aspects to classify pose estimation problems, summarized in Table 2.1.

Object type When estimating the pose of an object, the first aspect to consider is its rigidity. As defining a common frame of reference for deformable objects is a complex task, their pose is often estimated directly in image space, by predicting the position of

Object type	Rigid / Deformable
Reference frame	Scene-based / Object-based
Degrees of freedom	6D / 3D / 2D

Table 2.1: The different variants of the *pose estimation* problem

various landmark (Cootes et al., 1995; Kanazawa et al., 2016; Yang et al., 2017), or even as a dense prediction problem (Thewlis et al., 2017a). The most common occurrence of this problem is human pose estimation (Dalal and Triggs, 2005; Tompson et al., 2014; Newell et al., 2016; Cao et al., 2017), where the aim is to identify the position of different limbs of the person in an image. When dealing with rigid objects, on the contrary, it is possible to define a frame of reference with respect to a specific object instance or even a whole object category. Therefore, the task of pose estimation in this case refers to predicting, to various degrees, the relative rotation and translation between the reference and camera frame (Hinterstoisser et al., 2011, 2012; Xiang et al., 2014; Kehl et al., 2017; Rad and Lepetit, 2017; Liao et al., 2019; Mustikovela et al., 2020).

Reference frame A major distinction lies in how the reference frame is defined. In scene-based pose estimation, the goal is to recover the camera poses of multiple images taken from a single scene, with respect to an arbitrarily defined frame (Longuet-Higgins, 1981). This is traditionally done using Structure-from-motion algorithms, that learn to match similar image features across multiple views (Schonberger and Frahm, 2016). A more challenging setting is when the reference frame is defined for a whole category (Lowe, 1987; Huttenlocher and Ullman, 1987; Xiang et al., 2014), as models have to account for variations in shape and appearance of specific object instances among the category. In this case, the frame of reference is tied to arbitrary features of the object category, such as the ”front” side of a car.

Degrees of Freedom Depending on the data labeling and end task, different degrees of freedom can be considered. In a simplified setting, objects can be centered and viewed from a roughly constant distance, in which case only rotation, or joint positions in the deformable case, are predicted (Xiang et al., 2014). This follows the assumption that an object detector can be run to pre-process images, removing the redundant parts of the scene. In more complex settings, the complete 6 degrees of freedom - 3 rotations and 3 translations - can be estimated (Hinterstoisser et al., 2012). Because of the aforementioned labeling cost dataset can sometimes only be partially labeled, which leads to a coarser evaluation (Sedaghat and Brox, 2015).

2.1.1 Pose formalism

This work focuses on category-level rigid object pose estimation, meaning the models will be designed to predict the camera pose given an image of an object belonging to a predefined category, e.g. a car or a chair. More formally, given an image I of an object instance in pose p, the aim is to build a mapping f_p such that $f_p(I) = p$. As stated, p can differ depending on the data considered, and might be for instance a full camera matrix, a 3D rotation and/or a camera position, or even a simple pair of angles -azimuth and elevation - describing the viewpoint of I. As most 6DoF pose estimation systems follow a two-stage approach, first localizing the object in the image, then performing viewpoint estimation on the tightly cropped object (Rad and Lepetit, 2017), the main focus of this work will be on viewpoint estimation only, following the assumption that efficient object detectors already exist and training them does not require expensive data annotation (Ren et al., 2015; Long et al., 2015; Lin et al., 2017; Chen et al., 2017a; Redmon and Farhadi, 2018).

Therefore, the problem of pose estimation will be slightly simplified under the following assumptions:

1. Images show a single object instance

2. The instance is centered in the image
3. The images are cropped around the instance, in such a way that the apparent distance to the object is fixed.
4. The camera is held upright

These assumptions are used in the vast majority of synthetic datasets, and it is relatively easy to get a reasonable set of similar assumptions on real data by using an object detector. This allows us to reduce the problem of pose estimation to that of estimating the rotations of the object.

2.1.1.1 Rotation formalism

Rotations in three dimensions are known to form the special orthogonal group $SO(3)$. Due to its structure and properties, it has been extensively studied and as such, possesses multiple formalisms describing its elements. The most well-known ones include Euler angles, a set of three angles describing successive rotations about the x, y, and z axis, the axis-angle representations, composed of a unit vector describing the rotation axis and the angle of rotation, a unit quaternion which combines the axis and rotation angle in a single structure, and perhaps the most natural to perform computations with, rotation matrices. The choice of representation is particularly important as each has its own properties, which can be an advantage or drawback depending on the application. Table 2.2 briefly summarizes each representations properties from a viewpoint estimation standpoint.

While it is fairly common for dataset to be labeled with angles (Xiang et al., 2014; Sedaghat and Brox, 2015) as humans can easily grasp their meaning, trying to predict these values directly is likely to lead to large errors as this space exhibits a discontinuity - a rotation of 2π radians is also a rotation of 0 radians. Quaternions and rotations matrices are better representations due to their numerical properties, e.g. straightforward composition and stability. While quaternions are often seen as the most efficient representation and see many applications, e.g. describing the joint rotation in a robotic arm, matrices

Representation	Advantages	Drawbacks
Euler angles	• Easily interpretable • Common in hand-labeled datasets	• Discontinuous
Axis-angle	• Good interpretability	• Non-homogeneous (axis and angle represent different quantities) • Angle is discontinuous
Quaternion	• Efficient • Numerically stable	• Poor interpretability • Difficult to integrate in CV framework
Rotation matrix	• Numerically stable • Easy ML/CV integration as camera matrix	• Overparametrized • More expensive to compute

Table 2.2: Comparison of most common rotation representations

possess a key advantage that makes them a particularly good fit for this work: a camera pose is most commonly described using a camera matrix that contains a rotation matrix.

Trying to have a network predict a 3D rotation matrix directly is however not trivial due to the complex structure of such matrices (Zhou et al., 2019). In particular, its determinant is unlikely to be equal to 1. Given the assumptions that are made about the structure of the data, a natural solution to this issue consists in predicting the camera position in world coordinates. This makes recovering azimuth and elevation angles straightforward, and, using assumptions 2 and 4, easily translates to a camera matrix using a Gram-Schmidt orthonormalization process, otherwise known as a *LookAt* transformation. More precisely, using an arbitrarily defined upwards pointing vector u, the camera rotation matrix R can be recovered from the camera position p using Algorithm 1

Algorithm 1: Pseudocode for the LookAt algorithm

Input : camera position p, upwards vector u

Output: rotation matrix R

$l \leftarrow p/||p||$ ▷ normalize vector p

$s \leftarrow l \times u$ ▷ cross product

$s \leftarrow s/||s||$

$u \leftarrow s \times l$

$R \leftarrow [s, u, -l]^t$

On top of being relatively a straightforward process, this way of estimating poses through rotation matrices possesses desirable properties. First, matrix multiplication is at the core of most modern deep learning frameworks, meaning it can be integrated in a very natural way into deep learning systems. Second, the operations used to compute the full matrix are fully differentiable, hence, assuming p is the output of a neural network, rotation errors can directly be backpropagated to learn f_v

2.1.2 Metrics

Another particularity of pose estimation is that poses are continuous, although some approaches discretize them and transform the problem to a classification one (Tulsiani and Malik, 2015; Su et al., 2015b; Kanezaki et al., 2018; Kundu et al., 2018). Nonetheless, in order to evaluate how good a prediction p_{pred} is close to a target p_{gt}, it is necessary to have a notion of error in the pose space. As stated in Section 2.1.1, poses are in the most complex case made up of a 3D rotation R and translation T. For the translation part, a natural error is the Euclidian distance, that is:

$$err_T(T_{gt}, T_{pred}) = ||T_{gt} - T_{pred}||^2 \tag{2.1}$$

In certain cases, in order to be agnostic to the scale of the scene, translation errors can be

normalized by dividing it by a common value, in general defined as the average camera distance to the center of the scene. This helps compare performances when objects have different scales, i.e. a chair and an airplane, as the absolute translation error is likely to be much larger in the second case.

For the rotation part, there also exists a notion of rotation distance, although it is more involved than the Euclidian distance used for the translation. First, we can define the rotation difference between R_{gt} and R_{pred} by $R_{diff} = R_{gt}{}^t R_{pred}$. In linear transformation terms, this describes the rotation obtained by first applying R_{pred}, then applying the inverse of R_{gt}, and therefore captures this notion of difference, i.e. $R_{diff} = I_3 \iff R_{pred} = R_{gt}$. Then, the error can be defined as the angle θ of the rotation defined by R_{diff}, which can be computed using the formula $\theta = \arccos\left(\frac{Tr(R)-1}{2}\right)$, which gives:

$$err_R(R_{gt}, R_{pred}) = \arccos\left(\frac{Tr(R_{gt}{}^t R_{pred}) - 1}{2}\right) \tag{2.2}$$

Multiple metrics to compare rotations have been defined, however, using the angle of the relative rotation has been shown to possess the best properties and has been extensively used in recent publications (Huynh, 2009; Tulsiani and Malik, 2015; Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018; Meng et al., 2021)

Multiple aggregation strategies can be used to obtain a single value over a complete evaluation set $\{I_n\}_{n=1,\dots,N}$. While the arithmetic mean is a natural aggregate that is commonly used in machine learning, the specific structure of the rotation space led some authors to favour the median (Tulsiani and Malik, 2015), as it is more robust to outliers.

To aid in quickly grasping the performance of a system, it can be useful to define an accuracy-type metric, by using an angular threshold. For instance, rotational accuracy at 30° is defined as:

$$Acc@30^\circ = \frac{1}{N}\left|\left\{err_R(R_{gt_n}, R_{pred_n}) \leq \frac{\pi}{6} | n = 1, \dots, N\right\}\right| \tag{2.3}$$

Where the $|\cdot|$ operator returns the cardinality of the set. As there is no standard metric used in all pose estimation works, the results presented in this work will be reported under different metrics depending on the data labeling and the ones used in related works.

2.2 Unsupervised Learning of 3D information

As the goal of this thesis is to present ways to estimate the pose of objects without manual supervision, this section discusses the relevant methods that will serve as foundations of the work carried out in the next chapters.

2.2.1 Rigid object pose estimation

Object pose estimation is a long-standing objective of computer vision, and as such, has been explored for decades. However, only very recent studies show the possibility of estimating poses without labels. Historically, most methods were completely supervised.

2.2.1.1 Pre-deep learning methods

While deep learning methods have completely changed the way computer vision research is conducted, pose estimation has been discussed in classical vision methods. The general framework of these classical models is a two-stage process: first, image features like edges, gradients, or blobs are extracted, then, these are compared with predefined templates of specific objects (Lowe, 1987; Johnson and Hebert, 1998; Pope and Lowe, 2000; Weiss and Ray, 2001; Hinterstoisser et al., 2011, 2012). The template can come from captured views of the object, or in more recent methods, from renderings of 3D CAD models of the specific objects. Interestingly, recovering the pose was only a byproduct of the system in early methods, as the main goal was to classify the object category.

The main drawback of these methods is their reliance on templates to compare with. This is the source of multiple issues. First, it requires collecting a very large number of

views for each specific object instance, often multiple thousands, which is only reasonable using synthetic data. If precise pose needs to be recovered, these templates also need to be labeled. Second, and perhaps more importantly, they lack generalization to new object instances, as predefined templates will poorly match unseen instances. For specific categories where variations are reasonably controlled and enough data is available, deformable templates can be applied with good performances. A particularly relevant example is that of human faces, where such morphable models still achieve competitive results even though they were first developed twenty years before (Blanz and Vetter, 1999; Egger et al., 2020). Nonetheless, if the object is known beforehand, and obtaining labeled views is not a limiting factor, then these can be a relatively inexpensive and fast way to recover pose. Recent development shows these generalize well to complex scenarios like occluded views (Nguyen et al., 2022).

2.2.1.2 Deep models

Later models tend to split the task into object localisation and rotation estimation, and to replace low-performance machine learning algorithms with CNNs. Localisation is performed using pretrained models while rotation is recovered either by 3D bounding cube regression (Rad and Lepetit, 2017; Tekin et al., 2018) or viewpoint classification (Kehl et al., 2017).

Now that reasonable performances have been attained on a controlled setup, focus has shifted towards specificities of the task, such as avoiding errors caused by symmetries and lowering the necessary supervision by removing depth maps (Rad and Lepetit, 2017), or targeting real-time performances (Kehl et al., 2017). Direct regression over the rotation space has also been explored and proved to be on par with other ways of recovering rotations (Mahendran et al., 2017), while multitask approaches have also been shown to help by adding keypoint detection for instance (Tulsiani and Malik, 2015).

As training labels on real images are expensive, many approaches try to reduce supervision. The most common approach is to use synthetic data obtained from 3D CAD models

(Sundermeyer et al., 2018; Tan et al., 2018; Su et al., 2015b), however, these methods tend to have issues generalizing to real data. Other works explore using extremely limited sets of poses (Rhodin et al., 2018; Kanezaki et al., 2018), mainly because of dataset restrictions. Nonetheless, CAD models do not completely remove the need for pose labels, as they are considerably more expensive to produce compared with images.

The most popular dataset, ShapeNet (Chang et al., 2015), contains handcrafted 3D models from a few categories. Each of these took a substantial amount of time to create, and therefore, the cost of developing a CAD model dataset for a new category is much high than that of simply gathering pictures. Another alternative is to scan real objects in order to recover a 3D model of them, but this requires equipment and post-processing whose cost is comparable with that of annotating views.

2.2.2 Low supervision pose estimation

The advent of deep learning also enabled the design of models that deviate from the fully supervised regime by making use of unlabeled samples or noisy labels. DC-IGN (Kulkarni et al., 2015) composes a batch of images where only a single pose parameter is changing, e.g. the camera elevation, and uses this to create interpretable value in a latent space. Sundermeyer et al. (2018) propose to train a denoising autoencoder whose latent space can be mapped to viewpoints using only a few samples. More standard ways of performing semi-supervised learning have also been explored, like pseudo-labeling using feature matching (Tseng et al., 2019; Wang et al., 2021a).

More recently, some works have pushed pose estimation to new challenging settings, like zero-shot pose estimation, in which the task is to estimate pose for objects whose category is unseen at training time, by learning to align views of a wide variety of categories (Banani et al., 2020), or by finding correspondences in the features of a large pre-trained model (Goodwin et al., 2022).

2.2.3 Equivariance learning

A possible way of recovering 3D information from images is through the use of equivariant embedding. Equivariance is a concept used in multiple settings with different meanings. For the purpose of this work, we can define an equivariant representation as a representation that follows the same transformation as its source. Formally, a mapping f taking as input x is equivariant to a class of transformations $\mathcal{T}$ if and only if

$$\forall x, \forall T \in \mathcal{T}, f(T(x)) = T(f(x)) \tag{2.4}$$

An example of equivariance in computer vision is an object detector, that must be equivariant to image translations. In a slightly looser sense, a pose estimator is a system equivariant to object translation and rotations. Hence, a natural direction to recover pose without the use of labels is through the use of equivariant representations.

2.2.3.1 General equivariance

Equivariant representations have been proposed on 2D images as a way to capture the data structure in order to build more robust systems. A particular example is that of keypoint detection, as those follow specific object features, and are therefore equivariant to the transformations applied to the objects. Keypoints have been studied extensively, as tools to align different instances (Cootes et al., 1998), or more recently simply for representation purposes (Zhu and Ramanan, 2012). Recent advances allowed keypoints to be automatically discovered, through matching predictions under known transformations (Thewlis et al., 2017b).

Denser representations have also been proposed, most notably the spatial transformers (Jaderberg et al., 2015), that learn to transform feature maps in convolutional neural networks, in order to guarantee their equivariance. Unsupervised dense generalizations of keypoints were also proposed (Thewlis et al., 2017a, 2018).

All of these methods however are restricted to image transformations and therefore are not suitable to estimate 3D poses.

2.2.3.2 3D equivariance

Generalizing equivariant representations to 3D transformation is a natural step towards enabling models to be aware of the 3-dimensional structure of the world. Both keypoint discovery (Suwajanakorn et al., 2018; Jakab et al., 2021) and spatial transformers (Worrall et al., 2017) have been extended to work with 3D rotations. Specifically-designed 3D equivariant representations have been proposed, most notably spherical CNNs (Cohen et al., 2018; Esteves et al., 2017, 2019) which have been adapted for a variety of tasks, like segmentation of 360° images (Jiang et al., 2019), modeling climate event on a spherical representation of the earth (Defferrard et al., 2020), or predicting protein folding (Boomsma and Frellsen, 2017). These models are designed to operate on a spherical signal and therefore possess rotational equivariance by design.

A shared feature of all these models is their need for a form of pose supervision, be it explicit, in the form of a training objective, or implicit, by requiring canonically aligned 3d models. This brings the question of how to train such an equivariant representation without requiring pose supervision. A potential solution lies in the most obvious 3D equivariant representation there is: a complete 3D reconstruction model

2.2.4 3D reconstruction

Reconstructing three-dimensional objects is one of the main fields of computer graphics, either for the purpose of *modeling*, i.e. capturing volumetric data, or for the converse *rendering* operation that consists in producing images from 3D data. While 2D-based computer vision is one of the fields that gave birth to modern deep learning, 3D has been lagging behind and only recently came to prominence as a major topic in learning-based vision. A main factor in this delay is the computational cost of dealing with an extra dimension, as well as a more costly data gathering process. Nonetheless, 3D methods are

today one of the most active areas in deep learning, and recent advancements improved the quality of models by a large margin.

2.2.4.1 Learning-based multi-view reconstruction

A major challenge in integrating 3D reconstruction in a learning environment is making them compatible with the learning objective. Since deep learning uses gradient-based optimization, this means making operations differentiable, which is not guaranteed by 3D libraries. Therefore, most approaches chose to adapt deep learning methods to work with 3D data, like 3D convolutions (Wu et al., 2015; Su et al., 2015a), which were later adopted for rendering (Tulsiani et al., 2018). Most 3D modeling with deep learning uses one of four structures among voxels, point clouds, meshes and implicit representations (Table 2.3). More involved representations are possible, like octrees (Yu et al., 2021a) or spatial hashes (Müller et al., 2022), but these tend to be marginal as they are not easily compatible with deep learning models.

Although there were attempts to use learning-based models to reconstruct 3D data, like neural renderers (Nguyen-Phuoc et al., 2018; Olszewski et al., 2019), a learning-based rendering strategy tends to exhibit poor consistency, steering more recent models towards analytical rendering operation (Jimenez Rezende et al., 2016). Voxels (Yan et al., 2016; Tulsiani et al., 2017, 2018; Sitzmann et al., 2019a), point clouds (Fan et al., 2017; Lin et al., 2018; Insafutdinov and Dosovitskiy, 2018), and meshes (Kato et al., 2018; Kanazawa et al., 2018; Riegler and Koltun, 2020) have extensively been used to learn 3D representations of objects from images, with and without known camera pose. These approaches have mostly been superseded by neural field-based representations - referred to here as *implicit representations* - that encode spatial information in the weights of a neural network, while keeping analytical rendering to enforce 3D consistency (Park et al., 2019; Mescheder et al., 2019; Sitzmann et al., 2020; Niemeyer et al., 2020; Yen-Chen et al., 2021), although these models seem more complex to train. Being completely learning-based, they are more prone to overfitting and tend to require more computing power.

method	advantages	drawbacks
voxel	• Natural generalization of images • Easy ML integration	• Cubic complexity scaling in both memory and computation
point cloud	• Lightweight • Natural format for acquisition (e.g. lidar)	• Poor reconstruction due to data sparsity
mesh	• Extensively used in computer graphics • Good complexity/quality ratio	• Poor ML integration due to complex format
implicit	• Currently SotA in reconstruction • Easy ML integration	• Expensive to visualize • Complex training

Table 2.3: Comparison of classical Deep Learning 3D reconstruction models

2.2.4.2 3D generative models

A popular alternative approach to multi-view reconstruction is the use of generative methods. A major issue in applying multi-view approaches is precisely the requirement for multiple images of the same scene. Generative approaches are a way to solve this constraint, mainly in their adversarial form by replacing the hard reconstruction objective with a soft adversary fooling objective. This has been applied to neural rendering (Nguyen-Phuoc et al., 2019), voxels (Wu et al., 2016; Henzler et al., 2019), and to implicit representations (Schwarz et al., 2020; Niemeyer and Geiger, 2021; Gu et al., 2021).

Chapter 3

Principles of low-supervision viewpoint estimation

3.1 Introduction to analysis-by-synthesis models

Before attempting to build a fully unsupervised pose estimation system, it can be useful to relax the constraints and work in a low-supervision setting. This will help defining and motivating the main building blocks used throughout this work, as well as identifying the issues such a system can run into. Therefore, the first focus will be to design a method to operate in a semi-supervised setting and experiment with multiple training scenarios to determine the requirements for unsupervised operation.

The principal question that will be addressed in this chapter is that of leveraging unlabeled samples. The question of semi-supervised learning, that is, using a training set containing both labeled and unlabeled samples is long-standing in machine learning (Weston et al., 2008; Ranzato and Szummer, 2008; Lee et al., 2013; Rasmus et al., 2015;

The main findings of this chapter have been published in the ECCV 2020 workshop on recovering 6DoF object pose (Mariotti and Bilen, 2020).

Tarvainen and Valpola, 2017; Laine and Aila, 2017; Sohn et al., 2020), however, the typical semi-supervised framework is image classification, meaning these methods might perform poorly when applied to pose estimation.

Concurrently, methods have been developed to learn interpretable representations using an analysis-by-synthesis approach (Worrall et al., 2017; Jakab et al., 2018; Rhodin et al., 2018). The common point between these approaches is their use of a reconstruction process - the *synthesis* - in order to recover some latent parameter from the data -the *analysis*. To this end, these methods usually force the data through a task-specific representation, e.g. keypoints or viewpoints, and attempt to reconstruct the original samples from it. While they could in theory be used to recover object poses, no method has yet been able to learn such representations of complex 3D rotations without pose supervision. Still, if such representations are already available and do not need to be jointly learned with viewpoint, as is the case for human skeletons and morphable models, recovering pose becomes possible.

A natural idea is to combine these two frameworks, by using a small labeled set to train a viewpoint estimator, and integrating this in a larger analysis-by-synthesis pipeline that learns to reconstruct images in order to obtain a supervision signal from unlabeled samples. In a fully supervised setup, the reconstruction task provides a helpful auxiliary objective that can prevent overfitting. If some images lack annotation, reconstruction can still provide a supervision signal, as an incorrectly estimated viewpoint is likely to lead to poor reconstruction. By studying the limit cases where labels are very scarce to completely absent, it should be possible to identify the problems encountered in the unsupervised regime and work towards a solution.

More precisely, the proposed approach (illustrated in Fig. 3.1) takes in a pair of images that contain an object captured from different viewpoints, encodes the appearance of the object in the first image and estimates the viewpoint in the second image, then produces a reconstruction of the second image by combining the extracted viewpoint with the ap-

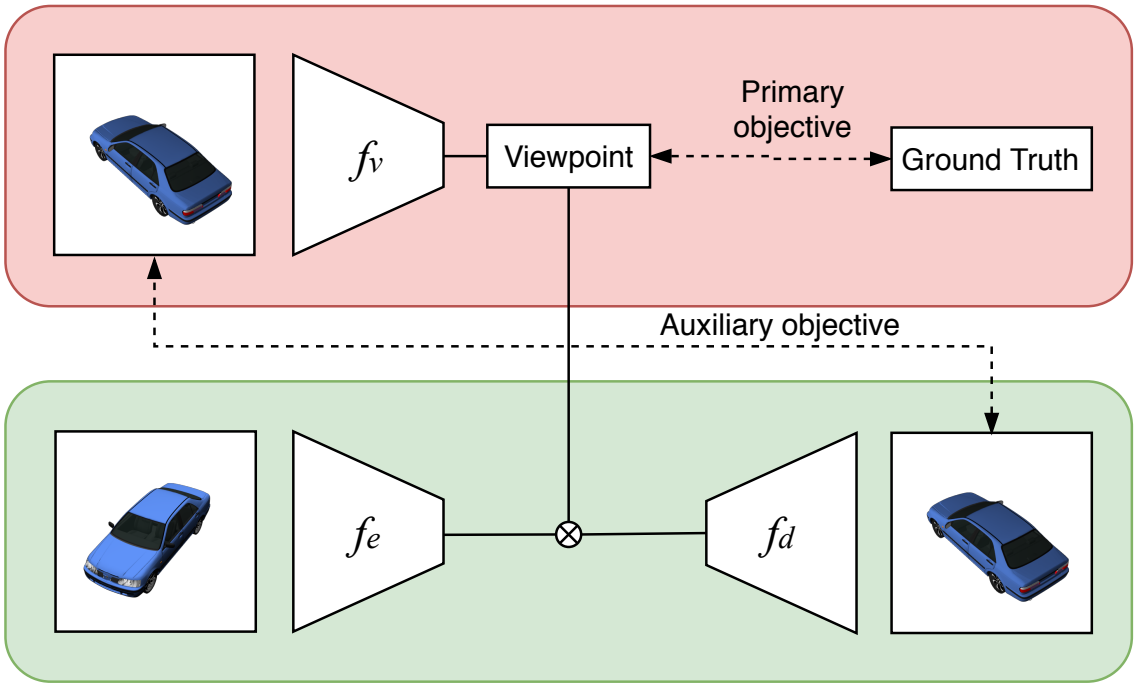


Figure 3.1: Overview of the semi-supervised framework. Our primary objective is to learn the camera viewpoint from the picture of an object. Given another picture of the same object, we also reconstruct the first using conditional generation to provide additional supervision.

pearance embedding.

