HyperPosePDF
Hypernetworks Predicting the Probability Distribution on SO(3)

Timon Höfer, Benjamin Kiefer, Martin Messmer, Andreas Zell
University of Tübingen
Wilhelm-Schickard-Institute for Computer Science, Sand 1, 72076 Tübingen
timon.hoefer@uni-tuebingen.de

Abstract

Pose estimation of objects in images is an essential problem in virtual and augmented reality and robotics. Traditional solutions use depth cameras, which can be expensive, and working solutions require long processing times. This work focuses on the more difficult task when only RGB information is available. To this end, we predict not only the pose of an object but the complete probability density function (pdf) on the rotation manifold. This is the most general way to approach the pose estimation problem and is particularly useful in analysing object symmetries. In this work, we leverage implicit neural representations for the task of pose estimation and show that hypernetworks can be used to predict the rotational pdf. Furthermore, we analyse the Fourier embedding on SO(3) and evaluate the effectiveness of an initial Fourier embedding that proved successful. Our HyperPosePDF outperforms the current SOTA approaches on the SYMSOL dataset.

1. Introduction

Pose estimation has gained an increasing interest in the last years. In many robotic applications, such as object grasping, tracking and occlusion handling, the robotic perception should be able to accurately estimate 3D poses to perform a valid grasp. Traditional approaches assume present depth information and estimate the pose by relying on local invariant features [1, 37] or template-matching [24]. These algorithms rely on expensive evaluations of multiple pose hypotheses rendering them inefficient. Furthermore, missing textures on many objects hamper their performance.

RGB-based methods, which do not require expensive depth sensors, have outperformed depth methods in terms of speed and accuracy using convolutional neural networks in the BOP challenge [26]. In this work, we will continue to focus on RGB-based methods.

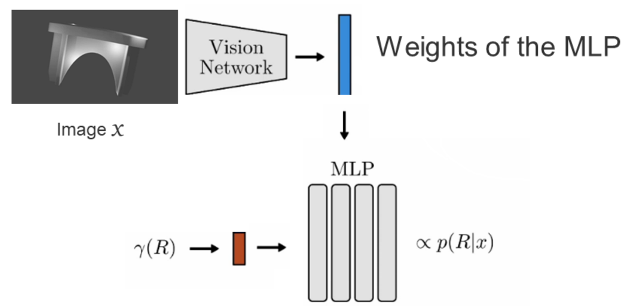


Figure 1: Overview of the network Architecture. An image x is fed through a vision network that predicts a feature vector. This feature vector is then used as the weights of an MLP. The MLP acts on the rotation manifold and takes as input the Fourier embedded rotation $\gamma(R)$ and outputs a probability.

One major problem in pose estimation are symmetries that arise in industrial settings or in our daily life (for example, a ball without texture or a cup, whenever the cup handle is not visible). Those challenges are tackled in different ways; for example, for the TLESS [25] and YCB-video [76] datasets, additional symmetry information is provided by [26] and available during training and inference. Still, classical pose estimators are trained to output a single pose and do not consist of any symmetry information handling. Hence, we want to focus on methods that can handle symmetries and quasi-symmetries.

In this work, we follow a general approach - predicting a probability distribution on the rotation manifold $p : \mathrm{SO}(3) \to \mathbb{R}^+$. Once obtained, present symmetries can be easily read from the probability distribution peaks while still allowing for single pose predictions by simply taking the maximum peak as the respective rotation. One approach can be to use multinomial mixture distributions of the gaussian distribution [58, 18, 11]. This would introduce the need to select a number of normal distributions, which differs from object to object, as different amounts of symmetries exist for objects like a pyramid or a cup. Additionally, with an object like a cup, the number of symmetries is dependent if the handle of the cup is visible or not, which is problematic for the mixture distribution to handle.

A more general approach is given by [46], where they use a multilayer perceptron (MLP) to represent the probability density function. In detail, they combine the image feature vector with the rotation feature vector and feed it jointly through the MLP to yield a probability of that rotation to be the actual rotation. This removes the need of a manual investigation of the symmetry count, as it learns it implicitly. In this way, [46] can show remarkable performance.