While the conditional generation from image pairs forces the network to learn factorized representations for appearance and viewpoint without explicit supervision, there is a high degree of ambiguity for representing the viewpoint in a deep neural network and no guarantee that the learned representation corresponds to anything interpretable. Therefore, the architecture is built around a rotation equivariant space, designed to produce views from any viewpoint by simply applying a 3D rotation to the features. Quantitative and qualitative results show that the method can effectively leverage the information in unlabeled images, improves viewpoint estimation with limited supervision over regression baselines and outperforms the state-of-the-art semi-supervised methods in a standard viewpoint estimation benchmark.

3.2 Related work

Early supervised pose estimation. The early models proposed in object pose estimation use classical computer vision techniques, and rely on matching image features like gradients or surface normals with predefined templates, either recovered from the object itself in a controlled setup or by using 3D CAD models to obtain rough estimates (Hinterstoisser et al., 2011, 2012). These methods require pose supervision and have limited applicability due to their lack of generalization.

Recent supervised pose estimation. More recent methods typically split the pose estimation into two sub-tasks, object localization and rotation estimation, and use a CNN for each. Localization is most often performed using pretrained models - e.g. r-CNN (Girshick, 2015), while rotation is recovered either by 3D bounding cube regression (Rad and Lepetit, 2017; Tekin et al., 2018; Grabner et al., 2018) or viewpoint classification (Kehl et al., 2017). As excellent performances have been reported on controlled setups, focus has shifted towards specifications of the task, such as avoiding errors caused by symmetries and lowering the necessary supervision by removing depth maps (Rad and Lepetit, 2017), or targeting real-time performances (Kehl et al., 2017). Direct regression over the rotation space has also been explored and proved to be on par with other ways of recovering rotations (Mahendran et al., 2017; Liao et al., 2019), while multitask approaches have also been shown to help by adding keypoint detection for instance (Tulsiani and Malik, 2015; Zhou et al., 2018). Viewpoint estimation specific architectures that go beyond generic CNNs are also getting proposed, in an effort to tailor neural networks to the characteristics of the task (Joung et al., 2020; Cohen et al., 2018; Esteves et al., 2017, 2019).

Pose estimation from synthetic data. As training labels on real images are expensive to obtain, many works try to reduce the cost of supervision. The most common approach is to use synthetic data obtained from 3D CAD models (Sundermeyer et al., 2018; Tan

et al., 2018; Su et al., 2015b), however, these methods tend to have issues generalizing to real data. Other works explore using extremely limited sets of poses (Rhodin et al., 2018; Kanezaki et al., 2018), mainly because of dataset restrictions.

3D reconstruction. Another line of work focuses on producing a 3D reconstruction of the object from 2D views by geometry-aware deep representations. However, as they are only interested in the 3D-aware representation, they tend to consider pose information as an already-acquired supervision. Nonetheless, several recent works show that pose supervision is not strictly required to produce a 3D model, either voxel-based (Yan et al., 2016; Yang et al., 2018; Tulsiani et al., 2018), point clouds (Insafutdinov and Dosovitskiy, 2018), or 3D meshes (Kato et al., 2018). The mesh approach was also extended in an unsupervised way (Kanazawa et al., 2018). In this case, pose is learned jointly with the reconstruction and supervision is done by rendering the 3D shape into a silhouette. The 2D image obtained is then compared with the ground-truth segmentation mask. These works involve heavy networks to deal with full 3D representations and a complex differentiable rendering stage. In contrast, we aim for a fully convolutional, more flexible architecture.

Geometry-aware representations. Another related line of approaches involves producing lighter-weight representations that describe the geometry of the object while being sensitive to pose. Often, these are designed following an equivariance principle, that is, applying a transformation e.g. a rotation to the object will have the effect of transforming the representation in a similar way. Precursory works specifically targeting equivariance rely on autoencoding architecture and constrain the encoding to respect the structure of the data (Kulkarni et al., 2015). A more involved approach consists in entangling a learned embedding with a rotation. This has first been proposed on feature maps and 2D rotations (Jaderberg et al., 2015), then adapted to general representations (Worrall et al., 2017) and applied on full 3D rotations (Rhodin et al., 2018), albeit with a very restricted set of poses. Other rotation-specific equivariant representations were also designed by adapting CNNs

to operate on spherical signals (Cohen et al., 2018; Esteves et al., 2017, 2019). These spherical CNNs rely on heavy 3D supervision and typically operate on a coarse scale due to their use of Fourier transform, but their construction guarantees good results on rotation estimation.

Keypoint-based methods. Keypoints are a natural equivariant representation: they describe the pose and it is intuitively possible to discover them without supervision. 2D keypoints have been discovered on humans and faces with (Jakab et al., 2018) or without (Thewlis et al., 2017b) reconstruction. 3D keypoints are used in the case of full 3D rotations (Suwajanakorn et al., 2018), however, no approach has been shown to reliably estimate them without strong pose supervision. Mapping the image pixels to a sphere has also been explored as a continuous generalization of keypoints, but this technique faces the same issues as its discrete counterpart (Thewlis et al., 2017a, 2018).

Generative-based methods. Recent advances in generative adversarial networks have allowed frameworks to learn geometry-aware representations, through the generation of images under different viewpoint (Nguyen-Phuoc et al., 2019; Mustikovela et al., 2020). These methods are still experimental and are still subject to a certain degree of unsuitability, but show a promising and novel angle of attack on viewpoint estimation.

3.3 Method

3.3.1 Supervised viewpoint estimation

Assume that we are given a set of m labeled images with their ground-truth viewpoints $\mathcal{T} = \{(I_i, v_i)\}_{i=1}^{m}$, where $I \in \mathcal{I}$ is an RGB image and $v = (v^1, v^2, v^3) \in \mathcal{V}$ is a 3-dimensional vector describing the viewpoint as a rotation on $\mathbb{R}^3$. There exist several ways to represent v, most common methods being a triplet of angles describing azimuth, elevation and in-plane rotations, an axis-angle representation, a unit quaternion or a rotation matrix. To

simplify the learning procedure, we model v by a normalized three-dimensional vector interpreted as the camera position in world coordinates as described in Section 2.1.1. The camera is thus assumed to lie on the surface of a unit sphere, pointing towards the center. We wish to learn a mapping from an image to its viewpoint $f_v : I \rightarrow \mathcal{V}$ such that $f_v(I;\theta_v) = v$ where θ_v are the parameters of f_v. One can learn such a mapping by minimizing the following empirical loss over the set $\mathcal{T}$ w.r.t. θ_v:

$$\sum_{(I,v)\in\mathcal{T}} ||f_v(I;\theta_v) - v||^2. \tag{3.1}$$

3.3.2 Geometry-aware representation

We are also given a set of n unlabelled image pairs $\mathcal{U} = \{(I_i, I_i')\}$ where each pair contains two images of an object instance (e.g. airplane, car, chair) that are captured at two different viewpoints. We assume that the ground-truth viewpoints of the images are not available and we wish to improve the performance of our viewpoint predictor f by leveraging the information in the unlabeled images.

A commonly used tool for unsupervised learning is an autoencoder architecture that encodes its input I into a low dimensional encoding $f_e(I;\theta_e)$ via an encoder network E and maps the encoding to the input space, i.e. $f_d(f_e(I;\theta_e);\theta_d)$, via a decoder network f_d to reconstruct the input. The encoder and decoder are parameterized by θ_e and θ_d respectively. Although autoencoders can successfully be utilized to learn informative representations that can reconstruct the original image, there is no guarantee for the embeddings to encode the 3D viewpoints of objects in a disentangled manner.

One solution to relate an embedding of an object in image I to its viewpoint v involves a conditional image generation technique. This was first proposed in (Worrall et al., 2017) for in-plane rotations and extended in (Rhodin et al., 2018) for 3D ones, In particular, given an image pair I and I' that contain the same object viewed from two different points and also given the viewpoint from which the object is seen in the images, this method couples the viewpoint and the appearance of the object in the encoding. To this end, the

embedding of image I, $f_e(I)$ is transformed by using the rotation $R(v')$ where v' is the viewpoint in image I' and $R(v') \in SO(3)$ computes the rotation matrix associated to v'. The rotated embedding is then decoded, i.e. $f_d(R(v') \times f_e(I;\theta_e));\theta_d)$, to reconstruct not the input I but I' by minimizing the following loss w.r.t. the parameters of the encoder and decoder:

$$\sum_{(I,I',v')\in\mathcal{U}} ||f_d(R(v') \times f_e(I;\theta_e));\theta_d) - I'||^2 \tag{3.2}$$

where the output of the encoder $f_e(I;\theta_e)$ is designed to be $3 \times k$ dimensional such that it can be rotated by the rotation matrix $R(v')$.

This presents a slight variation over the framework in (Rhodin et al., 2018) as the rotation here is absolute instead of relative. This means that the embedding $f_e(I;\theta_e)$ should represent the object from a canonical viewpoint instead of the one from which it appears in I.

This formulation enables the method to learn a "geometry aware" representation that can relate the viewpoint difference in 3D space to its projection in pixel space. However, it requires the ground-truth viewpoint for each image I', which limits the applicability of the method to supervision-rich setups. To address this limitation and extend the learning of the geometry-aware representations to image pairs with unknown viewpoints, we propose an analysis by synthesis method. To this end, we predict the viewpoint as $\hat{v} = f_v(I;\theta_v)$ for I by using the viewpoint estimator f, and substitute it with $R(v')$ in Eq. (3.2):

$$\sum_{(I,I')\in\mathcal{U}} ||f_d(R(f_v(I';\theta_v)) \times f_e(I;\theta_e));\theta_d) - I'||^2 \tag{3.3}$$

This formulation models the reconstruction loss as a function of viewpoint predictor f and therefore allows the gradients to flow in the pose regression network without any viewpoint supervision. Furthermore, working with absolute viewpoints not only allows a more straightforward optimization as we only need one viewpoint estimation whereas two would be needed to compute a relative pose. It also makes learning an encoding easier as it factors out the burden of estimating the pose.

3.3.3 Semi-supervised viewpoint prediction

Our hypothesis is that successful reconstruction of I' requires an accurate viewpoint estimation. However, given high-capacity encoder and decoder architectures, accurate viewpoints enable high-fidelity reconstructions, the converse is not necessarily true as the viewpoints in the encoding can be represented in infinitely different ways and there is no guarantee that the learned viewpoints for the images will match with their ground-truth view. For instance, the output of the viewpoint estimator can be distributed between 0 and π for each angle instead of the entire range of $[0, 2\pi)$ or the angles can be mapped to a non-linear and uninterpretable space, while the network preserves its reconstruction performance. Thus, we propose a semi-supervised formulation in which the estimated viewpoints are regularized as below by optimizing the combined loss terms in Eq. (3.1) and Eq. (3.3):

$$\min_{\theta_v,\theta_e,\theta_d} \sum_{(I,v)\in\mathcal{T}} ||f_v(I;\theta_v) - v||^2 + \lambda \sum_{(I,I')\in\mathcal{T}\cup\mathcal{U}} ||f_d(R(f_v(I';\theta_v)) \times f_e(I;\theta_e));\theta_d) - I'||^2 \quad (3.4)$$

where λ is a trade-off hyperparameter between the supervised and unsupervised loss terms. In words, the formulation allows gradients for the unsupervised loss to flow in the viewpoint network f_v, and the supervision imposed on the viewpoint space in turn constrains the learned representation to capture the structure of the object.

The supervision provided by the reconstruction task brings up the question of unsupervised viewpoint estimation using no pose labels. While theoretically possible, we find that it is likely to fail in complex scenarios, as the supervision signal is too weak to provide good viewpoint supervision. In particular, symmetries in real-world objects push the learned pose towards degenerate solutions. This is further demonstrated in Section 3.4.4 and 3.4.6.

3.4 Experiments

3.4.1 Dataset

We use the popular Shapenet (Chang et al., 2015) dataset that consists of a large bank of 3D CAD models, classified in different object categories. This makes obtaining a large number of views spreading various viewpoints fairly straightforward, as well as acquiring several views of the same object, a feature often absent in other 3D datasets like Pascal3D (Xiang et al., 2014). Because we render the 3D models, we automatically know the ground truth viewpoint as well, making data labeling a triviality. We mainly focus on three object categories, aeroplanes, cars and chairs, as they offer enough models to build a diversified image dataset. For each category, we render each model with 10 randomly selected viewpoints, with azimuth ranging the complete 360° rotation and elevation selected from -20° to 40°. The final datasets contain 40,460; 36,760 and 67,790 images for the aeroplane, car and chair categories respectively. We split the data in training, validation and testing sets, accounting for 70, 10 and 20 percent of the whole dataset respectively. To simulate a semi-supervised setup, we further split the training set by randomly selecting a subset of the data to act as the labeled set, the rest acting as unlabeled. We adjust the ratio of labeled samples in our experiments to show the effect of varying degrees of supervision. The splits are made on a model basis, that is, the different views from the same 3D model are either all labeled or all unlabeled.

To evaluate our framework, we use two popular metrics in viewpoint estimation (Tulsiani and Malik, 2015; Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018), the accuracy at 30°, and the median angular error in degrees. The accuracy is computed as the ratio of predictions within 30° of the ground truth viewpoint and gives a rough estimate of the network performances. The aggregator for angular error is chosen to be the median rather than the mean as it is less biased by outliers which are common in pose estimation due to symmetries.

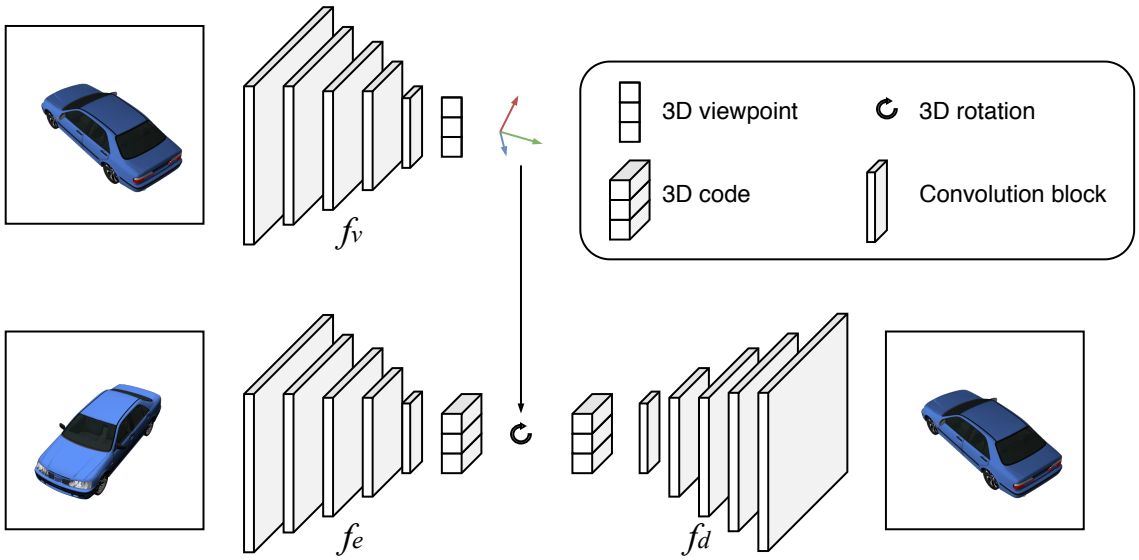


Figure 3.2: Detailed architecture of the network. The viewpoint estimator f_v outputs a normalized 3D vector interpreted as the camera position. This prediction is transformed into a rotation matrix, which is used to rotate the code produced by the encoder f_e. This rotated embedding is given to the decoder f_d to reconstruct the original image.

3.4.2 Implementation details

We model f_e, f_d and f_v with convolutional neural networks. We use a simple design, stacking several convolutional blocks with batch normalization and ReLU activation function. The encoder network has five blocks each consisting of two convolutions layers, with the second of each block using a stride of 2 in order to reduce the spatial dimension. All layers use 3 by 3 convolutions with channel count starting at 32 and doubling each block. On top of this, we use a fully connected layer to obtain the embedding. In order to interpret it as a geometric representation, we group the embedding values by triplets, effectively creating a collection of points in 3D space. This representation can then be rotated using the viewpoint rotation matrix. The architecture of f_d is simply a mirrored version of that of f_e. A schematic version of our framework is presented in Fig. 3.2, and specific network architectures are reported in Table 3.4 at the end of this section.

For the reconstruction objective, we use perceptual loss (Johnson et al., 2016), as it pro-

Method	Labels (%)	aeroplane		car		chair	
		Acc (%, ↑)	Err (°, ↓)	Acc (%, ↑)	Err (°, ↓)	Acc (%, ↑)	Err (°, ↓)
Regression	100	**87.3**	6.9	89.3	6.2	88.9	8.4
Ours	100	**87.3**	**6.1**	**91.4**	**4.6**	**89.7**	**7.8**
Regression	25	80.7	8.9	79.4	9.8	80.8	12.2
Ours	25	**84.9**	**6.4**	**86.6**	**5.8**	**86.2**	**8.5**
Regression	10	75.6	12.1	72.3	13.1	71.8	16.5
Ours	10	**83.2**	**6.5**	**83.7**	**6.4**	**81.0**	**9.4**
Regression	5	70.4	15.1	65.9	17.7	68.4	19.2
Ours	5	**81.4**	**7.4**	**73.8**	**9.0**	**76.3**	**15.1**
Regression	1	54.2	29.5	45.1	36.3	59.1	28.6
Ours	1	**64.9**	**17.1**	**62.4**	**14.5**	**57.9**	**25.1**

Table 3.1: Viewpoint prediction in terms of accuracy and error rates for varying label supervision. Regression denotes a supervised trained network trained on the corresponding proportion of labeled data. Acc: accuracy at 30°, Err: median angular error.

vides supervision of higher quality, translating to better learning signals for the viewpoint estimation. All training is done with the ADAM optimizer (Kingma and Ba, 2014) with default parameters and a batch size of 64. To prevent potential overfitting caused by the reconstruction task, we use early stopping, halting the training when no improvements are observed on the validation set for 30 epochs. We set the hyperparameter λ to be equal to the ratio between labeled and unlabeled samples. This way, when summed over the whole sets, the contributions of both losses are evened out.

3.4.3 Viewpoint estimation

We compare the results of our method with a simple regression baseline, as well as Mean Teacher, a state-of-the-art semi-supervised approach. Though it was proposed for classification, it is a generic approach that can therefore be extended to viewpoint regression. Training is done using 10 views per model, with varying degrees of supervision. The baseline is simply set as a viewpoint estimator without any added secondary objective, in order to study the effect adding reconstruction to the framework has.

The quantitative results in Table 3.1 show that our method outperforms simple regression in all cases. Unsurprisingly, performances are directly correlated with the amount of labeled data for all methods. It is worth noting that even when using 100% of the labels, our method still outperforms simple regression, showing that simply adding a reconstruction task helps refine the network predictions. However, the gap in performance increases more when lowering supervision, as the regression task is losing training samples while ours can still leverage them in a self-supervised way, demonstrating the effectiveness of reconstruction as a proxy for viewpoint estimation. When training with very low supervision, performances tend to drop sharply, as symmetries in the object make viewpoint estimation too difficult, and the reconstruction task becomes less effective. Indeed, producing an image from a symmetric viewpoint still provides decent minimization of the reconstruction loss. A significant failure of our system can be observed when using only 1% of the labels on the chair category, which comprises only 470 labeled images. Further details are discussed in Section 3.4.4.

We are also able to outperform mean teacher (Tarvainen and Valpola, 2017), demonstrating how building a problem-specific approach can easily lead to better performances (Table 3.2). Mean teacher relies on prediction consistency over the unlabeled set, using averaged models to predict soft targets. This constrains the learning procedure to be stable during training, making predictions more reliable. However, reliably wrong predictions will not be detected, in which case the unlabeled set is of no help. This is a common

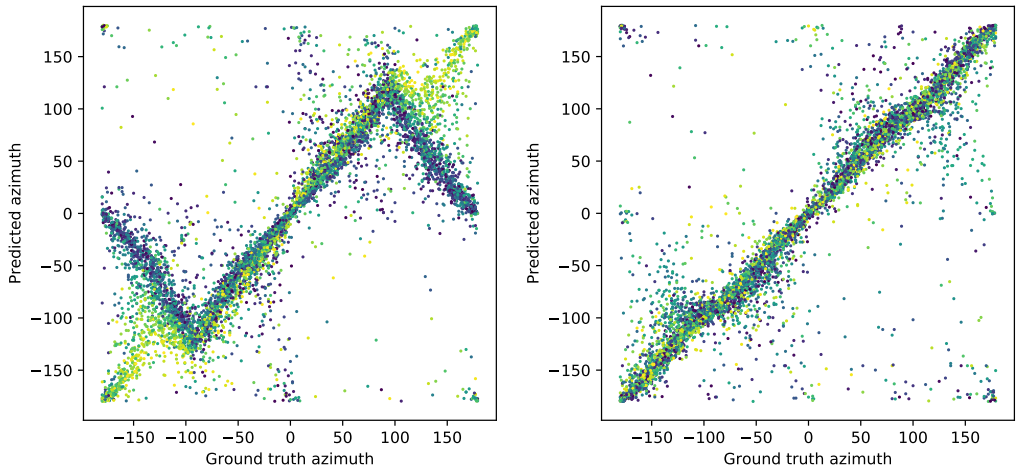
Method	labeling ratio(%)	aeroplane		car		chair	
		Acc (%, ↑)	Err (°, ↓)	Acc (%, ↑)	Err (°, ↓)	Acc (%, ↑)	Err (°, ↓)
Mean teacher	10	81.4	10.3	72.4	13.8	68.9	19.0
Ours	10	**83.2**	**6.5**	**83.7**	**6.4**	**81.0**	**9.4**
Mean teacher	1	28.9	44.0	8.5	67.7	34.3	39.5
Ours	1	**64.9**	**17.1**	**62.4**	**14.5**	**57.9**	**25.1**

Table 3.2: Comparison to the Mean Teacher (Tarvainen and Valpola, 2017) in terms of viewpoint accuracy and error rate. Acc: accuracy at 30°, Err: median angular error.

pitfall in viewpoint estimation because of the symmetries. In contrast, our method always provides a supervision signal in case of wrong reconstruction, effectively alleviating the issue. Similarly to our approach, mean teacher tends to fail when supervision is scarce, as illustrated by its results with 1% supervision.

3.4.4 Prediction analysis

An interesting phenomenon can occur when the supervision is low: the symmetries of the object will cause the emergence of degenerate solutions. If we consider a pair of images from two symmetric viewpoints, not only is it easy to mistake one viewpoint for the other when trying to learn it, reconstructing the wrong image is also not very penalizing. Those effects combined can push the network towards a local minimum from which escaping becomes impossible, as the reconstruction objective is likely to push the viewpoint estimation back. Fig. 3.3a shows this behavior with chairs, as 1% supervision sees the predicted azimuth ping back and forth when it should complete a full rotation. Increasing the supervision solves this issue, though we can still spot the occasional mistake (Fig. 3.3b).



(a) Ours, 1% labels

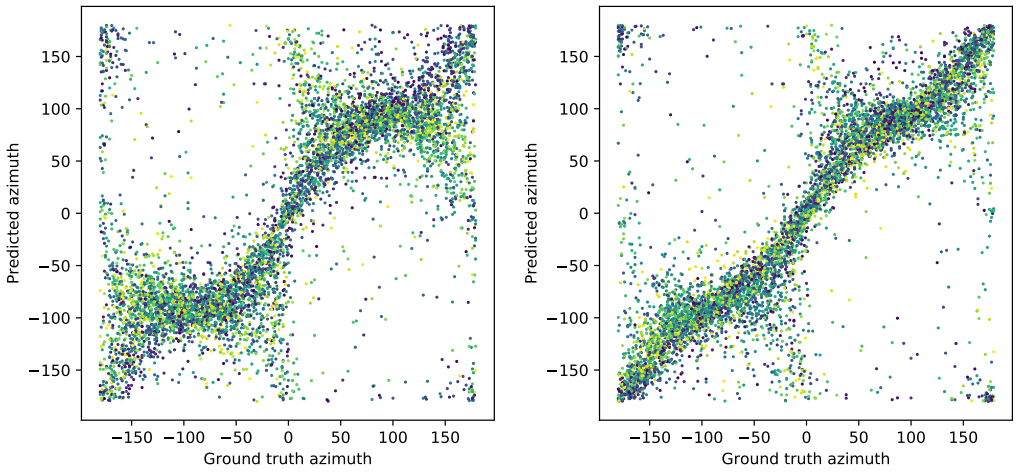
(b) Ours, 10% labels

Figure 3.3: Predicted vs ground truth azimuth for our method on test samples. Each point is colored with ground truth elevation.

We also compare with predictions of a simple regression model. We can see on Fig. 3.4 that while the global structure of predictions is similar, a simple regression involves more noise in the labels. In contrast, the predictions from our method are much finer, as the additional reconstruction provided gradients to correct small mistakes and give confidence to the viewpoint estimator.

3.4.5 Multiview supervision

We conducted experiments with varying numbers of views per model to assess the importance of multi-view supervision. We compared the performances of a network trained on 2, 5, or 10 views per model. For a fair comparison, we made sure that the training set size was constant throughout the different experiments: we truncated the 5 and 10 views sets in order to match the size of the 2 views for each model. This means that models trained on those sets will see more of each model, but fewer models in total. Similarly, the viewpoint labels will be concentrated on fewer models. Training was conducted with 10% of the labels in all cases.



(a) Regression only, 1% labels

(b) Regression only, 10% labels

Figure 3.4: Predicted vs ground truth azimuth for simple regression on test samples. Each point is colored with ground truth elevation.

The results in Table 3.3 show that multi-view supervision seems to be profitable for the network, as increasing the number of views leads to increased performances. One way to interpret this result is that more views allow the encoder to build a representation more representative of the global structure of the object, therefore making the reconstruction supervision more effective. Indeed, it will be easier for the network to learn global information about the object when presented with more views as the probability that the views cover the whole object increases while finding correspondences has to be performed across different models when the view count is low.

However, because the number of labeled models also decreases, there is a risk that not enough will be available to learn a correct viewpoint estimator, harming performances as seen with the chairs and cars. The viewpoint estimator falls in this case in a local minimum, as depicted in Section 3.4.4. We theorize then that multiple views benefit the framework, as long as it does not come to the detriment of variety in pose labels.

Views	aeroplane		car		chair	
	Acc (%, ↑)	Err (°, ↓)	Acc (%, ↑)	Err (°, ↓)	Acc (%, ↑)	Err (°, ↓)
2	56.9	25.1	40.7	39.1	30.1	44.7
5	**59.1**	**23.4**	48.0	32.6	**49.0**	**30.5**
10	34.4	45.2	**54.4**	**26.1**	26.7	49.2

Table 3.3: Viewpoint prediction performance for varying number of views, performed at 10% of the labels. Acc: accuracy at 30°, Err: median angular error.

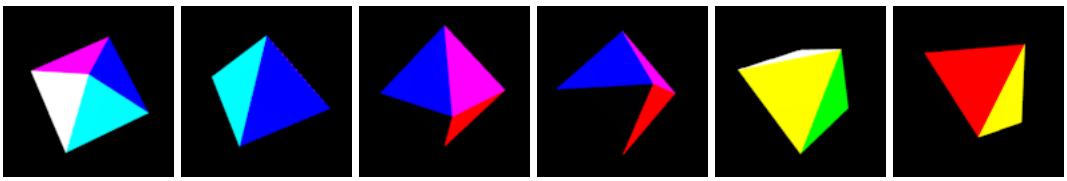
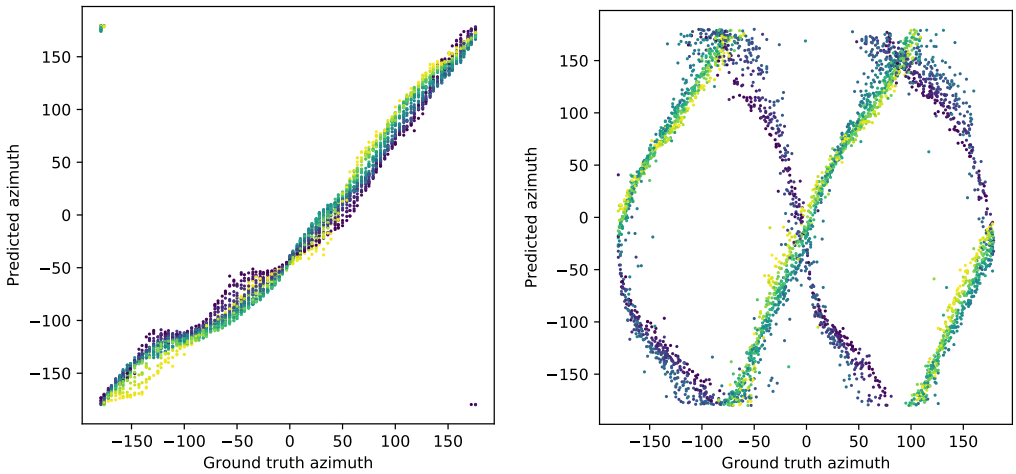


Figure 3.5: Example views from the toy dataset

3.4.6 Unsupervised viewpoint estimation

In these experiments, we assess the feasibility of training our framework in an unsupervised way, that is, without any pose labels, relying only on reconstruction. To this end, we designed a very simple dataset consisting of views from a single octahedron, with different colors on each face in order to break symmetries (Fig. 3.5). The results of the viewpoint prediction shown on Fig. 3.6a confirm that we can indeed learn the correct structure of the pose in easy cases. Having no reference point, this is learned up to a random rotation, which we recovered using the validation set by minimizing the distance between ground truth and prediction.

However, we found that our model was unable to learn the correct pose when confronted with more complex data, e.g. cars (Fig. 3.6b). We observe that the learned pose wraps twice around the pose space while the ground truth completes only one rotation. This is easily explainable as cars exhibit a strong symmetry when flipped 180° around the vertical



(a) Toy data (b) Cars

Figure 3.6: Predicted vs ground truth azimuth without pose labels. Each point is colored with ground truth elevation.

axis. Therefore, the viewpoint predictor identified these two poses to the same point. We also note that the above horizontal views are treated differently from the below ones. This is explained by the perceived way the object is rotating depending on whether the observer is located above or below the object.

Full predictions can be visualized Fig. 3.7. We see that the predictions for the octahedron are close to the ground truth, although they are squeezed - minimum and maximum elevation are about -1 and 1 radians respectively, although ground truth spans the whole $(-\pi/2, \pi/2)$ range.

Predictions for chairs however lack global structure. While close views are likely to be mapped together, illustrated by large clusters of predictions with similar color, there are abrupt changes, most notably around the 0° mark.

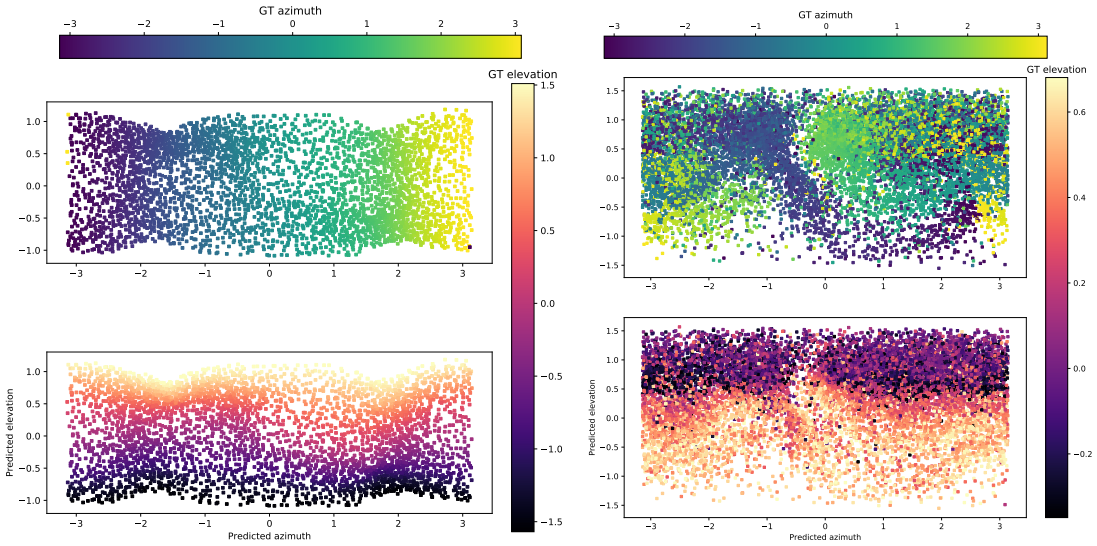


Figure 3.7: Visualization of predictions. Left: octahedron, Right: chairs

3.4.7 Novel view synthesis

We also demonstrate that our model is able to generate arbitrary views of an unseen object from a single image. To do so, we feed an image to the encoder to obtain an embedding defining the identity of the object we want to generate views from. Then, we rotate it at the desired viewpoint and decode it. Example results for all three categories are shown on Fig. 3.8, with viewpoints picked every 30 degrees in azimuth from the origin. We can observe that prominent features defining the identity of the object - e.g. global shape, texture - are preserved, and the viewpoints are correctly spaced. Of course, as with any other method doing view synthesis, errors occur as the model has to fill in the parts of the object that are self-occluded. This results in the loss of finer details, like spindles between chair legs. However, the correctness of the viewpoints means those pictures could be used to further refine predictions.

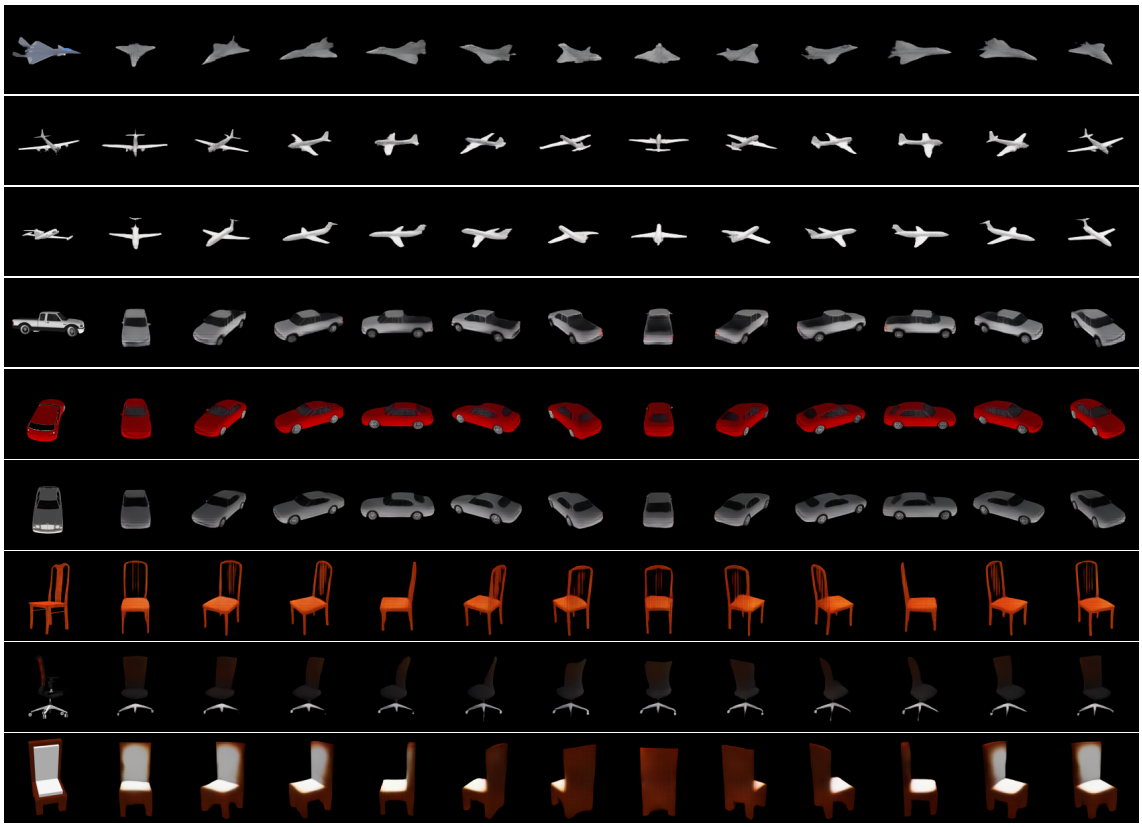


Figure 3.8: Novel view generation. Leftmost image provides object embedding

Layer	# channels	Kernel	Stride
Conv2D BatchNorm ReLU	32	3x3	1
Conv2D BatchNorm ReLU	32	3x3	2
Conv2D BatchNorm ReLU	64	3x3	1
Conv2D BatchNorm ReLU	64	3x3	2
Conv2D BatchNorm ReLU	128	3x3	1
Conv2D BatchNorm ReLU	128	3x3	2
Conv2D BatchNorm ReLU	256	3x3	1
Conv2D BatchNorm ReLU	256	3x3	2
Conv2D BatchNorm ReLU	512	3x3	1
Conv2D BatchNorm ReLU	512	3x3	2
Conv2D BatchNorm ReLU	variable	4x4	1

(a) Encoder architecture.

Layer	# channels	Kernel	Stride
ConvTranspose2D BatchNorm ReLU	512	4x4	1
ConvTranspose2D BatchNorm ReLU	512	3x3	2
ConvTranspose2D BatchNorm ReLU	256	3x3	1
ConvTranspose2D BatchNorm ReLU	256	3x3	2
ConvTranspose2D BatchNorm ReLU	128	3x3	1
ConvTranspose2D BatchNorm ReLU	128	3x3	2
ConvTranspose2D BatchNorm ReLU	64	3x3	1
ConvTranspose2D BatchNorm ReLU	64	3x3	2
ConvTranspose2D BatchNorm ReLU	32	3x3	1
ConvTranspose2D BatchNorm ReLU	32	3x3	2
ConvTranspose2D BatchNorm ReLU	32	3x3	1
ConvTranspose2D Sigmoid	3	3x3	1

(b) Generator architecture.

Table 3.4: Network architectures. Both f^v and f^e use the encoder architecture illustrated in Table 3.4a with output sizes 3 and 3*128=384 respectively, while f^d uses the generator architecture in Table 3.4b

3.5 Conclusion

The results obtained by this semi-supervised approach validate several important points:

Analysis by synthesis approaches are able to use unlabeled samples to help predict viewpoints. Compared with a strictly supervised baseline, the simple addition of extra images without annotations is sufficient to boost performances. This demonstrates that a reconstruction task provides a good target to guide viewpoint learning, provided that the viewpoint information is used in an interpretable way.

Designing a **pose-specific** approach, by making use of an interpretable prediction space is useful, allowing to surpass competitive generic semi-supervised approaches.

Completely **unsupervised pose estimation** seems possible, although experiments only show a functional system in an ideal unambiguous case. On more realistic data, the predicted poses lose meaning as the reconstruction network makes use of object symmetries to simplify its task. Nonetheless, this proves reconstruction alone can be sufficient as a supervision signal.

However its limitations are still too large to be applied to real scenarios, and even in toy cases, the exact pose learned only loosely follows the ground-truth labels. In particular, a relevant property that is not guaranteed in the proposed system is strict **geometric consistency** in the decoder, e.g. applying a 30° rotation along a specific axis to the representation exactly rotates the object by 30° along the same axis in the output image.

In the current architecture, the convolution layers that make up the decoder are not constrained enough to respect this and can learn arbitrary mappings from the embeddings to image space as long as the reconstruction objective is minimized, leading to the uninterpretable pose shown in Fig. 3.7. Closely related to this is the need to break symmetries, as the current approach can exploit them to minimize the reconstruction loss (Fig. 3.6b). These issues will be the main focus of the next chapters, in an attempt to design fully unsupervised pose estimation architectures.

Chapter 4

ViewNet: geometry guided unsupervised viewpoint estimation

4.1 Introduction

Although the method described in the previous chapter managed to learn an equivariant viewpoint space in a completely unsupervised manner on toy data, its failure on real objects makes it inadequate for real scenarios. In particular, the difficulty of modeling complex objects with possibly ambiguous views calls for careful design to avoid uninterpretable solutions. Two aspects of the problem can explain failure when no labels are available: object symmetries and the extra flexibility given by the convolution-based decoder.

Many of the objects we use in everyday life exhibit symmetric viewpoints. These can loosely be defined as a group of at least two viewpoints under which the object appear relatively similar. Some cases are obvious, for instance, glassware without marking has

The main findings of this chapter have been published in ICCV 2021 (Mariotti et al., 2021).

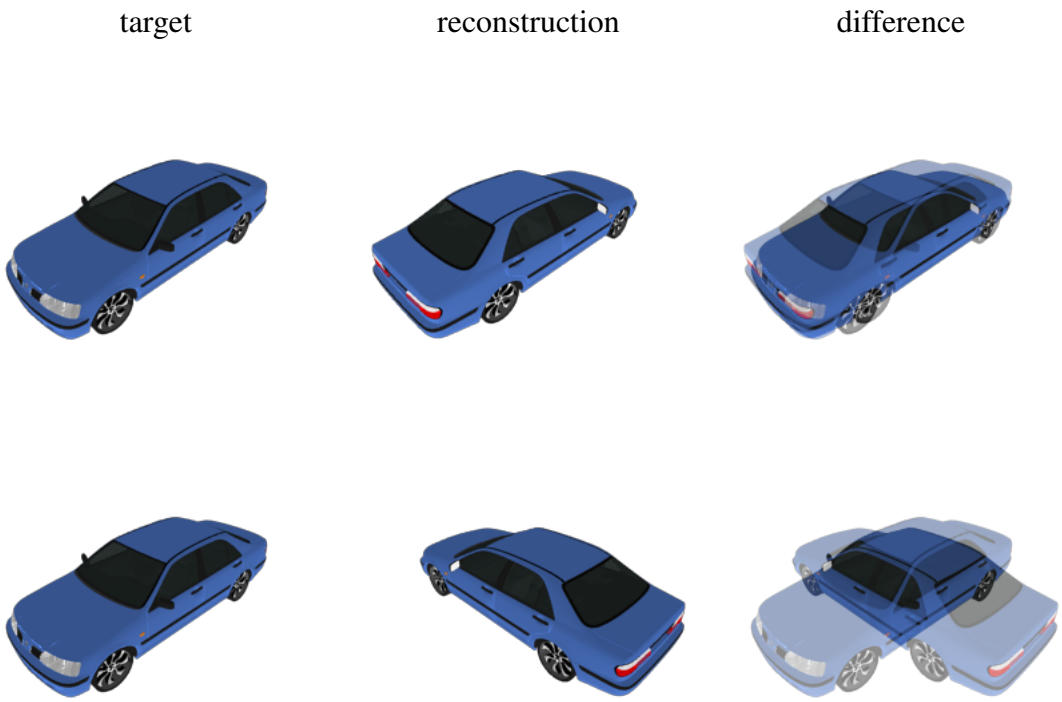


Figure 4.1: Illustration of the local minima caused by symmetries. In the first row, the predicted viewpoint is completely wrong, but the reconstruction error is relatively low as the two views align almost perfectly. In the second row, the prediction is closer to the ground truth, but the reconstruction error is much higher. This wrongly encourages models to represent the two views from the first row close together, while the second view of the second row will tend to be predicted away from the previous two

a perfect rotational symmetry and therefore appears exactly the same no matter which azimuth it is viewed from, while others can be more pervasive, as showcased in Fig. 4.1. Not only do these symmetries obviously hold the potential of causing mistakes in a pose estimation system, they are particularly problematic in a reconstruction-based approach. Indeed, mistaking a viewpoint for one of its symmetric views will only cause small reconstruction errors, making it a good strategy to optimize the objective. Additionally, even if the system manages to distinguish between symmetric viewpoints, they are likely to be mapped to the same neighborhood in pose space due to their similarities, as shown

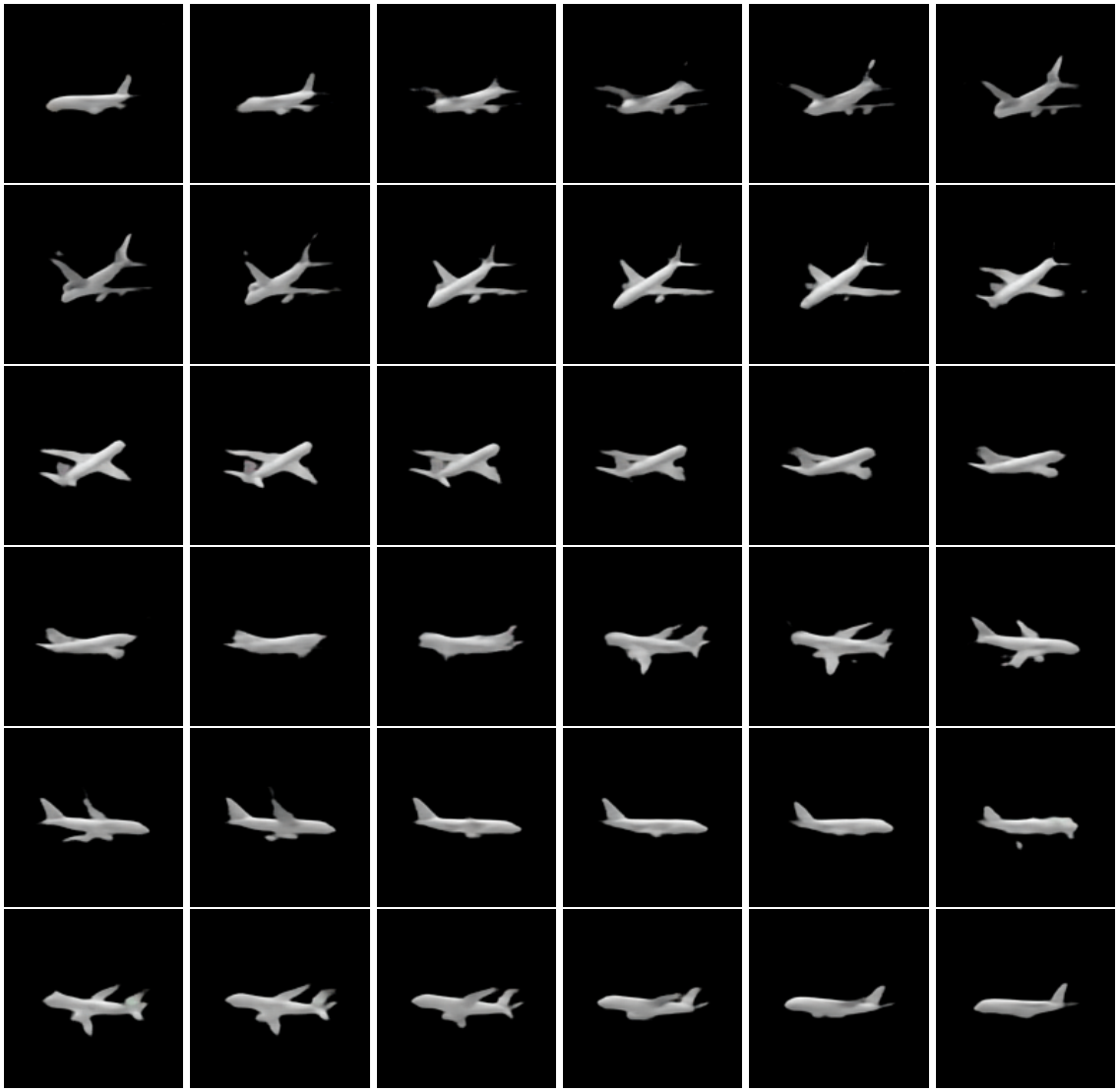


Figure 4.2: Reconstructions obtained by training the semi-supervised system described in Chapter 3 without labels, and querying the model with progressive azimuth. Each view is separated by 10°. While some transformations seem to roughly follow the object geometry, e.g. the third row, some non-geometrical transforms can happen in the span of a few degrees of rotation, like between the fifth and sixth frame of the fourth row

in Fig. 3.7. To prevent this later point, a good approach is to enforce strong geometric consistency during the decoding process.

When relying on a convolutional decoder to produce reconstructions, there is a possi-

bility that the pose mapping will converge to an uninterpretable solution, as illustrated in Fig. 4.2. This creates inescapable local minima that prevent convergence to a correct viewpoint embedding. The solution proposed in this chapter consists in removing the possibility for the model to learn such mapping by enforcing as much as possible a geometric consistency using a complete 3D reconstruction of the object. More precisely, instead of trying to learn an embedding of our object hoping it will be equivariant, equivariance is directly enforced by interpreting the representation as a voxel reconstruction of the object and rendering it according to the predicted viewpoint using completely analytical means. This approach is motivated by recent progress in 3D shape recovery with (Tulsiani et al., 2017; Yan et al., 2016) and without (Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018) supervision.

Therefore, the work presented in this chapter extends the previous approach and allows fully unsupervised viewpoint estimation on real objects, using image pairs as inputs as seen in Fig. 4.3 (Section 4.3). Evaluating on real data yield some surprising results, as unsupervised and even completely untrained methods can beat fully supervised approaches in some scenarios (Section 4.4). The reasons behind this phenomenon are twofold: a strong bias in the viewpoint labels of these categories, caused both by the careless data collection and noisy human labeling, and a label alignment procedure that allows for overfitting to this bias. Consequently, a bias-aware metric is proposed in order to mitigate the effects of unbalanced data (Section 4.4.4).