This opens up a connection to the field of Implicit Neural Representations (INRs) that recently has received significant attention. INRs use neural networks to map the input domain of the signal (e.g., coordinates of a specific pixel in the image) to a representation of color, occupancy or density at the input location. INRs have boosted the performance on texture synthesis [23, 49], shape representation [38, 39] and derivation of shapes from images [8, 10, 17, 16, 30, 43, 51].

To close the bridge to INRs, we want the rotation to be the sole input to the MLP; hence we propose using a hypernetwork. To do so, we define a vision network that receives the image as input and outputs the weights of the MLP, acting as the implicit neural representation. The usage of hypernetworks allows learning a prior over the space of parameterized functions and thus can be much faster to finetune, compared to models trained from scratch. Additionaly, our hypernetworks are trained end-to-end with backpropagation and therefore are efficient and scalable. Furthermore, it enables a knowledge transfer from INR theory to our problem domain. Specifically, we aim to utilize Fourier encodings in our settings, which have drastically boosted the performance of INR applications [71, 4]. In summary, we present the following contributions:

- HyperPosePDF - a hypernetwork to predict a non-parametric probability distribution on $\mathrm{SO}(3)$ given an image, that not only can do pose estimation but also inherently consists of all the symmetry information, thus allowing for uncertainty quantification.
- A transfer from the Fourier encoding used in traditional INR applications to the usage in a pose estimation scenario.

2. Related work

2.1. Hypernetworks

Hypernetworks have become very common in deep learning and date back as far as the beginning of the 1990s in the context of meta-learning and self-referential [64].

Several works explored the use of hypernetworks for RNNs [63, 19, 70, 22, 3, 20], CNNs [13, 32, 29, 5, 54, 31, 61] and Reinforcement Learning [14, 28, 60]. Architecture search algorithms incorporated forms of hypernetworks early on [68, 33, 6, 77]. Furthermore, the concept of self-attention can be viewed as a form of adaptive layers [62].

Finally, hypernetworks have also been introduced to the field of Implicit Neural Representations [65]. However, the use of hypernetworks has mainly been explored for 2D and 3D image and scene generation [42, 36, 66, 67, 74].

Likewise, we want to apply a hypernetwork to implicit neural representations associated with the task of predicting the probability distribution on the rotation manifold.

2.2. Implicit Neural Representations

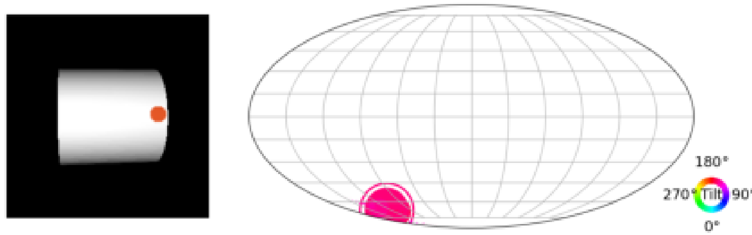
Inspired by its recent success, Implicit Neural Representations have recently received much attention. Especially in 3D computer vision works based on INRs achieved state-of-the-art results [2, 21, 30, 52, 7, 65]. Further impressive results are achieved across different domains, e.g., from 2D supervision [66, 48, 44] and 3D supervision [59, 50] to dynamic scenes [47], which use space-time INRs for representation.

One crucial part of the performance for INRs is the usage of an initial Fourier embedding. The lack of accuracy for fine details was tackled by the introduction of the well-known positional encoding [44]. With the finding that the main contributor of the Fourier embedding is its size and standard deviation [72, 4], other embeddings have been introduced; in its most extreme form, the random sampling from a gaussian distribution [72], which can outperform traditional embeddings if the standard deviation is chosen accurately.

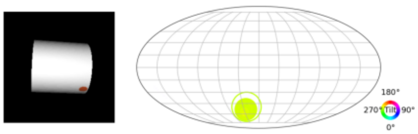
Recently, the theory of INRs inspired tackling the pose estimation problem with its specific focus on symmetries by learning the distribution on SO(3) and influencing the design choice of the network architecture [46].