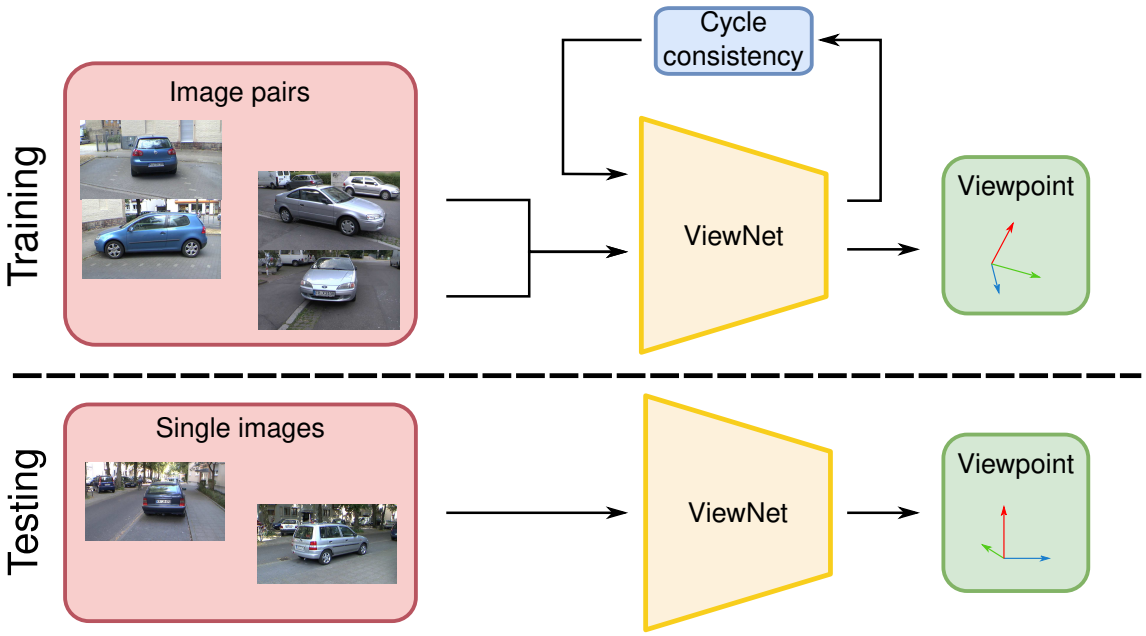


Figure 4.3: ViewNet learns to extract the camera viewpoint via self-supervised training on a collection of image pairs. During inference, it can estimate viewpoint from a single image.

4.2 Related work

Supervised pose estimation. Effective pose estimation from images has many real-world applications, e.g. in robotics or autonomous vehicles, and thus has been extensively studied. While early works performed pose estimation by matching low-level image descriptors (Hinterstoisser et al., 2011, 2012; Brachmann et al., 2014), more recent approaches employ deep networks for predicting 3D bounding boxes (Rad and Lepetit, 2017; Tekin et al., 2018; Grabner et al., 2018), or classifying viewpoints directly (Tulsiani and Malik, 2015; Kehl et al., 2017). Keypoint prediction is a closely related task, and multitask setups where both are jointly learned have been shown to be successful (Tulsiani and Malik, 2015; Zhou et al., 2018). Alternatively, one can be used to learn the other, as one can recover pose by aligning keypoints (Pavlakos et al., 2017), or discover them by enforcing a pose-aware sparse representation of objects (Suwajanakorn et al., 2018)

Recent work has proposed modeling the topology of the viewpoint space by quantifying the uncertainty with a von Misses distribution (Prokudin et al., 2018), learning 2D image embeddings that are equivariant to 3D pose (Esteves et al., 2019), employing a spherical exponential mapping at the regression output (Liao et al., 2019), or introducing cylindrical convolutions (Joung et al., 2020). However, all of these approaches are supervised and require pose annotations from datasets such as PASCAL3D+ (Xiang et al., 2014) or LINEMOD (Hinterstoisser et al., 2012). The first of which was manually annotated, and the second was created in a controlled lab setup where poses were collected with each image. Alternatively, coarse viewpoint estimation can be obtained without manual annotations using structure from motion algorithms on videos (Sedaghat and Brox, 2015; Novotny et al., 2017). Ground truth pose annotations are challenging to acquire, and recent benchmarks still require human intervention in order to set the coordinate system for each instance and to correct automatic pose errors (Ahmadyan et al., 2021).

3D-aware representations. A parallel line of work learns representations that are aware of the underlying 3D structure of objects from images. Earlier works employ auto-encoders to disentangle pose and object appearance, with (Worrall et al., 2017) or without (Kulkarni et al., 2015) pose supervision. More recent works extend this disentangled pose learning from in-plane rotations to full 3D poses by crafting models that reason with spherical representations (Cohen et al., 2018; Esteves et al., 2017), apply 3D rotations on embeddings to reconstruct images from a different viewpoint (Rhodin et al., 2018), use denoising auto-encoders to better extract viewpoint information (Sundermeyer et al., 2018), or by generalizing variational auto-encoders to spherical functions (Falorsi et al., 2018). First proposed for 2D feature maps, spatial transformers (Jaderberg et al., 2015) provide a way to apply in-plane transformations to any representation using spatial re-sampling and were later extended to 3D convolutions (Yan et al., 2016). These sampling operations can be used to represent complete 3d scenes from multiple views (Sitzmann et al., 2019a). In a related analysis by synthesis approach, (Chen et al., 2020) also learn pose representations via an appearance-based reconstruction loss. At inference time, they

iteratively optimize for the viewpoint that minimizes the appearance loss between the synthesized view and the input image. However, apart from a few unrealistically simplified experiments, all of these methods require 3D annotations in order to learn meaningful embeddings. Unlike in-plane rotations, which are simple enough to learn in an unsupervised way, 3D rotations can cause drastic appearance changes that are often too complex for networks to learn without pose annotations (Mariotti and Bilen, 2020).

Viewpoint-conditioned generation. An increasingly popular way of learning interpretable representations is by using a generation process conditioned on the relevant information. The two main ways of building such representation rely either on encode-decoder approaches, using image pairs where semantics are shared (Whitney et al., 2016; Jakab et al., 2018; Mariotti and Bilen, 2020), or on adversarial models to generate new samples in a controllable way (Chen et al., 2016; Nguyen-Phuoc et al., 2019). Both techniques have been shown to estimate viewpoints without labels (Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018; Mustikovela et al., 2020). Encode-decoder approaches are closely related to the field of unsupervised 3D reconstruction (Henzler et al., 2019; Niemeyer et al., 2020; Oechsle et al., 2019; Olszewski et al., 2019). In contrast to (Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018) that do both 3D reconstruction and pose estimation, we propose a simpler fully self-supervised approach that is able to leverage appearance matching as supervision, allowing for novel view synthesis that can be used to further refine predictions.

SSV (Mustikovela et al., 2020) uses an adversarial model to generate objects with random rotations while learning to regress viewpoint at the same time. In contrast, our proposed method ensures geometric consistency during the image generation process, allowing for more robust viewpoint estimation. Furthermore, GAN training can be unstable (Metz et al., 2016; Arjovsky and Bottou, 2017), an issue often reflected in the auxiliary objectives required to guide training. In contrast, our method operates via image reconstruction alone, and can easily generate images from novel viewpoints. Several non-adversarial

generative approaches have also been proposed (Mariotti and Bilen, 2020; Chen et al., 2020) that reconstruct specific object instances in order to leverage pixel-level supervision. However, unlike our approach, these methods require at least a partially labeled training set.

4.3 Method

Given a collection of unlabeled images $\mathcal{T}$, at training time we aim to learn a function $f_v : I \rightarrow \mathcal{V}$ that can map from image space I to pose space $\mathcal{V}$. At test time, we can then apply this function to a single image I, containing an object of interest, in order to estimate its 3D viewpoint v relative to the camera. 3D viewpoints can be represented in different ways, including the Euler angles (azimuth, elevation and tilt), or with a rotation matrix $R \in SO_3$, and we use both representations interchangeably.

As ground-truth viewpoints of the objects in $\mathcal{T}$ are challenging to acquire, we formulate our problem as a self-supervised task that uses principles from conditional generation and synthesis by analysis. To this end, we propose to factorize the viewpoint and appearance of objects via two functions f_v and f_a. Given an image I, f_a outputs an appearance feature $\mathbf{a}$ for the object contained in it.The decoder f_d, can reconstruct the image I given the pose of the object $\mathbf{v}$ and its appearance $\mathbf{a}$. f_v, f_a, and f_d are instantiated as neural networks parameterized by θ^v, θ^a and θ^d respectively. Clearly, such a factorization is not guaranteed without some constraints on f_v and f_a. To overcome this ambiguity, we use image pairs of rigid objects at training time that differ by their viewpoint. Such pairs can be extracted from video sequences, generated by perturbing still images or rendered from 3D CAD models. Hence we assume that the set of unlabeled images $\mathcal{T}$ can be described as N image pairs $\mathcal{T} = \{(I_i, I_i')\}_{i=1}^N$ where each pair contains images of the same object instance from two different viewpoints (v_i, v_i'), where the actual viewpoint information, relative or absolute, is not available. Given an image pair (I_i, I_i'), we propose to extract pose features $\mathbf{v}$ from I_i and appearance features $\mathbf{a}'$ from I_i', and use them to reconstruct I_i.

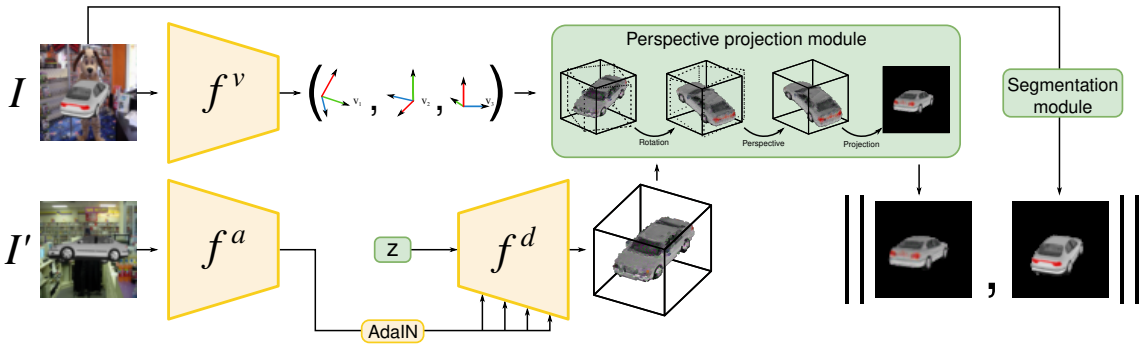


Figure 4.4: Overview of ViewNet. f_v is the viewpoint prediction network. At training time, f_a encodes the object appearance embedding from image I' which is decoded by f_d into a 3D representation and transformed by the estimated viewpoint into an image in the same pose as I using the projection module. This reconstruction, which can be segmented, is then compared to I to guide training. Yellow blocks indicate learned parameters, while green ones are fixed or analytical modules.

An overview of our model is shown in Fig. 4.4. Our learning task consists of solving the following objective:

$$\min_{\theta^v,\theta^a,\theta^d} \sum_{(I,I')\in\mathcal{T}} ||f_d(f_a(I'),f_v(I)) - I||. \tag{4.1}$$

4.3.1 Pose estimation network f_v

Similarly to Chapter 3, we design the pose estimation network to output a point on the 3D unit sphere (i.e. $f_v(I) = \mathbf{v} \in S^2$) by setting the output dimension to 3 and subsequently normalizing the output, and uniquely map each point on the sphere to a viewpoint. To this end, we apply an orthogonalization operation to f_v's output with the following steps. First, we define an arbitrary vector $\mathbf{u} \in S^2$ that represents the upwards direction, then we apply two successive cross products, $\mathbf{w} = \mathbf{v} \times \mathbf{u}$ and $\mathbf{u}' = \mathbf{w} \times \mathbf{v}$, and normalize the results to obtain orthogonal vectors. Finally we define the rotation matrix R as $[\mathbf{v}, \mathbf{w}, \mathbf{u}']$. This matrix is then used to rotate the object representation during the generative stage, described later. This approach uses an arbitrarily chosen upwards direction, meaning we assume images do not contain in-plane rotations. However, in the more general case, $\mathbf{u}$

can be learned jointly with $\mathbf{v}$, effectively describing the full range of 3D rotations, which in the general case is an efficient parameterization of poses for optimization purposes (Chen et al., 2022).

The main pitfall of unsupervised viewpoint estimation is the collapse of predictions caused by symmetries. Current approaches work well on simple objects e.g. a cube with each face colored differently. However, real-world objects tend to have at least one, if not multiple, symmetric viewpoint pairs. We say that two viewpoints v, v' form a symmetric pair, $v \sim v'$, if the image produced by observing the object from v is close to that produced from v'. For instance, in most cars, $(a, e, t) \sim (a + \pi, e, t)$ forms a symmetric pair for any azimuth a, elevation e, and camera tilt t. As a result of this, unsupervised methods based on reconstruction often equate those two viewpoints, leading to a collapse of the predictions. Different workarounds have been proposed to mitigate this, such as using adversarial losses to enforce a prior on the pose distribution (Tulsiani et al., 2018), using multiple prediction heads (Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018), or enforcing some symmetric consistency in the predictions using a flipped version of the image (Mustikovela et al., 2020). The main drawback of this last approach is that it is only valid for a left-right planar symmetry, and would likely fail in the aforementioned car example. To overcome this problem, we use multiple prediction heads for our pose estimator, resulting in multiple hypotheses for $\mathbf{v}$. Each head can learn to specialize in a subset of the viewpoints, and in the case of a symmetric pair $v \sim v'$, both can simultaneously be predicted by two different heads.

In practice, each head of the predictor f_v outputs a viewpoint prediction, and the one associated with the lowest reconstruction error is chosen as the prediction at training time:

$$\mathbf{v}^* = f_v(I)_{m^*} \quad \text{s.t.} \quad m^* = \arg\min_{m \in M} ||f_d(f_a(I'), f_v(I)_m) - I||, \tag{4.2}$$

where $f_v(I)_m$ denotes the prediction of the m-th head of the viewpoint predictor and M is the number of heads. Gradients will only be propagated through m^*, ensuring that symmetric pairs get separated. It might seem desirable to encourage diversity in the predic-

tions to prevent all heads from collapsing to the same output, however, we experimentally show that simply having multiple heads is sufficient.

At test time, ViewNet only requires the pose prediction network f_v, and does not need f_a or f_d in order to make a prediction. To achieve this, we jointly train a selection head, which is tasked with picking the best prediction for each input image given the range of options. This extra head is tasked to minimize the cross-entropy between the selection prediction and a one-hot distribution representing m^*, computed via Eq. (4.2). Formally, f_v outputs M viewpoint predictions and M logits that are trained to predict m^*. Although m^* is not guaranteed to be the prediction closest to ground-truth pose, we observe it is enough to differentiate between symmetric viewpoint pairs. Compared with Insafutdinov and Dosovitskiy (2018), this allows us to efficiently maintain multiple hypotheses at test time, which translates to more robust predictions, and we do not require complex solutions such as reinforcement learning as in Tulsiani et al. (2018).

4.3.2 Appearance encoding network f_a

The appearance $f_a(I') = \mathbf{a}' \in \mathbb{R}^n$ of the object represented in the input image is also learned with a convolutional network. In a standard encoder-decoder architecture, $\mathbf{a}'$ would be used as an input to f_d to produce a reconstruction. However, this offers no guarantee that the viewpoint $\mathbf{v}'$ and appearance $\mathbf{a}'$ embeddings are correctly factorized. In particular, information about $\mathbf{v}'$ could be encoded in $\mathbf{a}'$. This means that a change in $\mathbf{v}'$ could induce changes in the appearance of the reconstruction. In extreme cases, the network could even ignore $\mathbf{v}$ and reconstruct I by memorizing the (I, I') pairs. To mitigate this, we use an object-conditional generation process which makes use of adaptive instance normalization (AdaIN) (Huang and Belongie, 2017). Initially developed for style transfer, this approach is popular in GANs (Brock et al., 2019; Nguyen-Phuoc et al., 2019; Mustikovela et al., 2020) due to its ability to adapt the generation process at different scales.

Formally, AdaIN works similarly to regular instance normalization that performs channel-wise normalization of its input $\mathbf{x}$, but uses affine scaling parameters γ and β that are fed by an external process - in our case, the prediction of f_a - in order to alter the meaning of a feature map while preserving its information.

$$\text{AdaIN}(\mathbf{x}, \mathbf{a}) = \gamma_{\mathbf{a}} \left(\frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})} \right) + \beta_{\mathbf{a}} \tag{4.3}$$

where $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ are computed only over spatial dimensions.

Our generation pipeline works by refining a random static code $\mathbf{z} \in \mathbb{R}^m$ through the decoder network to the final rendering stage. $\mathbf{z}$ is randomly picked from a normal distribution at the beginning of the training process and remains constant. Its purpose is to encode the average object in a canonical pose. The appearance of the object is gradually encoded by AdaIN layers (see Fig. 4.4), which apply an affine transformation to the features parameterized by $\mathbf{a}$. As they are applied uniformly over each feature channel, it is impossible for them to alter local information of the features. Additionally, a standard encoder-decoder architecture would only use $\mathbf{a}$ as input of the decoder, meaning fine details of the object can be lost during the complex decoding process. By comparison, applying transformations across different layers means they can influence all levels of the reconstruction, resulting in more faithful reconstructions.

Additionally, one could think of guiding the optimization process by adding a consistency loss enforcing $f_a(I)$ and $f_a(I')$ to be close, since both contain the same object. However, we found that the reconstruction objective is sufficient to properly learn f_a and therefore did not include such an objective for the sake of simplicity.

4.3.3 Decoder network f_d

In order to ensure an accurate viewpoint prediction, we aim to strictly enforce geometric consistency during the generation process. To this end, f_d is modeled using 3D convolutional layers, and uses a 3D spatial transformer with perspective for image rendering,

similar to those used in Yan et al. (2016), and is combined with a pseudo-ray tracing operation inspired by (Tulsiani et al., 2017). Placing the rotation at the final stage of the network, as close as possible to the reconstruction loss, ensures that gradients are efficiently propagated to f_v. The absence of parametric transformations between f_v and the target image guarantees that viewpoint errors cannot be compensated for by convolutional layers, as can happen in GAN-based models.

Our rendering module consists of three main steps and is related to those used in Yan et al. (2016); Tulsiani et al. (2017, 2018); Insafutdinov and Dosovitskiy (2018), however, our pipeline also makes use of texture information. Specifically, the steps involve: (i) Rotation. Given a 3D volume V, a spatial transformer can be used to rotate it using a rotation matrix R. The new volume is obtained by resampling the data along the rotated axis. (ii) Perspective. Similarly, perspective can be simulated with a spatial transformer. The single point perspective of a pinhole camera will have the effect of decreasing the apparent size of objects proportional to distance. We can therefore resample the volume by dilating close points and contracting distant ones. (iii) Projection-based ray-tracing. Finally, the volume is projected to a two-dimensional image plane. As parts of the objects will be subject to self-occlusion, we use a pseudo ray marching operation to compute which voxels will appear in the output image, ensuring geometric consistency.

For each entry in the 3D volume V, the first three channels C represent the RGB channels of an image, while the fourth one Q is an occupancy map, containing information about the shape of the object. The value of each cell is interpreted as the probability of the object occupying the corresponding spatial location. To compute the projection, we have to estimate where each light ray is likely to stop. Since we already accounted for the perspective, all our rays are parallel, leaving only the depth of each stopping point to be computed. Compared with Tulsiani et al. (2017), we do not have to compute a path for each light ray i.e. it is embedded in the shape of the tensor. This means we can compute all lights paths simultaneously using efficient parallel operations, in a manner similar to the orthographic projection used in Gadelha et al. (2017). The probability of the light ray,

at pixel coordinates i, j, stopping at depth k is given by

$$Q'_{i,j,k} = Q_{i,j,k} \times \prod_{l=0}^{k-1} (1 - Q_{i,j,l}), \tag{4.4}$$

with the convention that an empty product is equal to 1. The first term represents the probability of the voxel at coordinates (i, j, k) being occupied, and the second one is the probability of all the previous ones being not visible. Hence, the final pixel value at coordinate i, j is

$$\hat{I}_{i,j} = \sum_{k=1}^{n} \left[C_{i,j,k} \times Q_{i,j,k} \times \prod_{l=0}^{k-1} (1 - Q_{i,j,l}) \right]. \tag{4.5}$$

This is similar to the formulation in Tulsiani et al. (2017, 2018), although in our case, the ray-tracing is parallelized and used to sample RGB values, rather than computing depth or ray termination.

A failure case of our approach consists of ViewNet using the volume V as a canvas and "painting" the object in different poses on the sides, illustrated by the failed results in Fig. 4.6b. More generally, this results in errors in the predicted shape of the object, since we do not know which pixels belong to it. To address this, instead of trying to directly estimate occupancy Q, we learn Q' such that $Q = S + Q'$ where S is a three-dimensional Gaussian distribution centered on V. Q' can be interpreted as a residual that deforms a shape prior S so that it matches the shape of the observed object. S encodes a prior for the shape and position of the object, following the assumption that the object is at the center of the volume, while discouraging the network from using voxels that are far away from said center.

4.3.4 Cycle consistency supervision

Using appearance supervision, as opposed to only object silhouettes as in Yan et al. (2016); Tulsiani et al. (2018); Insafutdinov and Dosovitskiy (2018), enables ViewNet to also represent appearance information. This has two key advantages. First, our method can generate images of objects from novel views. Second, we can use these novel views to

regularize our model during training by enforcing consistency between a generated image and its known viewpoint.

Given a randomly sampled viewpoint $\tilde{\mathbf{v}} \sim \mathcal{U}(\mathcal{V})$, we can render a novel image $\widetilde{I} = f_d(\tilde{\mathbf{v}}, \mathbf{a}')$ using appearance information in $\mathbf{a}'$ extracted from image I'. By feeding this to f_v, we can compute the distance between the sampled viewpoint $\tilde{v}$ and its estimated viewpoint $f_v(\widetilde{I})$, i.e. $\mathcal{L}_{cycle} = ||f_v(\widetilde{I}) - \tilde{\mathbf{v}}||$ and backpropagate this error to the viewpoint estimator. Assuming the reconstructions are of sufficient quality, this allows us to generalize beyond the potentially limited set of poses that are present in the training set, and these newly generated samples help regularize the viewpoint estimation network. This is the main technique used to estimate viewpoints in Mustikovela et al. (2020), although it is used here in a reconstruction-based rather than generative setting.

4.4 Experiments

Here we present 3D pose estimation results on both synthetic and real image datasets.

4.4.1 Implementation details

ViewNet consists of three sub-networks: f^v, f^a, and f^d. Both f^v and f^a contain seven convolutional layers interleaved with batch normalization and ReLU activation functions respectively. f^v takes a 64×64 RGB image I as input and outputs $M = 3$ viewpoint hypotheses. f^a encodes a second RGB image I', depicting the same object instance captured from another viewpoint, and outputs a 256 dimensional appearance vector. The input to f^d is a 1024 dimensional fixed canonical code vector. The canonical code is passed through seven 3D transposed convolutions, each followed by a ReLU, and the feature maps are further conditioned on the output of f^a via adaptive instance normalization (AdaIN) layers. Detailed network architectures are presented in Table 4.6 at the end of this section.

The resulting 3D feature map is projected to an image based on the predicted pose and used to compute the reconstruction error w.r.t. I. We use a perceptual loss (Johnson et al., 2016), as it provides more informative gradients compared with standard pixel-level reconstruction losses.

In all experiments, we set the minibatch size to 64 and use the Adam optimizer (Kingma and Ba, 2014) and select the model with the best performances on a held out validation set, stopping the training if no improvement is observed for 30 epochs. For each experiment, we train a separate model per category, replicating the framework of the approaches we compare with. In theory, it could be possible to use a single model and use a different F for each category, however, this would make the optimization much harder. Furthermore, it violates the assumption that a natural reference frame exists for all object instances, as it is not obvious to align a car with a table based on their respective semantics.

As our method is unsupervised, all viewpoints are predicted up to a random rotation. In order to evaluate our model, we must align its predictions with the ground truth. The standard alignment technique, performed by Tulsiani et al. (2018); Insafutdinov and Dosovitskiy (2018), involves computing the rotation that best aligns the predicted viewpoints with the ground truth. This is obtained from a small batch of validation images, using the orthogonal Procrustes algorithm. An alternative alignment procedure, used in SSV (Mustikovela et al., 2020), learns the parameters of a more flexible affine transformation that best maps the predicted viewpoints to the ground truth. This can shrink and/or expand the predicted viewpoint estimation compared to applying a single 3D rotation to translate them. We discuss potential issues with this approach in Section 4.4.3. We report performances in standard viewpoint estimation measures: accuracy at 30° and median angular error.

	Accuracy (%, ↑)			Median error (°, ↓)		
	airplane	car	chair	airplane	car	chair
MVC	69	87	81	14.3	5.2	7.8
Pointclouds	75	86	86	8.2	**5.0**	8.1
ViewNet	82	89	89	8.6	6.7	7.3
ViewNet + cycle	**86**	**91**	**92**	**7.7**	6.7	**7.0**

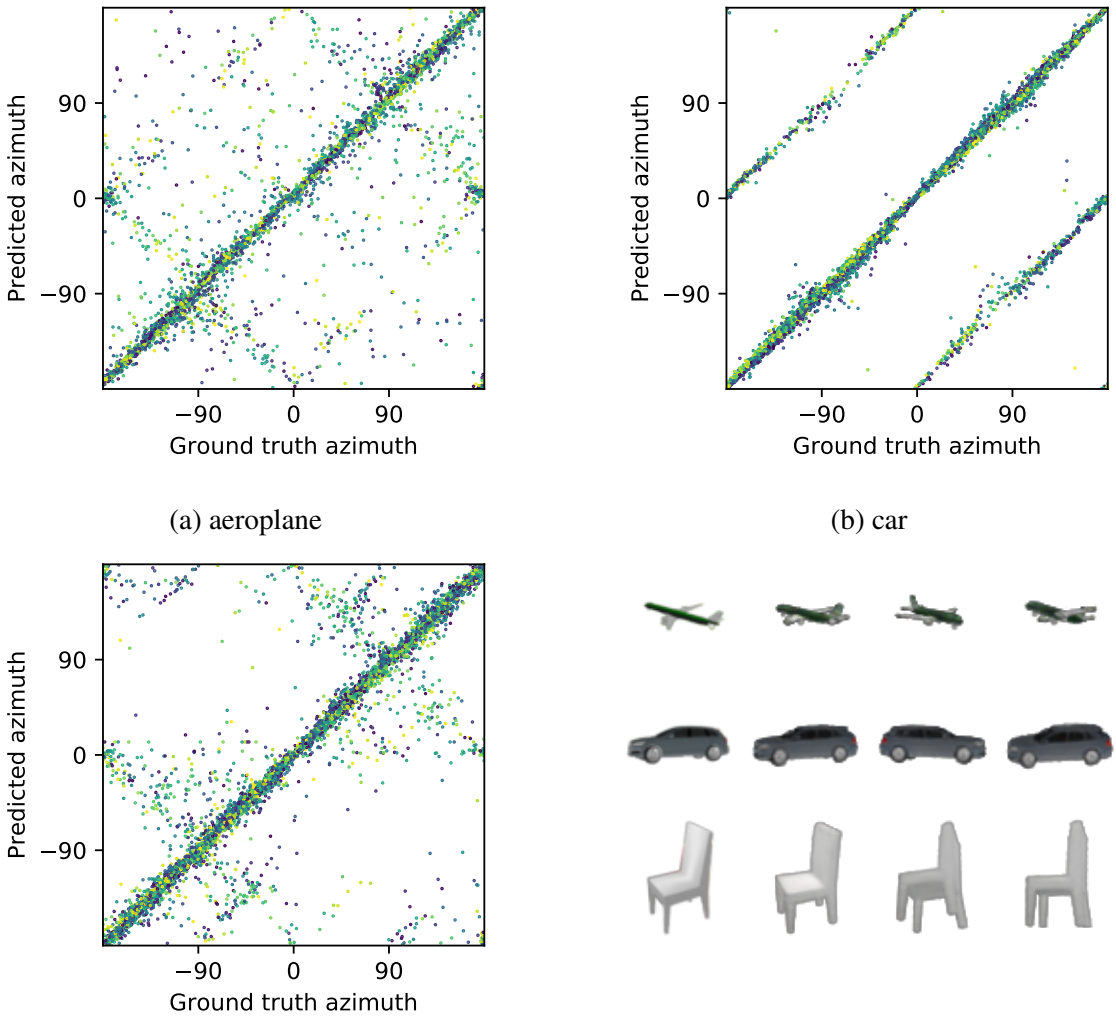
Table 4.1: ShapeNet results for unsupervised methods. Bold entries are the best-performing models for each category. Acc: accuracy at 30°, Err: median angular error.

4.4.2 ShapeNet results

Following Tulsiani et al. (2018); Insafutdinov and Dosovitskiy (2018), we evaluate ViewNet on the ShapeNet dataset (Chang et al., 2015), which contain 7.5k, 6.8k, and 4k 3D CAD models for cars, chairs, and planes respectively. Training, validation, and testing sets are created by splitting CAD models into (0.7, 0.1, 0.2) fractions respectively. To render image pairs, we randomly select viewpoints and light sources uniformly over the $[0^o, 360^o)$ azimuth range and $[-20^o, 40^o]$ elevation range. We report results for the standard setting where each CAD model is rendered from five random viewpoints at train and test time.

The results in Table 4.1 show that ViewNet outperforms existing unsupervised approaches, except for median error on cars. ViewNet learns to reconstruct textures in addition to shape, and this supervision is more informative compared to only binary masks, as we can leverage texture cues to efficiently disentangle symmetric viewpoints. For example, red tail lights on a car can indicate the rear. This penalizes a model that would reconstruct white headlights in their place. We investigate this further in our ablation study.

We observe that viewpoint cycle consistency provides a further boost in accuracy ('ViewNet + cycle'). Here, novel views are rendered at the same time as the regular reconstruction objective and then fed back to the viewpoint estimator. This indicates that our model can generate both novel and accurate images for a given viewpoint and learns to refine its



(a) aeroplane

(b) car

(c) chair

(d) Reconstruction proposals

Figure 4.5: (a) - (c) Comparison of ground truth versus predicted azimuth on three ShapeNet categories. A perfect predictor would appear as a single diagonal line. (d) Candidate reconstructions for each of the three heads. The left image is the input of the pose estimator, and the three following images are the renderings for each head, ranked by increasing mean-squared reconstruction error.

output in a self-learning manner.

We analyze the viewpoint predictions of ViewNet in Fig. 4.5, and plot the predicted az-

imuth against the ground truth. The strong diagonal indicates accurate predictions, while off-diagonal points are errors. The results reveal that the majority of the errors are caused by symmetries. For example, the car category shows a second line of predictions shifted by 180°. This corresponds to the $(a,e,t) \sim (a+\pi,e,t)$ symmetry mentioned in Section 4.3.1. Other categories showcase different symmetry-induced issues, with planes and chairs having a retrograde symmetry $(a,e,t) \sim (\pi-a,e,t)$. Samples for each proposed viewpoint are shown in Fig. 4.5 (d). One can see that the global input shape is matched relatively well across the different predicted views. Interestingly, reconstruction error is not necessarily directly correlated with viewpoint error, as the second proposed viewpoint for the car has a lower reconstruction error than the third, despite being rendered from a completely different viewpoint.

4.4.2.1 Ablation study.

Quantitative In Table 4.2, we study the effect of each proposed component in our pipeline. First, we reduce the number of heads in the viewpoint estimators to one and observe a large overall drop in the viewpoint accuracy. For the car category, the single-head estimator cannot deal with the front/back symmetries, resulting in a large performance loss. Second, we modify ViewNet to reconstruct a binary segmentation mask similar to Tulsiani et al. (2018) and Insafutdinov and Dosovitskiy (2018) instead of the pixel values. Using segmentation masks as targets achieves results comparable with previous segmentation-based approaches in Table 4.1. This indicates that ViewNet can leverage texture information to achieve better predictions. Third, we remove our Gaussian shape prior and directly estimate the occupancy grid Q, instead of Q', and observe that this does not have any significant effect on planes and cars, but causes a dramatic drop for chairs as the network tries to 'paint' the object on the faces of the volume. Next, we evaluate the conditioning strategy by removing the AdaIN layers and feeding the output of f^a in the first layer of f^d, similar to a traditional encoder-decoder pair. While this does not cause drastic performance issues, the reconstructions are less accurate, limiting the abil-

ity of this model to use them for self-training. Finally, we replace the analytic renderer with a learnable decoder using the deconvolutional architecture from Nguyen-Phuoc et al. (2019). In addition to causing the largest performance drop of all ablations, reconstructions from this model do not exhibit geometric consistency as the generated views do not smoothly change as the object rotates. Qualitative samples are shown in Fig. 4.6.

Qualitative In addition to the quantitative ablation study, we provide here a supporting qualitative analysis. In particular, we illustrate image reconstructions for ViewNet as well as three different ablation experiments on the ShapeNet dataset:

1. No Shape Prior: removing the Gaussian shape prior from the decode, that is, trying to learn Q directly.
2. Encoder-Decoder: Using a regular encoder-decoder architecture instead of decoding a canonical code adapted with adaptive instance normalization.
3. HoloGAN-Decoder: Using extra convolution layers after the rotation and projection, as per HoloGAN and SSV.

Reconstructions for different variants of ViewNet are shown in Fig. 4.6. These were obtained by feeding the leftmost image to the appearance network, then sampling the viewpoint space at regular azimuth and decoding the representation along those viewpoints. We note the difference in canonical viewpoints adopted by each model, as the first image in each series corresponds to different viewpoints. Interestingly, all models have learned to remove the piece of ground that appears under the car in the appearance image (visible as a plane that can be seen when zooming in), most likely because this is an uncommon feature in the dataset.

The black spots that appear around the object (Fig. 4.6b) indicate that the model fails to learn the proper shape of the object and considers the background as part of the object. An extreme case of this is when the model does not learn shape at all and tries to "paint" the object on the volume, such as chairs in Fig. 4.6b. This emphasizes the importance of

	Accuracy (%, ↑)			Median error (°, ↓)		
	airplane	car	chair	airplane	car	chair
ViewNet	**82**	**89**	**89**	**8.6**	6.7	**7.3**
Single-head	72	51	66	18.1	27.1	16.3
Segmentation Target	71	85	88	12.9	8.0	8.1
No Shape Prior	78	**89**	73	9.6	**6.6**	31.3
Encoder-Decoder	**82**	**89**	88	8.7	6.8	7.8
HoloGAN-Decoder	66	52	72	19.6	27.6	14.5
Constant	20	22	19	61.7	65.2	58.1

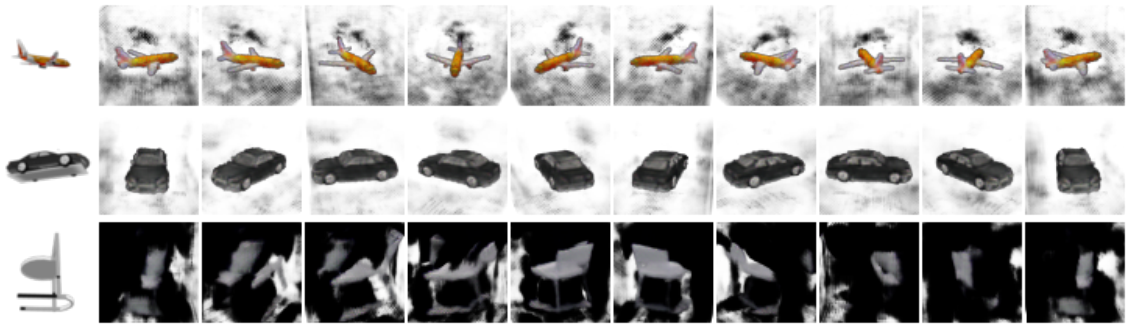
Table 4.2: Ablation study results. Here we compare different variants of ViewNet on ShapeNet. Acc: accuracy at 30°, Err: median angular error.

our shape prior.

Although their performance is similar, the views generated using an Encoder-Decoder architecture in Fig. 4.6c are not as faithful to the original object compared to using a generator with adaptive instance normalization. More precisely, the objects tend to be closer to the average object in the category, with the aeroplane being grey, as most aircraft tend to be, and the chair seat being square instead of round. The lack of fidelity in the reconstructions is apparent when comparing to ViewNet in Fig. 4.6a.

For the HoloGAN-style generation (Fig. 4.6d), there is no distinction between object and background as additional layers are used after the projection stage, translating in a black background for all generated images. We note that the geometry is not preserved, and we see poor consistency when traversing the viewpoint space, which is most apparent in the case of the airplane.

(a) ViewNet



(b) No Shape Prior

(c) Encoder-Decoder

(d) HoloGAN-Decoder

Figure 4.6: Generated views at constant elevation for ablated models. The leftmost image provides the object appearance.

		airplane	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv
Accuracy (%, ↑)	Constant	45	23	28	96	79	29	58	48	32	81	95	**89**
Unsupervised	VGG view*	64	63	25	96	78	56	76	48	46	86	**96**	85
	SSV*	–	–	–	–	**82**	67	–	–	–	–	**96**	–
	ViewNet*	**72**	79	29	96	75	84	**86**	52	72	87	**96**	**89**
	ViewNet	**72**	**81**	38	**97**	75	87	82	54	75	86	85	86
	ViewNet + cycle *	67	70	23	96	77	87	83	50	74	**89**	95	87
	ViewNet + cycle	71	80	**47**	96	80	**88**	83	**57**	**78**	88	88	82
Sup.	L. et al.	**88**	**88**	61	**96**	**97**	93	**93**	**74**	**93**	**98**	84	**95**
	G. et al.	83	82	**64**	95	**97**	**94**	80	71	88	87	**93**	86
Median error (°, ↓)	Constant	32.6	56.6	61.6	8.2	16.7	55.0	25.2	31.9	53.9	13.6	8.8	14.0
Unsupervised	VGG view*	20.8	22.2	55.8	7.9	9.7	25.8	14.4	29.6	33.0	10.0	8.6	**11.3**
	SSV*	–	–	–	–	**9.0**	10.1	–	–	–	–	**5.3**	–
	ViewNet*	15.0	16.0	54.1	8.1	16.1	12.9	12.3	27.0	16.9	10.1	9.1	14.4
	ViewNet	**14.0**	13.4	38.4	**7.2**	16.2	5.9	**10.1**	**24.4**	14.2	**9.2**	7.4	13.8
	ViewNet + cycle *	18.2	17.1	61.3	8.0	16.3	6.7	11.7	28.8	14.7	**9.2**	9.3	14.0
	ViewNet + cycle	14.4	**12.2**	**20.6**	**7.2**	14.9	**5.6**	11.5	25.0	**11.6**	11.5	15.6	15.8
Sup.	L. et al.	**9.2**	**11.6**	20.6	**7.3**	3.4	**4.8**	**8.2**	**8.5**	**12.1**	**8.7**	**6.1**	**10.1**
	G. et al.	10.0	15.6	**19.1**	8.6	**3.3**	5.1	13.7	11.8	12.2	13.5	6.8	11.0

Table 4.3: PASCAL3D+ results. Bold entries indicate the best-performing models in each category. Entries followed by a star (*) use a linear regression alignment procedure, and those without use a single global rigid alignment. Acc: accuracy at 30°, Err: median angular error.

4.4.3 PASCAL3D+ results

Next, we evaluate ViewNet on the challenging real-world PASCAL3D+ (Xiang et al., 2014) dataset. It contains real images from the PASCAL VOC and ImageNet datasets along with annotated viewpoints, including azimuth and elevation. As this dataset does not provide image pairs that contain the same object instance with varying viewpoints, we evaluate our models with 10 views per CAD model trained on ShapeNet. As PASCAL3D+ images have backgrounds, we synthetically add random background images

from SUN397 (Xiao et al., 2010) to our ShapeNet rendered views during training. These backgrounds are only added to the input training pairs to make ViewNet robust to backgrounds at test time. However, ViewNet is trained to reconstruct only the object, as it would require additional logic to reconstruct the background.

We report our results in Table 4.3. We observe that for some categories e.g. bottle, bus, sofa, train, and tv monitors, the ranges of viewpoints it contains are extremely restricted and concentrated around specific viewpoints. We reason that the viewpoint alignment procedure used for unsupervised methods is very effective in reaching strong performances on these classes. To test this hypothesis, we build a simple viewpoint predictor, a constant predictor, that outputs the average viewpoint from the validation set for each object category. This mimics the behavior of an untrained viewpoint estimator that has not learned anything useful and gets calibrated on validation data. We see that this method performs surprisingly well and even outperforms Grabner et al. (2018), a supervised approach on some categories. Even on non-trivial categories, the constant predictor performs surprisingly well, for instance, it obtains 43% accuracy on airplanes and 58% on chairs. By comparison, the same predictor on ShapeNet achieves a much lower performance (see Table 4.2), as the dataset was specifically crafted not to be biased.

To mitigate biases in the evaluation set, we propose a different evaluation strategy that consists of splitting the viewpoint space into discrete bins and then averaging performance over each bin. Doing so prevents biased predictors from reaching near-perfect performance. Results under this scheme are presented in Section 4.4.4.

As an additional baseline, we reproduce the setup used in SSV (Mustikovela et al., 2020) and fit a linear regressor to VGG16 (Simonyan and Zisserman, 2015) Conv5 features, and train it to regress the pose using the same small number of PASCAL3D+ images we use to align our predictions – see 'VGG View' results in Table 4.3.

We directly evaluate our ShapeNet-trained ViewNet model on PASCAL3D+ images. We provide results for two alignment methods, the optimal rotation using orthogonal Pro-

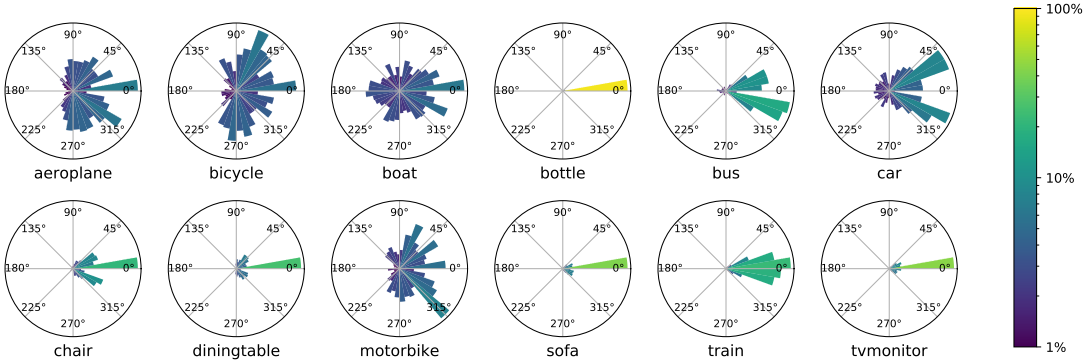


Figure 4.7: Viewpoint distribution of the evaluation split for each of the PASCAL3D+ categories. The training and validation distribution are similar

crustes, and the linear regression as used in SSV (Mustikovela et al., 2020), which takes the predicted viewpoint and applies a linear regressor to modify its predictions. Depending on the category, two behaviors can be identified: either the two alignment procedures provide similar results (e.g. bike, bottle), or the linear regression approach significantly outperforms the optimal rotation. We observe that the second behavior is correlated with categories where PASCAL3D+ contains highly biased viewpoints, i.e. where most viewpoints are clustered around a single one. We theorize that the linear regression approach can artificially boost performance in those categories by collapsing viewpoint predictions towards the common view. This can be achieved by learning zero weights for the predicted viewing angles and encoding the average viewpoint as the bias term.

Similar to our ShapeNet experiments, we also evaluate the impact of training with our cycle-based generated views. Depending on the categories, it often provides a small accuracy boost at the cost of higher median error. This median error increase could be due to the higher domain gap between generated views and real-world images.

	airplane	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv
Confidence index	1	1	1	.17	.5	1	.5	.67	.92	.42	.33	.33
Constant	30	20	21	48	30	19	21	31	22	32	34	49
VGG view*	49	58	**33**	47	32	56	37	37	32	40	36	**60**
ViewNet*	60	67	19	48	**33**	84	49	36	64	57	31	50
ViewNet	61	**72**	20	47	**33**	**86**	**78**	**40**	69	**86**	27	55
ViewNet + cycle *	53	62	21	48	29	85	55	34	67	50	34	50
ViewNet + cycle	**62**	71	21	**54**	30	85	61	37	**76**	76	**43**	58

Table 4.4: Discretized Viewpoint Accuracy at 30° error. Entries followed by a star (*) use the linear regression alignment procedure, and those without use a single global alignment.

4.4.4 Debiasing Viewpoint Evaluation

The standard evaluation used with the PASCAL3D+ dataset involves computing an average of the viewpoint prediction accuracy across the entire evaluation set and the median error in degrees for a given object class. As discussed earlier, many categories in the PASCAL3D+ dataset are strongly biased as test images are taken from a limited number of viewpoints. We illustrated this problem by creating a simple baseline, "Constant", that outputs the average viewpoint in the training set. Results in Table 4.3 show that this achieves surprisingly strong performance in some categories e.g. "bottle" (96%), "bus" (79%), "sofa" (81%),"train" (95%), and "tv" (89%). This is explained by the highly concentrated number of viewpoints in the data, as illustrated in Fig. 4.7. Note that this issue has also been reported in Figure 2 of the original PASCAL3D+ paper (Xiang et al., 2014).

To address the issue caused by this viewpoint bias, we propose a more balanced evaluation by introducing a new metric, Discretized Viewpoint Accuracy (DVA). To this end, we split the ground-truth viewpoints in the evaluation set into bins, each spanning 30° azimuth-wise and compute standard viewpoint accuracy for each bin, before averaging the results. This ensures that highly populated viewpoints do not cause performances to

be overestimated and requires a model to perform well over not only a single subset but all subsets to reach high performances. Clearly, when no samples belong to a bin, it is not possible to measure the performance in this interval. Hence we omit empty bins in the evaluation. Note that in the extreme case where all samples belong to a single bin, DVA is equal to the standard viewpoint accuracy, limiting its effectiveness as an unbiased metric. To account for this, we also compute an auxiliary dataset statistic called Confidence Index. It is defined as the fraction of bins having at least 10 samples. Hence, categories with a low confidence index are more likely to have their performances overestimated by viewpoint biases and their results should be interpreted cautiously. In Table 4.4 we present results using our DVA metric.

4.4.5 Qualitative visualizations

In order for the cyclic consistency loss to operate correctly, we need our model to be able to predict good 3D reconstructions of objects, at least on the training set. Fig. 4.8 illustrates some voxel reconstructions on the training split ShapeNet dataset for airplanes, cars and chairs. The reconstructions are of high enough quality that the object can be rendered while remaining identifiable in most cases. Outliers, like the third airplane which seems to be modeled after a shark, or the seventh, which is close to a spaceship, can still produce nonsensical reconstructions. Interestingly, the third and fifth chairs are extremely close, possessing the same color scheme and global structure, yet our model is still able to differentiate them.

We also show reconstructions of real objects from unobserved PASCAL3D+ images based on ShapeNet-trained models, for both ViewNet and ViewNet + cycle, in Fig. 4.9. This is illustrated by the fact that the rendered image from the model has the same pose as the input image. Note that background is not part of the reconstruction, as the models were trained to reconstruct objects only as explained in Section 4.4.3. The middle row of each category shows in particular how adding cycles has helped the model gain a better understanding of the object, leading to more faithful reconstruction, e.g. in the red tail

Figure 4.8: Visualization of 3D models learned from different appearance images of the training set. The top image is fed to the appearance network, and the bottom one is the output of the decoder shown in the canonical pose

of the plane or the vertical bars in the chair. The bottom row show failure cases where models failed to capture the object's appearance. We observe that even when this is the case, the viewpoint is still correctly predicted.

4.4.6 Other dataset results

Up until this point, we have only trained ViewNet on the synthetic ShapeNet dataset and evaluated it on either synthetic or real data. Our method can also be trained on real data that consists of image pairs of the same object which vary in their viewpoints. To this end,

(a) Airplanes (b) Cars (c) Chairs

Figure 4.9: Reconstructions of real images from PASCAL3D+. For each of the three object categories, the left image is the original, the middle image is the standard ViewNet reconstruction, and the right image is the ViewNet + cycle reconstruction.

we use the recently proposed Objectron dataset (Ahmadyan et al., 2021), and the Freiburg cars dataset (Sedaghat and Brox, 2015). For Objectron we train on the chair category, as it is present in ShapeNet and contains sufficiently diverse high-quality images in contrast to the other categories where images are blurry or there are too few videos. While ViewNet does not require segmentation masks at test time, it does require segmented objects as the target for training.

Objectron. We first randomly sample ten frames per videos and obtain foreground masks using two different semantic segmentation methods: DeepLabV3 (Chen et al., 2017b), trained on COCO (Lin et al., 2014) ground-truth segmentation masks and a weakly supervised method (Araslanov and Roth, 2020), trained on Objectron frames using only *image-level* labels. We start from a model pretrained on ShapeNet to prevent overfitting on the relatively low amount of instances from Objectron. ViewNet without cycles obtains 91% and 89% accuracy with 8.8° and 10.1° median error on PASCAL3D+ chairs for the supervised and weakly-supervised segmentation settings respectively. This is a significant improvement from the 83% accuracy obtained by using only the ShapeNet-trained model.

	VGG view	VpDR-Net + FrC	ViewNet	ViewNet + cycle
Accuracy (%, ↑)	56	~ 50	**61**	59
Median error (°, ↓)	25.8	29.6	**16.1**	19.1

Table 4.5: Comparison of models trained on Freiburg Cars and evaluated on PASCAL3D+.

Freiburg Cars. As the dataset only contains 48 videos, we use all frames, i.e. between 120 and 130 per instance. We also use segmentation masks obtained from a pre-trained supervised Mask R-CNN model (He et al., 2017). Results are shown in Table 4.5. ViewNet obtains stronger results than the unsupervised approach of Novotny et al. (2017). Adding our cycle loss does not improve performances as real cars exhibit specular reflections that ViewNet is unable to reproduce. Novotny et al. (2017) does not report accuracy, however, we can estimate it to be around 50% from the reported median error of 29.6°, as 50% accuracy exactly corresponds to a median error of 30°.

Layer	# channels	Kernel	Stride
Conv2D BatchNorm ReLU	64	3x3	2
Conv2D BatchNorm ReLU	128	3x3	2
Conv2D BatchNorm ReLU	256	3x3	2
Conv2D BatchNorm ReLU	512	3x3	2
Conv2D BatchNorm ReLU	512	3x3	2
Conv2D BatchNorm ReLU	512	2x2	1
Conv2D BatchNorm ReLU	variable	1x1	1

(a) Encoder architecture.

Layer	# channels	Kernel	Stride
ConvTranspose3D AdaptiveIN ReLU	512	4x4	1
ConvTranspose3D AdaptiveIN ReLU	128	4x4	2
ConvTranspose3D AdaptiveIN ReLU	128	3x3	1
ConvTranspose3D AdaptiveIN ReLU	128	4x4	2
ConvTranspose3D AdaptiveIN ReLU	128	3x3	1
ConvTranspose3D AdaptiveIN ReLU	16	4x4	2
ConvTranspose3D AdaptiveIN ReLU	16	3x3	1
ConvTranspose3D AdaptiveIN ReLU	4	4x4	2

(b) Generator architecture.

Table 4.6: ViewNet network architectures. Both f^v and f^a use the encoder architecture illustrated in Table 4.6a with output sizes 3 and 256 respectively, while f^d uses the generator architecture in Table 4.6b

4.5 Limitations and Conclusion

Using a geometrically consistent reconstruction method, ViewNet manages to learn **category-level viewpoints in a complete self-supervised manner**. The addition of mechanisms to prevent pose collapse proves to be necessary when the object categories exhibit symmetries, which is the case for most everyday objects.

Biases in the viewpoint distribution of PASCAL3D+ are also shown to have pervasive effects in evaluation, a result even more so surprising that these were reported in the original publication, but generally ignored by works that use the dataset for evaluation. Of particular concern are works that report an average performance of categories, effectively mixing up challenging tasks with absolutely trivial ones. This call for a more carefully designed evaluation dataset, specifically one that ensures a reasonably uniform distribution of viewpoints.

A particular limitation of ViewNet is its requirement for foreground masks at training time as the model is unable to extract background information from the appearance image. In experiments on real data, pre-trained segmentation models (Chen et al., 2017b; Araslanov and Roth, 2020; He et al., 2017) are used to estimate these masks. Still, the viewpoint estimator can be applied to unsegmented images at test time, making the model fast and efficient as long as a reasonably good instance segmentation model is available during training. It also relies on having image pairs during training in order to disentangle viewpoint and object appearance, which limits its real-world application to video datasets.

Finally, the object appearance is assumed to be independent from the viewpoint, but this is often violated by non-Lambertian surfaces, e.g. cars, which prevent efficient training on real data - illustrated partly by the performance drop between ShapeNet-trained models (Table 4.3) and Freiburg Cars-train models (Table 4.5). Qualitative results in Fig. 4.9 further illustrate the difficulties encountered when attempting to reconstruct real images.

While results show that models trained on synthetic data generalize well to real images,

the ideal training case would be to use real views directly. This would remove the need for CAD models, which may not exist for a specific category, allowing instead to train a model with images taken directly in context. The next chapter focuses on developing a system capable of operating more easily on real images thanks to a better reconstruction pipeline.

Chapter 5

ViewNeRF: unsupervised viewpoint estimation from real images

5.1 Introduction

Recently, neural radiance fields (NeRF) (Mildenhall et al., 2020) have achieved unprecedented quality in the 3D reconstruction and rendering of scenes. Deviating from the traditional geometrically-explicit representations, NeRF belongs to the family of implicit 3D representations (Park et al., 2019; Mescheder et al., 2019; Chen and Zhang, 2019; Sitzmann et al., 2019b; Lombardi et al., 2019), encoding spatial information in the weights of a neural network which allows them to operate at continuous 3D coordinates and hence at high image resolutions.

Despite the remarkably fast progress, illustrated by the number of recently published NeRF-inspired models (Liu et al., 2020; Barron et al., 2021; Martin-Brualla et al., 2021; Sitzmann et al., 2021; Yu et al., 2022, 2021a), these approaches typically require accu-

The main findings of this chapter have been published in BMVC 2022 (Mariotti et al., 2022).

rate camera poses during training, and are also limited to modeling a single scene. This restricts their application to controlled settings with labeled poses, or to scenes where enough high-quality camera poses can be obtained via Structure-from-Motion.

Recent extensions aim to alleviate the requirements for known camera poses, making up the family of pose-free NeRF models. Instead of assuming known camera pose for each image of the scene, these attempt to learn camera parameters along with the scene using gradient descent. This formulation limits unsupervised runs to simple forward-facing scenes, while for more complex camera distribution, a non-trivial initialization step is required (Wang et al., 2021b; Jeong et al., 2021; Lin et al., 2021; Meng et al., 2021).

Similarly, some models attempt to handle multiple instances, but they require ground-truth camera poses (Yu et al., 2021b) and some even use expensive test-time optimization in order to synthesize novel views (Jang and Agapito, 2021) (see Table 5.1).

From an analysis-by-synthesis approach, an interesting property of NeRF is its ability to synthesize high-quality views of scenes thanks to viewpoint-dependent modeling of the scene, allowing to learn complex surface properties like reflections or anisotropic lighting. These make them interesting candidates to improve over ViewNet and design an unsupervised viewpoint estimation system able to natively deal with real data.

Motivated by the limitations of both ViewNet and so-called pose-free NeRF, this chapter proposes to integrate a neural radiance field in an analysis-by-synthesis approach that leverages their powerful 3D modeling ability for unsupervised category-level viewpoint estimation.

5.2 Neural radiance fields

Implicit volumetric models propose to represent spatial data in the weights of a neural network. These were initially developed for learning shapes, in the form of occupancy networks (Mescheder et al., 2019) or signed distance function (Park et al., 2019). These

rely on a neural network to learn shapes as a classification (for occupancy networks) or regression task (for SDF). More precisely, given a point in space $\mathbf{x} \in \mathbb{R}^3$, these models learn a mapping f_s, usually modeled by a fully connected architecture, to represent the shape of an object, i.e. $f_s(\mathbf{x}) = 1$ if and only if $\mathbf{x}$ belongs to the interior of the object. By querying f_s densely in space, it is possible to have it memorize the spatial extent of the object, provided ground truth labels -i.e. a 3D model- are available. It is clearly impossible to sample every possible spatial coordinate, however, by choosing an appropriate model size, it is possible to make use of the continuity of neural networks to learn a close approximation of the shape

Implicit models quickly got extended with the addition of a rendering operation, allowing them to be supervised by images rather than 3D models. Spearheaded by Neural Radiance Fields, they learn to reconstruct specific views $\{I_k\}_{k=1,\dots,N}$ of a scene observed from the corresponding viewpoint $\{p_k\}_{k=1,\dots,N}$. More formally, to reconstruct an image pixel $I_{i,j}$, NeRF uses an implicit model f_r that is queried multiple times at coordinates of increasing depths along the ray of light $r_{i,j}$ corresponding to $I_{i,j}$. f_r is parameterized to output the color and density of each point a tuple (σ, c), providing an approximation of the objects and textures encountered along the ray. Then, they make use of a rendering operation (Kajiya and Von Herzen, 1984) to obtain the final value $I_{i,j}$, integrating the values of the ray according to their density and past transmittance. For a ray $\mathbf{r}(t) = \mathbf{o} + t\rho$ with origin $\mathbf{o}$ and direction ρ, corresponding to pixel $I_{i,j}$, the rendering equation between near and far bounds t_n and t_f is:

$$\hat{I}_{i,j} = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))c(\mathbf{r}(t)), \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(t))\right) \tag{5.1}$$

$T(t)$ represents the accumulated transmittance up to the spatial coordinate $\mathbf{r}(t)$, i.e. the probability that the light travels up to $\mathbf{r}(t)$. It is easy to see that $T(t_n) = 1$ and that T converges to 0 as the ray travels through regions with large density σ, providing a model for occlusion. In practice, the integral cannot be exactly computed as it would require

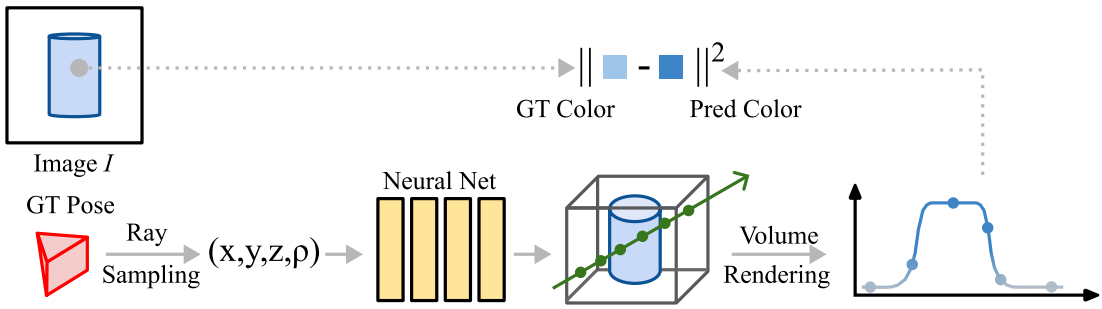


Figure 5.1: Overview of the NeRF pipeline

an infinite number of queries to the neural network. Therefore, it is estimated using a Riemann sum between the multiple values queried along the ray.

A particularity of NeRF is to model the texture of objects according to their viewing direction, in order to account for complex light patterns like non-Lambertian reflections. To this end, Eq. (5.1) is modified by having the color depend on the ray direction along with the spatial coordinate, i.e. $c(\mathbf{r}(t), \rho)$. However, to preserve geometry, the density $\sigma(\mathbf{r}(t))$ keeps depending only on the location. The predicted value $\hat{I}_{i,j}$ is finally compared with the ground truth pixel color $I_{i,j}$ to provide supervision. A schematic representation of a NeRF model is shown in Fig. 5.1.

An important aspect of implicit models is that trying to predict volumetric data directly from spatial coordinates usually produces poor reconstruction, due to the low dimension and low frequency of the input space (Rahaman et al., 2019). To make it more expressive, common practice is to perform feature augmentation using cosine embedding, mapping spatial coordinates to a higher dimension space using multiple periodic functions with increasing frequencies.

Formally, for a scalar $x \in \mathbb{R}$, the cosine embedding of order k $\gamma_k(x)$ is defined as:

$$\gamma_k(x) = \cos(2^k \pi x), \sin(2^k \pi x) \tag{5.2}$$

In practice, multiple orders are used over each spatial coordinate, yielding, for a point

$\mathbf{x} \in \mathbb{R}^3$:

$$\gamma(\mathbf{x}) = \{\gamma_k(x_i) | k = 0, \ldots, n; i = 0, 1, 2\} \tag{5.3}$$

Where N is the number of frequencies to use, typically between 3 and 10. $\gamma(\mathbf{x})$ is then used as input of f_r instead of $\mathbf{x}$.

The global geometry for the scene coupled with analytical rendering grants NeRF relatively strong 3D consistency - although not perfect, as noted in (Zhang et al., 2020). Along with its flexible resolution, this property makes it particularly suitable as a reconstruction model in a pose estimation pipeline.

5.3 Related work

Unsupervised viewpoint estimation. Despite the large body of supervised methods (Rad and Lepetit, 2017; Kehl et al., 2017; Tulsiani and Malik, 2015; Choy et al., 2016), viewpoint estimation is still a challenging task due to the cost of building large labeled datasets. Hence a growing number of methods attempt to limit the amount of supervision needed at training time. SfM approaches such as COLMAP (Schonberger and Frahm, 2016) use multi-view geometry to infer camera poses using only images, but are limited to single scenes, require many views, and are thus unsuitable for estimating poses across category-centric datasets. Recently several deep learning pose-estimation methods have been proposed for various levels of pose supervision including semi-supervised (Mariotti and Bilen, 2020; Wang et al., 2021a), few/zero-shot learning (Xiao and Marlet, 2020; Banani et al., 2020; Goodwin et al., 2022) and unsupervised ones (Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018; Mariotti et al., 2021; Mustikovela et al., 2020).

Most related to ours, the unsupervised methods learn to disentangle category-level pose and appearance using an analysis-by-synthesis pipeline. ViewNet (Mariotti et al., 2021) generates a voxel-based reconstruction of a specific object instance and renders it from the predicted viewpoint. This approach is limited by the spatial resolution of the voxel grid

	Pose-free 360° training	Real data 360° training	One shot pose on new views	Multiple scenes
NeRF (Yen-Chen et al., 2021)				
INeRF (Yen-Chen et al., 2021)				
NeRF– (Wang et al., 2021b)†				
BARF (Lin et al., 2021)				
SCNeRF (Jeong et al., 2021)		✓		
GaRF (Chng et al., 2022)†				
GNeRF (Meng et al., 2021)	✓		✓‡	
CodeNeRF (Jang and Agapito, 2021)				✓
Ours	✓	✓	✓	✓

Table 5.1: Comparison of pose-free NeRF methods on 360° scenes. Most of these approaches require ground-truth poses or initial estimates.†Untested on 360° scenes. ‡Contains a pose estimator but it is not used during evaluation.

making it unable to reconstruct fine details and viewpoint-dependent illumination effects. SSV (Mustikovela et al., 2020) adopts a generative approach, using 3D latent feature maps to represent the scene. However, it fails to apply consistency between inferred geometry and pose, resulting in noisy viewpoint estimation due to its decoder's flexibility and overfitting to geometrically implausible poses.

Generic Neural Radiance Fields (NeRF). 3D data is traditionally represented using meshes (Kanazawa et al., 2018; Kato et al., 2018), voxels (Yan et al., 2016; Tulsiani et al., 2017, 2018; Sitzmann et al., 2019a), or point clouds (Insafutdinov and Dosovitskiy, 2018). Recently, the implicit representations (Park et al., 2019; Mescheder et al., 2019; Chen and Zhang, 2019; Sitzmann et al., 2019b; Lombardi et al., 2019) have emerged as an effective tool in 3D modeling. They represent 3D data implicitly in the parameters of a fully connected neural network, that takes 3D coordinates as input and attempts to predict properties such as their occupancy and color of the scene at the specified 3D location. NeRF (Mildenhall et al., 2020) models 3D scenes by mapping both 3D coordinates and

the viewing direction to RGBA space, achieving breakthrough performances in novel view synthesis. Multiple works extend the NeRF paradigm towards higher reconstruction quality (Barron et al., 2021; Zhang et al., 2020) and faster runtime (Liu et al., 2020; Yu et al., 2022, 2021a). Two directions, particularly related to object pose estimation, are to adapt the NeRF beyond the strict single-scene setting and to remove dependency on the training time poses.

Flexible NeRF. NeRF in the Wild (Martin-Brualla et al., 2021) learns to aggregate views of the same scene taken in different settings by learning image-specific embeddings, while Nerfies (Park et al., 2021) learn to deform rays to represent deformable objects. PixelNeRF (Yu et al., 2021b) learns scene-based dense embeddings to represent multiple scenes. CodeNeRF (Jang and Agapito, 2021) disentangles shape and texture across instances from the same category. These methods, however, require ground-truth camera poses during training. Though several generative methods (Schwarz et al., 2020; Niemeyer and Geiger, 2021; Gu et al., 2021) have been proposed to model object categories, they are unable to estimate poses and employ neural rendering that can violate the scene geometry as in SSV (Mustikovela et al., 2020). Unlike them, through the use of an implicit 3D representation and analytical rendering, our reconstructions are 3D consistent.

Pose-free NeRF. Multiple pose-free NeRF methods (Wang et al., 2021b; Lin et al., 2021; Jeong et al., 2021; Chng et al., 2022) attempt to learn camera poses during training by refining initial pose estimates through gradient coming from the NeRF model itself. However, these methods are only pose-free on forward-facing scenes, needing COLMAP as initialization for 360° scenes. While those methods can be used to retrieve the pose of new images under certain conditions (Yen-Chen et al., 2021; Jang and Agapito, 2021), the process they use for it involves expensive test-time optimization. By comparing image reconstructions based on initial noisy pose estimates to the target image, they perform many iterations of gradient descent on the camera parameters to gradually align the two images. In addition to being a slow process, this approach can get trapped in local minima

in a multi-object setting. In comparison, our model can predict poses of unseen instances in a single shot. A notable exception from other pose-free NeRF is GNeRF (Meng et al., 2021), which can operate without initialization thanks to its adversarial training, but is limited to single scenes and is slow to train as a result of the additional GAN-based objectives. A comparison to related NeRF-based approaches is shown in Table 5.1.

5.4 Method

Here we outline the main components and training procedure for our ViewNeRF model. Unlike existing methods, ViewNeRF is capable of estimating the pose for held-out images and can be trained on images from multiple instances (i.e. from different scenes) of the same object category *without* requiring any ground-truth pose supervision.

Our goal is to estimate viewpoint/camera pose of an unseen object instance from a known category (e.g. car, chair, etc.) in an image. To this end, we wish to learn a function, f_p that takes an image I as input and outputs the corresponding viewpoint represented as the rotation and translation, i.e. $p = f_p(I)$. As ground-truth viewpoints are not available for training, we treat learning pose prediction as an image reconstruction problem.

Similar to existing work (Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018; Yu et al., 2021b; Jang and Agapito, 2021; Mariotti et al., 2021), we exploit multi-view information in the form of image pairs. Specifically, given N unlabeled image pairs $\{I_n, I'_n\}_{n=1}^{N}$, where each pair contains source and target images of the same object instance which differ only in their viewpoints, our objective is to reconstruct the target image I from the source image I'. Clearly, one needs a good estimate of the viewpoint of I in order to reconstruct it from I'. During training, we reconstruct the target image I using its estimated viewpoint $f_p(I)$ and the appearance information extracted from our viewpoint-independent appearance encoder f_a for the source image I'. For rendering, we pass the pose and appearance information to a NeRF-based decoder f_r, to reconstruct the target image I, and minimize an image reconstruction loss to simultaneously learn the weights of f_p, f_a, and f_r which

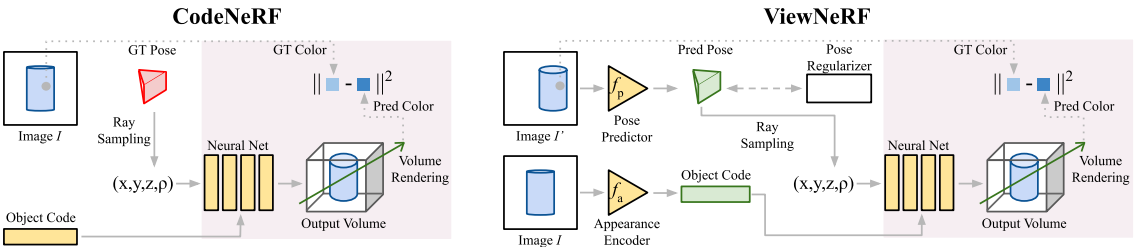


Figure 5.2: (Left) CodeNeRF (Jang and Agapito, 2021) requires ground-truth pose information at training time and performs expensive direct optimization for each object appearance code. In practice, CodeNeRF also enforces object codes for distinct views of the same object to be the same which provides some multi-view signal. (Right) In contrast, our ViewNeRF approach is fully self-supervised by making use of a separate pose predictor f_p and appearance encoder f_a that can be applied to any image in a single-shot fashion.

are instantiated as neural networks,

$$\frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(f_r(f_a(I'_n),f_p(I_n)),I_n)+\lambda\mathcal{L}_{\text{reg}}(f_p(I_n)). \tag{5.4}$$

$\mathcal{L}$ is a loss function that measures the difference between the reconstructed and target images, and $\mathcal{L}_{\text{reg}}$ is a regularization term applied to the viewpoint predictions, which is weighted by a scalar λ. The training pipeline for our model, ViewNeRF, is shown in Fig. 5.2 (right).

Clearly, the target image cannot be successfully reconstructed without its viewpoint information. However, in the case of an arbitrary decoder, where the appearance and viewpoint encodings are passed through multiple arbitrary nonlinear transformations, there are at least two challenges: it is not guaranteed that (i) the estimated viewpoints are disentangled from the appearance and, if they are, (ii) the estimated viewpoints are geometrically meaningful transformations. As a result, it is crucial that the decoder utilizes the estimated viewpoint in a geometrically consistent way.

5.4.1 NeRF decoder - f_r

Object appearance conditioned NeRF decoder. Standard NeRFs are trained to model a single 3D scene, effectively memorizing its shape and appearance from multiple viewpoints. In a category-based setting, this would mean training an individual model for each object instance, which would be very time-consuming and no information across instances would be shared. Hence, a better approach is to implement a conditioning mechanism to allow the NeRF decoder f_r to reconstruct specific instances. This conditioning should be pose-agnostic in order to let the pose predictor f_p learn to disentangle pose from appearance. Therefore, we adopt a strategy inspired by ViewNet (Mariotti et al., 2021) and CodeNeRF (Jang and Agapito, 2021) where object instances are fully described by a latent object code $\mathbf{a}$. Similar to (Jang and Agapito, 2021), $\mathbf{a}$ is mapped at different depths of the NeRF model to condition its activations, and similar to (Mariotti et al., 2021), $\mathbf{a}$ is predicted by an appearance network f_a that learns a global latent space shared between all object instances.

5.4.2 Pose estimator - f_p

Conventional NeRF methods require ground-truth camera poses during training. Recent extensions (Yen-Chen et al., 2021; Wang et al., 2021b; Jeong et al., 2021; Lin et al., 2021; Chng et al., 2022) allows for estimating camera poses with NeRF by letting reconstruction gradients flow to the camera parameters. However, this approach suffers from two issues: i) it is computationally expensive, requiring hundreds of steps to converge, if at all, and ii) it can get stuck in local minima, limiting its operation to forward-facing scenes when a reasonable pose initialization is not available. Following the recent advances in viewpoint estimation (Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018; Mustikovela et al., 2020; Mariotti et al., 2021), we posit that a better solution is to estimate poses directly from images using a pose predictor f_p. This enables fast prediction during inference and generalizes efficiently to new object instances. However, f_p is still subject to local minima and may not receive meaningful gradients from f_r, as reconstruction errors can

arise either from the reconstruction process or an incorrect pose prediction. This can lead to the collapse of pose predictions and to degenerate solutions where the model relies only on the appearance encoding **a** to reconstruct I'. Next, we introduce two mechanisms to prevent this.

5.4.2.1 Multi-hypothesis predictions.

We supply f_p with a multi-head predictor as used in (Insafutdinov and Dosovitskiy, 2018; Mariotti et al., 2021). During training, we let the pose estimator output multiple hypotheses $f_p(I) = p_1, \ldots, p_K$, and each of them is fed to f_r to produce a low-resolution reconstruction. These are then compared to the target, and the pose p^* resulting in the best reconstruction is selected. p^* is then passed again to the NeRF decoder, this time at full resolution. For inference, a student head is jointly trained to predict p^*, removing the need for multiple outputs (Mariotti et al., 2021).

5.4.2.2 Pose regularization.

Multiple pose predictions alone are not always sufficient to prevent training collapse or instability as all heads can still predict the same pose. Inspired by generative models like (Niemeyer and Geiger, 2021; Gu et al., 2021) that sample poses during training, we encourage the predicted pose distribution to follow a prior distribution $\mathcal{P}$. As generative models do not aim to reconstruct images from a specific viewpoint, they can directly sample a pose from the prior $p \sim \mathcal{P}$ and use it to generate an image. However, this is unfeasible in our case, as a random pose would not match that of the specific image that we would like to reconstruct. Instead, we attempt to match batch-wise distributions, following the assumption that predicted poses across a mini-batch of images should closely follow the pose prior $\mathcal{P}$, similarly to how batch normalization approximates dataset statistics using a batch of inputs (Ioffe and Szegedy, 2015).

Specifically, given a batch of B predicted poses $p^*_{1,\ldots,B}$, we sample K pseudo-targets $p'_{1,\ldots,K} \sim \mathcal{P}$ and compute for each p'_i its closest match p^*_j in the batch. Finally, the distance

Algorithm 2: Pose regularization

Input : Minibatch of predicted poses $p^*_{1,\dots,B}$, Prior distribution $\mathcal{P}$, number of samples K

Output: Regularization loss $\mathcal{L}_{reg}$

$\mathcal{L}_{reg} = 0$

for $i \in 1 \dots K$ **do**

$p' \sim \mathcal{P}$; `// draw a pseudo-target from` $\mathcal{P}$

dists $= ||p^* - p'||$; `// distance between each predicted pose and p', size` B

weights $= SoftMax(-\text{dists})$; `// Batch-wise SoftMax`

weighted_dists $=$ weights $*$ dists

$\mathcal{L}_{reg} += \frac{1}{K} * Avg(\text{weighted\_dists})$; `// Batch-wise Average`

$||p'_i - p^*_j||^2$ is added to the loss, i.e.:

$$\mathcal{L}_{\text{reg}} = \frac{1}{K} \sum_{i=1}^{K} \min_{j=1\dots B} ||p'_i - p^*_j||^2 \tag{5.5}$$

This prevents the collapse of all predictions to a single point while being very cost-efficient. To prevent unnecessary noise, the regularization weight λ in Eq. (5.4) is progressively tuned down during training.

We provide pseudo-code for our pose regularization method in Algorithm 2. Note that K might not be equal to B. In practice, instead of using the minimal distance, we use a soft minimum to decrease noise.

Regularization strength λ starts at 1 and undergoes exponential scheduling, being multiplied by 0.1 every 10 epochs before being turned off at epoch 30.

5.4.3 Reconstruction objective

Reconstructing full-resolution images requires millions of queries to the NeRF decoder and hence is expensive, so NeRFs are usually trained by sampling a subset of pixels per image, per iteration. This strategy works well when using ground-truth camera poses, as each sampled pixel corresponds to one exact ray that will stay constant during training. However, when jointly estimating poses and training the NeRF model, it can introduce a significant amount of noise, as the randomly selected pixels might not contain enough relevant information to recover incorrectly estimated poses. In particular, some object categories such as cars can exhibit symmetries that can only be broken by focusing on fine details (e.g. color of the headlights).

To this end, we deviate from the standard NeRF training procedure and instead use reconstructions of the entire image, however, in low-resolution coupled with a perceptual loss (Johnson et al., 2016) (denoted as $\mathcal{L}$ in Eq. (5.4)) that provides a finer structure-preserving objective. As the perceptual loss aims to match activation maps from a pretrained network, it is more sensitive to salient image features like edges that would be missing or misplaced under wrongly estimated poses.

5.5 Experiments

5.5.1 Implementation details

We use EfficientNet (Tan and Le, 2019) backbones for our pose and appearance encoders f_p and f_a, and an efficient NeRF model with two fully connected layers with 128 dimensions for the decoder. Note that our decoder is designed to be significantly smaller than the one used in (Meng et al., 2021) and (Jang and Agapito, 2021) to discourage unnecessarily complex mappings and to make it more efficient in training and evaluation. Computational efficiency is particularly important in our experiments, as we use perceptual loss for reconstruction error which requires producing a full image.

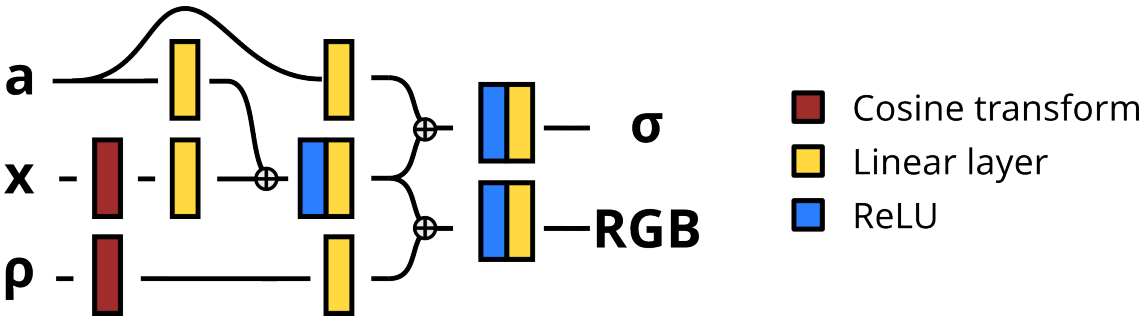


Figure 5.3: NeRF archtitecture used in ViewNeRF. **a** depicts the appearance embedding, while **x** and **ρ** are the spatial coordinate and viewing direction

Camera pose prediction To link the predictions of f_p to real-world poses, we need to ensure it can be interpreted as such. Similar to ViewNet and GNeRF, we formulate pose as a point on the 3D unit sphere $\mathcal{S}^2$ from which we derive a camera matrix using a Gram-Schmidt orthogonalization process described in Algorithm 1. While other representations like quaternions are possible, this provides a simple way to enforce constraint and shows good properties for optimization (Zhou et al., 2019).

For synthetic datasets, we follow GNeRF's assumption that the object is located at the center of the scene where the camera is pointed to, and that the camera is held upright, i.e. it is aligned with the z vector in world coordinates. On Freiburg cars, this is not strictly verified. Therefore, we additionally allow our viewpoint estimator to predict a camera distance, target point, i.e. a point along the camera principal axis, and an upwards direction to account for in-plane rotation. To regularize training, we enforce those predictions to be close to their non-learnable values used in the synthetic case for the first 10 epochs of training.

NeRF architecture The architecture of our NeRF decoder is depicted in Fig. 5.3. To encourage 3D consistency, we use cosine embeddings of size 8 and 1 for **x** and **ρ** respectively. They are then mapped with linear layers to the inner dimension of the model which is 128.

	Supervised				Unsupervised			
	CodeNeRF, w/o init		CodeNeRF, w/ init		ViewNet		Ours	
	car	chair	car	chair	car	chair	car	chair
Accuracy at 10° (%, ↑)	08.5	03.4	**82.1**	**60.2**	61.2	76.7	**70.0**	**82.8**
Median rotation error (°, ↓)	115	108	**3.53**	**7.70**	6.54	4.25	**5.71**	**4.18**
Median translation error (%, ↓)	139	134	**5.9**	**13.9**	n/a	n/a	**8.0**	**6.4**

Table 5.2: Multi-instance results on Shapenet-SRN (Sitzmann et al., 2019b). CodeNeRF pretrained models were kindly provided by the authors. When initialized, pose estimates were randomly drawn within 30° of the ground truth, where **bold** results indicate the best model per category.

Evaluation Following the evaluation in (Mariotti et al., 2021), we align our estimated camera poses with the ground-truth labels by solving an orthogonal Procrustes problem, as poses are predicted up to an arbitrary rotation. As our main goal is to estimate viewpoint from single images rather than high-fidelity reconstruction, we evaluate our method in terms of viewpoint accuracy, reporting rotation and translation errors. The other approaches we compare to (Meng et al., 2021; Jang and Agapito, 2021; Mariotti et al., 2021) each use their own sets of metrics making direct comparison difficult.

Hence, for a fair comparison, we re-evaluate their models and report viewpoint accuracy with a 10° threshold, along with median rotation and translation error for each method in all experiments. Taking the median instead of the average provides a less noisy estimate in the presence of strong symmetries (Tulsiani and Malik, 2015). Translations errors are normalized by the camera distance to the origin to account for scaling differences.

5.5.2 Multi-instance results

In Table 5.2 we first evaluate our ViewNeRF approach on the ShapeNet-SRN dataset (Sitzmann et al., 2019b) which contains renderings of ShapeNet (Chang et al., 2015) cars and chairs. We compare to CodeNeRF (Jang and Agapito, 2021), a *supervised* NeRF-

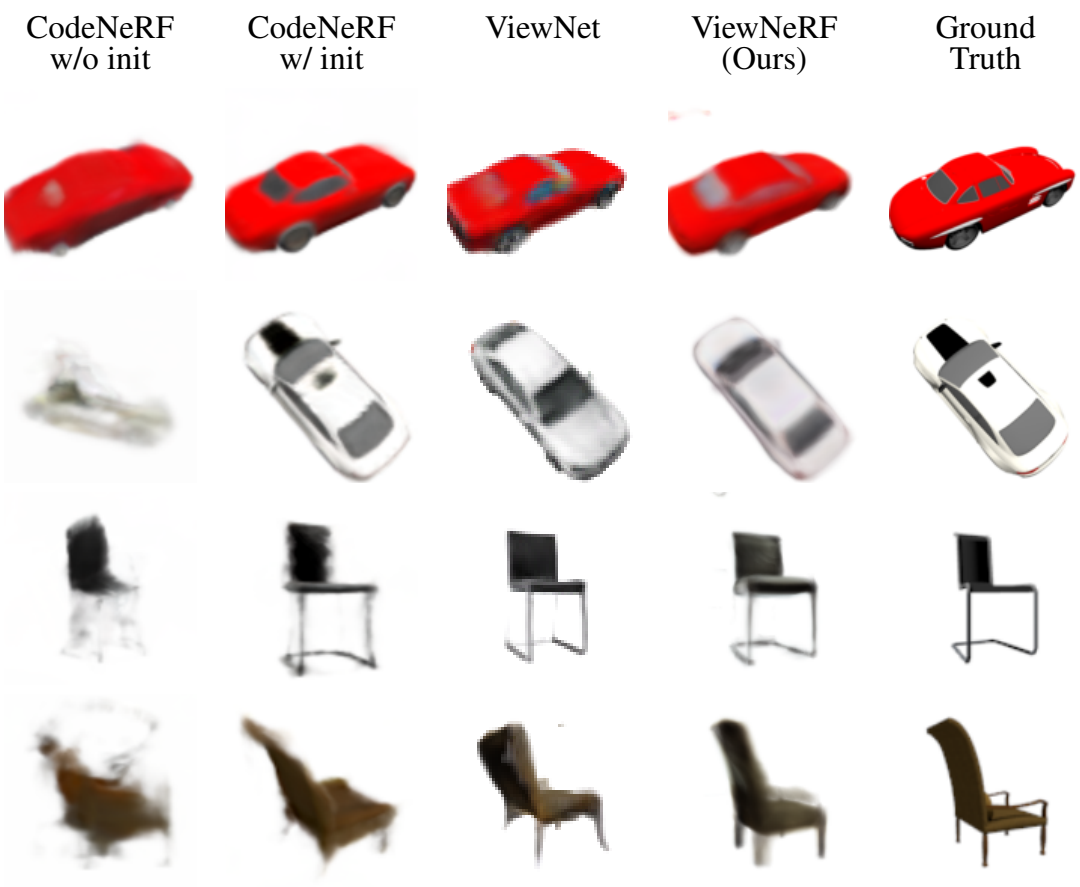


Figure 5.4: Comparison of reconstructions from the supervised CodeNeRF, unsupervised ViewNet, and our unsupervised ViewNeRF.

based model and the *unsupervised* voxel-based ViewNet (Mariotti et al., 2021). While ViewNet reports results on ShapeNet, it is only trained on a limited set of viewpoints, i.e. the elevation of views only spans $[-20^{\circ}, 40^{\circ}]$, instead of the full range in ShapeNet-SRN. Hence, we retrain it using code from the authors on this new data split.

CodeNeRF requires expensive test-time optimization to perform pose estimation and only reports results for a *single* object instance in their paper, i.e. not multiple instances from the same category, starting from hand-selected poses[1]. Therefore, we re-evaluated it on each test instance, under two settings, a realistic one where the starting pose is uniformly

[1]Confirmed via correspondence with the authors.

sampled according to the training distribution ('w/o init'), and an easier setting, in which the initial pose is chosen to be within 30° of the ground-truth ('w/ init').

The results in Table 5.2 illustrate that CodeNeRF, despite being trained with ground-truth pose, is unable to properly estimate pose when the initial estimate is noisy. Since both pose and object embeddings have to be jointly optimized, the process can converge to a degenerate solution, relying mostly on the appearance embeddings rather than pose to minimize the reconstruction error. Test-time reconstructions shown in Fig. 5.4 confirm this. While reconstructions with good initializations are accurate, noisy initialization results in poor reconstructions. Finally, compared with ViewNet, our approach reaches higher pose prediction performances by making fewer gross pose errors, e.g. ViewNet predicts the wrong orientation for the car in the second row of Fig. 5.4.

5.5.3 Real scenes results

Here we demonstrate ViewNeRF's ability to work on real images using the Freiburg cars dataset (Sedaghat and Brox, 2015). The target car instance in each image is first segmented using MaskRCNN (He et al., 2017), and out of the 48 scenes, the first 40 are used for training, the next three for validation, and the remaining five for testing. As the data is only labeled with weak viewing direction information, we only report rotation-based metrics. The results in Table 5.3 illustrate a large gap between the performances of our approach and ViewNet. We mostly attribute it to ViewNet's inability to model the complex illumination patterns (e.g. reflections) on real cars. Qualitative results in Fig. 5.5 further illustrate this, i.e. reconstructions from ViewNet, while having sharper colors are very noisy. In addition, it seems that ViewNet is unable to differentiate the front from the back of the red car.

In Fig. 5.6, we provide extra comparison between our method and ViewNet on Freiburg cars, by sampling views at a 45° interval around reconstructed test instances. It is apparent that ViewNet does not manage to reconstruct the back of the car correctly.

Figure 5.5: Comparison of reconstructions from the unsupervised ViewNet and ViewNeRF on held-out test instances from the Freiburg Cars dataset.

	ViewNet	Ours	Ours, no predictor	Ours, MSE	Ours, singlehead	Ours, no reg.	Ours w/ bg, failed	Ours w/ bg, success.
Acc@10° (%, ↑)	50.0	**73.5**	00.8	04.1	11.4	54.5	01.7	**21.1**
Med. rot. err. (°, ↓)	9.99	**8.05**	90.8	91.0	67.4	9.09	85.6	**20.5**

Table 5.3: Comparison to ViewNet, ablated versions, and unsegmented runs of our ViewNeRF method on the Freiburg Cars dataset.

Ablated models To validate our design choices, we also evaluate multiple ablated versions of our model on the Freiburg car dataset. We evaluate four variations: (i) removing estimators and trying to learn poses directly with backpropagation, i.e. standard pose-free NeRF training, (ii) using MSE loss instead of the perceptual loss for reconstruction, (iii) using a single pose hypothesis, and (iv) removing pose regularization $\mathcal{L}_{\text{reg}}$. All ablations produce worse performance, with the first three resulting in catastrophic failure (Table 5.3).

5.5.4 Single instance results

Finally, we evaluate ViewNeRF in the single-scene setting with full 360° rotations on the synthetic-NeRF datasets (Mildenhall et al., 2020) used in GNeRF (Meng et al., 2021).

Figure 5.6: Reconstructions of Freiburg car test instances for ViewNet and ViewNeRF. The bottom row is the frame used for providing appearance embedding.

	GNeRF			Ours		
	Acc (%,↑)	MR (°,↓)	MT (%,↓)	Acc (%,↑)	MR (°,↓)	MT (%,↓)
Chair	**100**	**2.645**	**4.401**	**100**	3.012	4.680
Drums	**98.5**	**3.307**	**5.489**	80.5	5.212	8.356
Hotdog	74.5	7.120	11.12	**96.0**	**2.412**	**3.898**
Lego	**91.5**	5.153	8.313	87.0	**4.659**	**7.571**
Mic	**97.5**	**3.022**	**4.787**	93.5	4.169	6.823
Ship	15.0	28.23	43.56	**69.5**	**6.674**	**9.946**

Table 5.4: Single scenes, NERF synthetic scenes. Acc: Accuracy at 10°, MR: Median rotation error, MT: Median translation error. GNeRF models were retrained using published code.

GNeRF is closely related to our model as it can estimate pose with a simple single forward pass. However, the pose results reported in the original GNeRF paper are from the training split of the data, where they are learned using a mixture of gradient descent optimization and soft-labeling. In Table 5.4 we instead evaluate the GNeRF pose predictor on the test split in order to perform a fair comparison with our model. We observe that in spite of its strong performance on the training split and the much larger model it uses, GNeRF results are broadly comparable to ours during inference. This can be explained by the size of the training set, that only contains 100 samples, hinting towards overfitting. The ship scene exhibits a strong rotational symmetry and is thus particularly challenging for both methods.

5.5.5 Unsegmented scenes study

An additional advantage of NeRF over explicit representation is its ability to model spatial information at arbitrary coordinates, although the cosine embedding imposes a periodicity in the input space (Eq. (5.2)). This allows them to deal with almost unbounded scenes, or

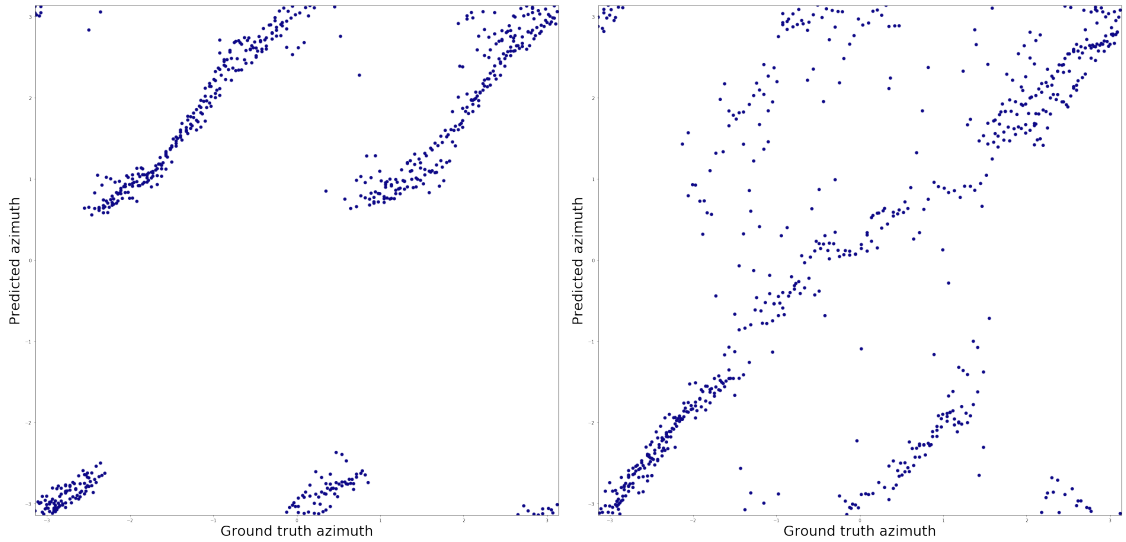


Figure 5.7: Test predictions for two different runs of ViewNeRF on unsegmented Freiburg Cars. Left figure pictures a failed run, and the right a more successful one.

Figure 5.8: ViewNeRF reconstructions of Freiburg car test instance with background. The top row is a failed run, the bottom is a successful one.

in practice, large distances, which is often too memory intensive for explicit representations, in particular for voxels used in the previous chapter. This potential grants them the ability to use unsegmented images during training, by asking the model to also reconstruct backgrounds.

Naively attempting to reconstruct backgrounds in 360° scene is often met with poor results due to the additional large amount of information needed to model, and specific extensions have been developed to provide better background modeling capabilities (Zhang et al.,

2020). In the case of category-level viewpoint estimation, the natural assumption that all instances can easily be aligned is broken, as there is no trivial way to align complete scenes (object + background). For instance, a dataset of car-based scenes like Freiburg cars can have a tree in front of the car in one scene, and another tree behind the car in another. Aligning the scenes with respect to the cars would misalign the trees and vice-versa.

Furthermore, in the segmented case, all images are semantically close, as they contain a unique instance of the object category, making it relatively straightforward for a single model to reconstruct. Backgrounds introduce a vast amount of extra information that is *not* shared between scenes, therefore making it much more complex to reconstruct.

Nonetheless, we can still attempt to train ViewNeRF without segmented targets. Results show a large training instability, with some runs managing to capture the common information between scenes, i.e. a car in the center and some background further away, while others fail to do so and fall in symmetry-induced local minima. Fig. 5.7 illustrates the model predictions for a failed and a more successful one. Predictions in both cases exhibit strong symmetry issues, to the point that quantitative results are poor for both, although significantly worse in the failed case (Table 5.3). Qualitative results shown in Fig. 5.8 further illustrate this phenomenon, with however some reasonably decent reconstructions in the second case, hinting towards the possibility of success given a properly designed model and enough training samples.

5.6 Limitations and Conclusion

Although ViewNeRF outperforms prior works in category-based viewpoint estimation, it also has certain limitations that hinder its applicability to more complex scenarios. While the requirement for multiple views is common, it limits its application to a few multi-view datasets. It would be desirable to build a model that can learn object categories from different instances without requiring multi-view data. While generative methods

(Nguyen-Phuoc et al., 2019; Mustikovela et al., 2020; Niemeyer and Geiger, 2021; Gu et al., 2021) possibly possess this ability, they also employ neural-based decoders that hurt 3D consistency. Another limitation is the need for segmenting foreground objects from cluttered backgrounds. We observed that when using unsegmented views, complex backgrounds forming the majority of an image prevented NeRF to pay enough attention to the object to capture the details needed for estimating category-level pose. Forcing the reconstruction objective to focus less on the background during training by using two separate NeRFs as in (Niemeyer and Geiger, 2021) could be a potential solution.

By using a NeRF-based reconstruction pipeline, ViewNeRF is able to produce higher-quality images and more importantly, model the complex light patterns of real images, allowing it to work natively on real cars. The addition of an extra pose regularization method helps make the training more stable, and comparisons against other pose-free NeRF models show the benefits of using a pose estimator rather than a costly and unstable gradient-based approach to recover poses of unseen images.

Chapter 6

Conclusion

6.1 Impact

The work carried out in this thesis presents a practical approach towards unsupervised viewpoint estimation of object categories, articulated around some of the difficulties encountered when undertaking such a task. In particular, it highlights some interesting points that could be useful for future research:

Experimental results show the possibility of **unsupervised pose estimation** or, depending on precise nomenclature, self-supervised. Building from equivariant representation (Worrall et al., 2017; Rhodin et al., 2018) and unsupervised 3D reconstruction models (Tulsiani et al., 2018; Insafutdinov and Dosovitskiy, 2018), the models developed here are able to use simple images as a reconstruction target and learn and interpretable pose space using differentiable rendering. In contrast with more classical methods like structure from motion (Schonberger and Frahm, 2016) they use reconstructions to naturally align all images of an object category.

Importance of geometry-preserving models Experiments ran at the end of Chapter 3 show that in complex scenarios, 2D decoders will find shortcuts to reconstruct images,

often by bringing together similar-looking views like the left and right sides of a car.

Somewhat going against the current trend of coupling 3D representations with a neural decoder (Nguyen-Phuoc et al., 2019; Mustikovela et al., 2020; Niemeyer and Geiger, 2021; Gu et al., 2021; Chan et al., 2022), it seems like reliable pose can only be extracted with *strict geometric consistency* between the decoded views. While CNN-based decoders have a positive impact on image quality, their unconstrained nature allows for geometry-destructing transformations to take place, leading to possibly nonsensical poses. Interestingly, this fact is only seldom reported in publications, although it is relatively easy to observe when evaluating these models on datasets with 360° rotations. Unfortunately, the standard evaluation sets for these models appear to be human faces or at best forward facing scenes. Coupled with a focus only on image quality metrics, this makes the issue go relatively unnoticed in the field, although a few recent works try to propose solutions for it in the form of more constrained decoders (Gu et al., 2021; Karras et al., 2021). Still, it appears these are currently not enough to grant 3D consistency to 2D convolutional decoders. Hence, until 2D decoders are proven to be geometry-preserving, preference should be given to models with geometry-preserving rendering processes.

Evaluation biases and unsupervised approaches This problem mostly crept up due to the **inadequacy of evaluation process**, which is i) only concerned about image quality, and ii) using too simplistic settings to evaluate pose (i.e. forward facing only). In a more pervasive way, PASCAL3D+ (Xiang et al., 2014) is still used as a viewpoint prediction benchmark, despite the fact that it contains multiple categories that are so biased they can be solved with constant predictors. At best, supervised models simply reproduce training biases on the evaluation set, which is a standard concern in all machine learning approaches. Were the two distributions different, models would perform poorly and the issue would automatically be acknowledged. For unsupervised models, however, this can be harder to detect, as it only plays a role during the alignment of the predictions, i.e. a system can predict perfectly uniform viewpoints and still be made to overfit the evaluation

distribution.

This issue should be seen as a major concern in the field of object pose estimation and is partly addressed by the recent release of pose-focused video dataset (Ahmadyan et al., 2021; Reizenstein et al., 2021). Though it can be mitigated to some extent by adopting slightly more robust metrics, this calls for more care when collecting data, and for more critical analysis of results when reporting on biased data. Even on synthetic datasets like ShapeNet (Chang et al., 2015), biases can creep up due to the lack of care taken when rendering images. As an example, the data used in SRN (Sitzmann et al., 2019b) follows different pose distributions between the chair and car categories. For chairs, a set of 100 uniformly distributed viewpoints is drawn per model, while in cars, the same 100 uniformly sampled poses are used for each model. While both can be argued to be uniformly sampled, the discrepancy between the two can lead to wrong assumptions about the data.

6.2 Limitations

The different methods described in this thesis are still hindered by several limitations:

First and foremost, they rely on a common set of **assumptions about poses** of the object at training time and during inference, limiting them to restricted scenarios. The need for an unoccluded single object in the frame can to some extent be emulated by an object detection system, but these might not be available for certain categories, or simply can fail. Occlusions can theoretically be synthetically added by augmentation during training, but unoccluded views are still needed for reconstruction targets. The requirement for spherical distribution of camera poses can be relaxed to a degree, e.g. on Freiburg cars (Sedaghat and Brox, 2015), but break when objects go partially out of frame, either due to the camera pointing elsewhere or coming too close to the object. Finally, the assumption of an upright-held camera is verified on many datasets and allows models to run in ideal circumstances, but would prevent it from working if an object is knocked over.

Another hard limitation is the need for **segmented training images**. While inputs to the viewpoint predictors need not be segmented, making the model applicable to unsegmented images during inference, the target image has to contain only the object. While this practice is quite common, even in recent NeRF methods (Mildenhall et al., 2020; Wang et al., 2021b; Yu et al., 2021b; Jang and Agapito, 2021; Meng et al., 2021), it relies in practice on a pretrained instance segmentation model. While this limitation is strict in the case of ViewNet, as perspective transformers are unable to reconstruct backgrounds due to their limited spatial extent, several NeRF extensions have been proposed that make it possible to model backgrounds with reasonable accuracy by adding an additional model in charge of distant objects (Zhang et al., 2020; Niemeyer and Geiger, 2021). However, in the case of category-level models, backgrounds would be particularly hard to model, as they would introduce significant amounts of information that would need to be reconstructed, potentially making the task unsolvable beyond a handful of scenes.

Finally, the application for viewpoint estimation systems is generally for robotic manipulation systems or 3D modeling. Current models developed in this thesis are still too limited for deployments due to their aforementioned limitations, although they present an entryway. In particular, since they already incorporate a 3D reconstruction pipeline, a natural extension could be to integrate pose estimation directly in a 3D modeling system in order to have a single end-to-end system that learns accurate poses and high-quality reconstructions, for instance by coupling it with a full-capacity NeRF.

6.3 Future works

Multiple directions can be considered as developments of the models proposed in this thesis.

As a short-term extension, integrating a background model is a possible way to extend the applicability of these approaches by removing the need for segmentation masks. As previously mentioned, multiple extensions of NeRF have been proposed for this, however,

these are typically designed for a single scene, or as a generative process where the proper pose is considered to be of little to no importance. To overcome the requirements of well-distributed poses, a possible solution is a curriculum approach, as observed in some recent pose-free NeRF models (Lin et al., 2021), starting from easy samples and progressively making the problem harder introducing more complex scenes. Still, this would require a way to quantify how complex the pose of a sample is, which is hard to do without viewpoint information. Orthogonally, trying to find correspondences within and across scenes could help make prediction more robust, especially to partially occluded or out-of-frame objects.

Recent models try to operate in even more challenging scenarios like zero-shot pose estimation (Banani et al., 2020; Goodwin et al., 2022), so it would be interesting to see how analysis by synthesis models could transfer their knowledge to new categories. While some transfers seem relatively straightforward, e.g. cars to buses, or chairs to sofas, a more general transfer could be envisioned, although there does not seem to be an obvious way to do this.

A loosely related direction is to explore NeRF conditioning mechanism. As it was originally designed for single scenes, and multi-scene conditioning only came later, there is no standard conditioning mechanism for NeRF, and a variety of approaches have been proposed. It can take the form of a simple input vector concatenated to the positional embedding, as in NeRF in the wild (Martin-Brualla et al., 2021), a mapping into middle representations of the NeRF network as in ViewNeRF and CodeNeRF (Jang and Agapito, 2021), a grid of pre-computed spatial features as in pixelNeRF (Yu et al., 2021b) or EG3D (Chan et al., 2022), or a more complex transformer-like attention mechanism as the one developed in CO3D (Reizenstein et al., 2021). Comparative analysis of these different methods could be a good indicator of the mechanisms behind NeRF conditioning and how to efficiently share information between different parts of the scene, or conversely prevent unwanted leakage between independent components.

As generative NeRF-based models have become increasingly popular (Schwarz et al., 2020; Niemeyer and Geiger, 2021; Gu et al., 2021; Kosiorek et al., 2021; Kabra et al., 2021; Chan et al., 2022), a comparison of generative against reconstruction-based pose estimation could open up interesting research directions. In theory, GAN-based models are applicable to more complex scenarios, as they only require minimizing a loose adversarial objective instead of reconstructing specific samples, however, their ubiquitous reliance on convolutional decoders to increase image quality has adverse effects on geometric consistency and therefore on pose estimation capabilities. GNeRF (Meng et al., 2021) manages to learn proper poses by mixing adversarial and reconstruction losses - although it only operates on single scenes - and does not use an additional CNN decoder. Therefore, properly quantifying the benefits of CNN decoders in generative NeRF could be a starting point for a new type of generative 3D consistent models.

Finally, longer terms goals would be integrating these systems to their proper application environments. While most pose estimation papers currently only treat pose estimation as an end task (Xiang et al., 2014; Liao et al., 2019; Mustikovela et al., 2020), recent publications show specific interest in designing pose estimation systems with specific applications in mind like robotic manipulation (Yen-Chen et al., 2022; Kupcsik et al., 2021) or digitization (Gafni et al., 2021; Liu et al., 2021).

Bibliography

Ahmadyan, A., Zhang, L., Ablavatski, A., Wei, J., and Grundmann, M. (2021). Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Computer Vision and Pattern Recognition*.

Araslanov, N. and Roth, S. (2020). Single-stage semantic segmentation from image labels. In *Computer Vision and Pattern Recognition*, pages 4253–4262.

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations (ICLR)*.

Banani, M. E., Corso, J. J., and Fouhey, D. F. (2020). Novel object viewpoint estimation through reconstruction alignment. In *Computer Vision and Pattern Recognition*, pages 3113–3122.

Barron, J. T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., and Srinivasan, P. P. (2021). Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *International Conference on Computer Vision*, pages 5855–5864.

Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194.

Boomsma, W. and Frellsen, J. (2017). Spherical convolutions and their application in

molecular modelling. *Advances in neural information processing systems*, 30.

Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., and Rother, C. (2014). Learning 6d object pose estimation using 3d object coordinates. In *European Conference on Computer Vision*, pages 536–551.

Brock, A., Donahue, J., and Simonyan, K. (2019). Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299.

Chan, E. R., Lin, C. Z., Chan, M. A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L. J., Tremblay, J., Khamis, S., et al. (2022). Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chen, J., Yin, Y., Birdal, T., Chen, B., Guibas, L. J., and Wang, H. (2022). Projective manifold gradient layer for deep rotation regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6646–6655.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017b). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, X., Dong, Z., Song, J., Geiger, A., and Hilliges, O. (2020). Category level object pose estimation via neural analysis-by-synthesis. In *European Conference on Computer Vision*.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2172–2180.

Chen, Z. and Zhang, H. (2019). Learning implicit fields for generative shape modeling. In *Computer Vision and Pattern Recognition*, pages 5939–5948.

Chng, S.-F., Ramasinghe, S., Sherrah, J., and Lucey, S. (2022). Garf: Gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *arXiv e-prints*, pages arXiv–2204.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. (2018). Spherical cnns. In *International Conference on Learning Representations (ICLR)*.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. In *European conference on computer vision*, pages 484–498. Springer.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee.

Defferrard, M., Milani, M., Gusset, F., and Perraudin, N. (2020). Deepsphere: a graph-based spherical cnn. *arXiv preprint arXiv:2012.15000*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Egger, B., Smith, W. A., Tewari, A., Wuhrer, S., Zollhoefer, M., Beeler, T., Bernard, F., Bolkart, T., Kortylewski, A., Romdhani, S., et al. (2020). 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38.

Esteves, C., Allen-Blanchette, C., Makadia, A., and Daniilidis, K. (2017). 3d object classification and retrieval with spherical cnns. *arXiv preprint arXiv:1711.06721*.

Esteves, C., Sud, A., Luo, Z., Daniilidis, K., and Makadia, A. (2019). Cross-domain 3d equivariant image embeddings. In *International Conference on Machine Learning (ICML)*.

Falorsi, L., de Haan, P., Davidson, T. R., De Cao, N., Weiler, M., Forré, P., and Cohen, T. S. (2018). Explorations in homeomorphic variational auto-encoding. *arXiv preprint arXiv:1807.04689*.

Fan, H., Su, H., and Guibas, L. J. (2017). A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613.

Gadelha, M., Maji, S., and Wang, R. (2017). 3d shape induction from 2d views of multiple objects. In *International Conference on 3D Vision (3DV)*, pages 402–411.

Gafni, G., Thies, J., Zollhofer, M., and Nießner, M. (2021). Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8649–8658.

Georgakis, G., Reza, M. A., Mousavian, A., Le, P.-H., and Košecká, J. (2016). Multiview

rgb-d dataset for object instance detection. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 426–434. IEEE.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Goodwin, W., Vaze, S., Havoutis, I., and Posner, I. (2022). Zero-shot category-level object pose estimation. *arXiv preprint arXiv:2204.03635*.

Gowda, S. N., Rohrbach, M., and Sevilla-Lara, L. (2021). Smart frame selection for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1451–1459.

Grabner, A., Roth, P. M., and Lepetit, V. (2018). 3d pose estimation and 3d model retrieval for objects in the wild. In *Computer Vision and Pattern Recognition*, pages 3022–3031.

Gu, J., Liu, L., Wang, P., and Theobalt, C. (2021). Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. In *ICLR*.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *International Conference on Computer Vision*, pages 2961–2969.

Henzler, P., Mitra, N. J., and Ritschel, T. (2019). Escaping plato's cave: 3d shape from adversarial rendering. In *International Conference on Computer Vision*, pages 9984–9993.

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

Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., and Navab, N. (2012). Model based training, detection and pose estimation of texture–less 3d objects in heavily cluttered scenes. In *Asian Conference on Computer Vision (ACCV)*, pages 548–562.

Hodan, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., and Zabulis, X. (2017). T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE.

Huang, X. and Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In *International Conference on Computer Vision*, pages 1501–1510.

Huttenlocher, D. P. and Ullman, S. (1987). Object recognition using alignment. In *Proceedings of the DARPA Image Understanding Workshop*, pages 370–380.

Huynh, D. Q. (2009). Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164.

Insafutdinov, E. and Dosovitskiy, A. (2018). Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2802–2812.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2017–2025.

Jakab, T., Gupta, A., Bilen, H., and Vedaldi, A. (2018). Unsupervised learning of object landmarks through conditional image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4016–4027.

Jakab, T., Tucker, R., Makadia, A., Wu, J., Snavely, N., and Kanazawa, A. (2021). Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12783–12792.

Jang, W. and Agapito, L. (2021). Codenerf: Disentangled neural radiance fields for object categories. In *International Conference on Computer Vision*, pages 12949–12958.

Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., and Park, J. (2021). Self-calibrating neural radiance fields. In *International Conference on Computer Vision*, pages 5846–5854.

Jiang, C., Huang, J., Kashinath, K., Marcus, P., Niessner, M., et al. (2019). Spherical cnns on unstructured grids. *arXiv preprint arXiv:1901.02039*.

Jimenez Rezende, D., Eslami, S., Mohamed, S., Battaglia, P., Jaderberg, M., and Heess, N. (2016). Unsupervised learning of 3d structure from images. *Advances in neural information processing systems*, 29.

Johnson, A. E. and Hebert, M. (1998). Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16(9-10):635–651.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *International Conference on Computer Vision (ECCV)*, pages 694–711.

Joung, S., Kim, S., Kim, H., Kim, M., Kim, I.-J., Cho, J., and Sohn, K. (2020). Cylindrical convolutional networks for joint object detection and viewpoint estimation. In *Computer Vision and Pattern Recognition*, pages 14163–14172.

Kabra, R., Zoran, D., Erdogan, G., Matthey, L., Creswell, A., Botvinick, M., Lerchner, A., and Burgess, C. (2021). Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. *Advances in Neural Information Processing Systems*, 34:20146–20159.

Kajiya, J. T. and Von Herzen, B. P. (1984). Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174.

Kanazawa, A., Jacobs, D. W., and Chandraker, M. (2016). Warpnet: Weakly supervised

matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261.

Kanazawa, A., Tulsiani, S., Efros, A. A., and Malik, J. (2018). Learning category-specific mesh reconstruction from image collections. In *European Conference on Computer Vision*, pages 371–386.

Kanezaki, A., Matsushita, Y., and Nishida, Y. (2018). Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. (2021). Alias-free generative adversarial networks. In *Proc. NeurIPS*.

Kato, H., Ushiku, Y., and Harada, T. (2018). Neural 3d mesh renderer. In *Computer Vision and Pattern Recognition*, pages 3907–3916.

Kehl, W., Manhardt, F., Tombari, F., Ilic, S., and Navab, N. (2017). Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *International Conference on Computer Vision*, pages 1521–1529.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kosiorek, A. R., Strathmann, H., Zoran, D., Moreno, P., Schneider, R., Mokrá, S., and Rezende, D. J. (2021). Nerf-vae: A geometry aware 3d scene generative model. In *International Conference on Machine Learning*, pages 5742–5752. PMLR.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Kulkarni, T. D., Whitney, W. F., Kohli, P., and Tenenbaum, J. (2015). Deep convolu-

tional inverse graphics network. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2539–2547.

Kundu, A., Li, Y., and Rehg, J. M. (2018). 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3559–3568.

Kupcsik, A. G., Spies, M., Klein, A., Todescato, M., Waniek, N., Schillinger, P., and Bürger, M. (2021). Supervised training of dense object nets using optimal descriptors for industrial robotic applications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6093–6100.

Laine, S. and Aila, T. (2017). Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Lee, D.-H. et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Li, Y., Mao, H., Girshick, R., and He, K. (2022). Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*.

Liao, S., Gavves, E., and Snoek, C. G. (2019). Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *Computer Vision and Pattern Recognition*, pages 9759–9767.

Lin, C.-H., Kong, C., and Lucey, S. (2018). Learning efficient point cloud generation for dense 3d object reconstruction. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lin, C.-H., Ma, W.-C., Torralba, A., and Lucey, S. (2021). Barf: Bundle-adjusting neural radiance fields. In *International Conference on Computer Vision*, pages 5741–5751.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

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

Liu, L., Gu, J., Zaw Lin, K., Chua, T.-S., and Theobalt, C. (2020). Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663.

Liu, L., Habermann, M., Rudnev, V., Sarkar, K., Gu, J., and Theobalt, C. (2021). Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics (TOG)*, 40(6):1–16.

Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., and Sheikh, Y. (2019). Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Longuet-Higgins, H. C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135.

Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial intelligence*, 31(3):355–395.

Mahendran, S., Ali, H., and Vidal, R. (2017). 3d pose regression using convolutional

neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2174–2182.

Mariotti, O. and Bilen, H. (2020). Semi-supervised viewpoint estimation with geometry-aware conditional generation. In *European Conference on Computer Vision*, pages 631–647. Springer.

Mariotti, O., Mac Aodha, O., and Bilen, H. (2021). Viewnet: Unsupervised viewpoint estimation from conditional generation. In *International Conference on Computer Vision*, pages 10418–10428.

Mariotti, O., Mac Aodha, O., and Bilen, H. (2022). Viewnerf: Unsupervised viewpoint estimation using category-level neural radiance fields. *arXiv preprint arXiv:2212.00436*.

Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., and Duckworth, D. (2021). Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Computer Vision and Pattern Recognition*, pages 7210–7219.

Meng, Q., Chen, A., Luo, H., Wu, M., Su, H., Xu, L., He, X., and Yu, J. (2021). Gnerf: Gan-based neural radiance field without posed camera. In *International Conference on Computer Vision*, pages 6351–6361.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *Computer Vision and Pattern Recognition*, pages 4460–4470.

Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. (2016). Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer.

Müller, T., Evans, A., Schied, C., and Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*.

Mustikovela, S. K., Jampani, V., Mello, S. D., Liu, S., Iqbal, U., Rother, C., and Kautz, J. (2020). Self-supervised viewpoint learning from image collections. In *Computer Vision and Pattern Recognition*.

Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer.

Nguyen, V. N., Hu, Y., Xiao, Y., Salzmann, M., and Lepetit, V. (2022). Templates for 3d object pose estimation revisited: generalization to new objects and robustness to occlusions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6771–6780.

Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., and Yang, Y.-L. (2019). Hologan: Unsupervised learning of 3d representations from natural images. In *International Conference on Computer Vision*, pages 7588–7597.

Nguyen-Phuoc, T. H., Li, C., Balaban, S., and Yang, Y. (2018). Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. *Advances in neural information processing systems*, 31.

Niemeyer, M. and Geiger, A. (2021). Giraffe: Representing scenes as compositional generative neural feature fields. In *Computer Vision and Pattern Recognition*, pages 11453–11464.

Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. (2020). Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Computer Vision and Pattern Recognition*, pages 3504–3515.

Novotny, D., Larlus, D., and Vedaldi, A. (2017). Learning 3d object categories by looking around them. In *International Conference on Computer Vision*, pages 5218–5227.

Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., and Geiger, A. (2019). Texture fields: Learning texture representations in function space. In *International Conference on Computer Vision*, pages 4531–4540.

Olszewski, K., Tulyakov, S., Woodford, O., Li, H., and Luo, L. (2019). Transformable bottleneck networks. In *International Conference on Computer Vision*, pages 7648–7657.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In *Computer Vision and Pattern Recognition*, pages 165–174.

Park, K., Sinha, U., Barron, J. T., Bouaziz, S., Goldman, D. B., Seitz, S. M., and Martin-Brualla, R. (2021). Nerfies: Deformable neural radiance fields. In *International Conference on Computer Vision*, pages 5865–5874.

Pavlakos, G., Zhou, X., Chan, A., Derpanis, K. G., and Daniilidis, K. (2017). 6-dof object pose from semantic keypoints. In *International Conference on Robotics and Automation (ICRA)*, pages 2011–2018.

Pham, H., Dai, Z., Xie, Q., and Le, Q. V. (2021). Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568.

Pope, A. R. and Lowe, D. G. (2000). Probabilistic models of appearance for 3-d object recognition. *International Journal of Computer Vision*, 40(2):149–167.

Prokudin, S., Gehler, P., and Nowozin, S. (2018). Deep directional statistics: Pose estimation with uncertainty quantification. In *European Conference on Computer Vision*, pages 534–551.

Rad, M. and Lepetit, V. (2017). 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*, pages 3828–3836.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. (2019). On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR.

Ranzato, M. and Szummer, M. (2008). Semi-supervised learning of compact document representations with deep networks. In *Proceedings of the 25th international conference on Machine learning*, pages 792–799.

Rasmus, A., Berglund, M., Honkala, M., Valpola, H., and Raiko, T. (2015). Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28.

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

Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., and Novotny, D. (2021). Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Rhodin, H., Salzmann, M., and Fua, P. (2018). Unsupervised geometry-aware representation for 3d human pose estimation. In *European Conference on Computer Vision*, pages 750–767.

Riegler, G. and Koltun, V. (2020). Free view synthesis. In *European Conference on Computer Vision*, pages 623–640. Springer.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Schonberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. In *Computer Vision and Pattern Recognition*, pages 4104–4113.

Schwarz, K., Liao, Y., Niemeyer, M., and Geiger, A. (2020). Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166.

Sedaghat, N. and Brox, T. (2015). Unsupervised generation of a viewpoint annotated car dataset from videos. In *International Conference on Computer Vision*, pages 1314–1322.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473.

Sitzmann, V., Rezchikov, S., Freeman, B., Tenenbaum, J., and Durand, F. (2021). Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34.

Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., and Zollhofer, M. (2019a). Deepvoxels: Learning persistent 3d feature embeddings. In *Computer Vision and Pattern Recognition*.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. (2019b). Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015a). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953.

Su, H., Qi, C. R., Li, Y., and Guibas, L. J. (2015b). Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694.

Sundermeyer, M., Marton, Z.-C., Durner, M., Brucker, M., and Triebel, R. (2018). Implicit 3d orientation learning for 6d object detection from rgb images. In *European Conference on Computer Vision*, pages 699–715.

Suwajanakorn, S., Snavely, N., Tompson, J. J., and Norouzi, M. (2018). Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2059–2070.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Tan, V., Budvytis, I., and Cipolla, R. (2018). Indirect deep structured learning for 3d human body shape and pose prediction. In *British Machine Vision Conference (BMVC)*.

Tarvainen, A. and Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204.

Tekin, B., Sinha, S. N., and Fua, P. (2018). Real-time seamless single shot 6d object pose prediction. In *Computer Vision and Pattern Recognition*, pages 292–301.

Thewlis, J., Bilen, H., and Vedaldi, A. (2017a). Unsupervised learning of object frames by dense equivariant image labelling. In *Advances in Neural Information Processing Systems*, pages 844–855.

Thewlis, J., Bilen, H., and Vedaldi, A. (2017b). Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5916–5925.

Thewlis, J., Bilen, H., and Vedaldi, A. (2018). Modelling and unsupervised learning of symmetric deformable object categories. In *Advances in Neural Information Processing Systems*, pages 8178–8189.

Tompson, J. J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. *Advances in neural information processing systems*, 27.

Tseng, H.-Y., De Mello, S., Tremblay, J., Liu, S., Birchfield, S., Yang, M.-H., and Kautz, J. (2019). Few-shot viewpoint estimation. *arXiv preprint arXiv:1905.04957*.

Tulsiani, S., Efros, A. A., and Malik, J. (2018). Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Computer Vision and Pattern Recognition*, pages 2897–2905.

Tulsiani, S. and Malik, J. (2015). Viewpoints and keypoints. In *Computer Vision and Pattern Recognition*, pages 1510–1519.

Tulsiani, S., Zhou, T., Efros, A. A., and Malik, J. (2017). Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Computer Vision and Pattern Recognition*, pages 2626–2634.

Wang, A., Mei, S., Yuille, A. L., and Kortylewski, A. (2021a). Neural view synthesis and matching for semi-supervised few-shot learning of 3d pose. *Advances in Neural Information Processing Systems*, 34.

Wang, Z., Wu, S., Xie, W., Chen, M., and Prisacariu, V. A. (2021b). Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.

Weiss, I. and Ray, M. (2001). Model-based recognition of 3d objects from single images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):116–128.

Weston, J., Ratle, F., and Collobert, R. (2008). Deep learning via semi-supervised embedding. In *Proceedings of the 25th international conference on Machine learning*, pages 1168–1175.

Whitney, W. F., Chang, M., Kulkarni, T., and Tenenbaum, J. B. (2016). Understanding visual concepts with continuation learning. *arXiv preprint arXiv:1602.06822*.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. (2017). Interpretable transformations with encoder-decoder networks. In *International Conference on Computer Vision*, pages 5726–5735.

Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. (2016). Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920.

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

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

Xiao, Y. and Marlet, R. (2020). Few-shot object detection and viewpoint estimation

for objects in the wild. In *European conference on computer vision*, pages 192–210. Springer.

Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. (2016). Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1696–1704.

Yang, G., Cui, Y., Belongie, S., and Hariharan, B. (2018). Learning single-view 3d reconstruction with limited pose supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 86–101.

Yang, J., Liu, Q., and Zhang, K. (2017). Stacked hourglass network for robust facial landmark localisation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 79–87.

Yen-Chen, L., Florence, P., Barron, J. T., Lin, T.-Y., Rodriguez, A., and Isola, P. (2022). NeRF-Supervision: Learning dense object descriptors from neural radiance fields. In *IEEE Conference on Robotics and Automation (ICRA)*.

Yen-Chen, L., Florence, P., Barron, J. T., Rodriguez, A., Isola, P., and Lin, T.-Y. (2021). iNeRF: Inverting neural radiance fields for pose estimation. In *IROS*.

Yu, A., Fridovich-Keil, S., Tancik, M., Chen, Q., Recht, B., and Kanazawa, A. (2022). Plenoxels: Radiance fields without neural networks. In *Computer Vision and Pattern Recognition*.

Yu, A., Li, R., Tancik, M., Li, H., Ng, R., and Kanazawa, A. (2021a). Plenoctrees for real-time rendering of neural radiance fields. In *International Conference on Computer Vision*, pages 5752–5761.

Yu, A., Ye, V., Tancik, M., and Kanazawa, A. (2021b). pixelnerf: Neural radiance fields from one or few images. In *Computer Vision and Pattern Recognition*, pages 4578–4587.

Zhang, K., Riegler, G., Snavely, N., and Koltun, V. (2020). Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*.

Zhou, X., Karpur, A., Luo, L., and Huang, Q. (2018). Starmap for category-agnostic keypoint and viewpoint estimation. In *European Conference on Computer Vision*, pages 318–334.

Zhou, Y., Barnes, C., Lu, J., Yang, J., and Li, H. (2019). On the continuity of rotation representations in neural networks. In *Computer Vision and Pattern Recognition*, pages 5745–5753.

Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2879–2886. IEEE.