2.3. Pose Estimation

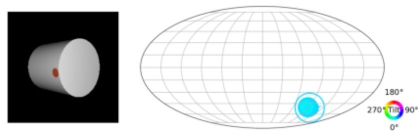
In recent years, pose estimation methods based on RGB images using convolutional neural networks [34, 69, 27] have outperformed the classic approaches [26] while also reaching higher fps. As symmetries occur plentifully in industrial or everyday objects, it is interesting and essential to conduct further research on their occurrence. If object symmetries are known during training, it is possible to group equivalent rotations to a single one, allowing training to proceed as in classical single-valued regression [56]. In [9], manually labeled symmetries of 3D poses are needed to learn the embedding and classification of the symmetry order together.



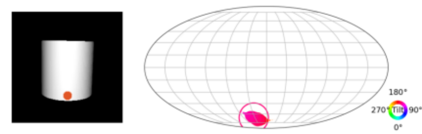
(a) If the red marker is visible, the rotation of the *cylinder* is unique.



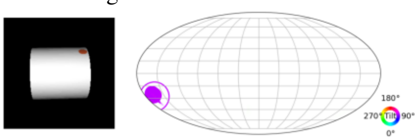
(b) Rotation around the x-axis with the marker being moved to the bottom.



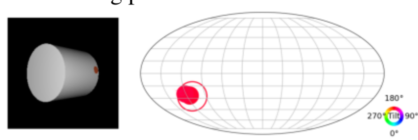
(c) Rotation around the y-axis with the marker being pushed to the front.



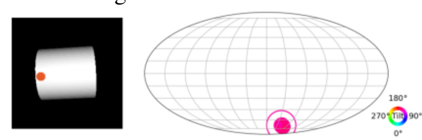
(d) Rotation around the z-axis with the marker being moved to the bottom.



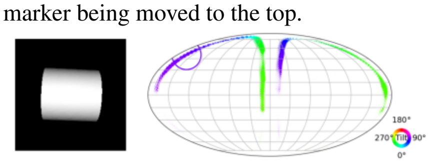
(e) Rotation around the x-axis with the marker being moved to the top.



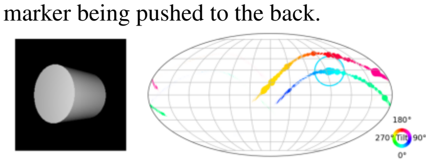
(f) Rotation around the y-axis with the marker being pushed to the back.



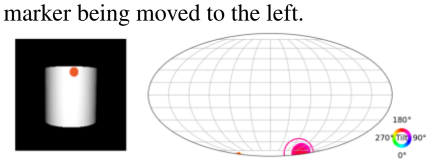
(g) Rotation around the x-axis with the marker being moved to the left.



(h) As the marker is not visible, a continuous symmetry can be seen. Only half of the symmetry axis of a normal *cylinder* is displayed as the model learned to nullify the subspace of rotations for which the marker would be visible.



(i) The marker is not visible, therefore our model continuous symmetry axis with a gap in between representing the area, where the marker would be visible.



(j) The movement around the z-axis has the effect of maintaining the tilt colour. As in this scenario the marker was always visible, only one unique rotation is present.

Figure 2: Visualization of results on the cylO object from the SYMSOL II dataset. Elements of SO(3) with a positive probability are visualized as points on the grid. Intuitively, we can consider each point on the grid as the direction of a canonical z-axis, and the color indicates the angle of inclination axis around this axis. Note that the hollow circle indicates the ground truth pose, while the filled area depicts the predicted poses. Note that in the case of a missing red dot, the ground truth pose may be ambiguous and we plot only one possible ground truth pose. The visualization tool was introduced by [46].

On the contrary, [69, 27] make pose or symmetry supervision unnecessary by using an augmented autoencoder to isolate pose information. During inference, they receive a latent representation, compare it to a fully covered sample in a codebook of saved latent representations of rotations and take the closest one.

As symmetries are not the only source of pose uncertainty, it is interesting to utilize a more flexible representation. Recent works focused on a statistical approach by considering parametric probability distributions. [53, 11, 18] regressed the parameters of a von Mises distribution over Euler-angles and [45] utilize Matrix Fisher distributions on SO(3). To this end, [58, 18, 11] propose using multimodal mixture distributions. One challenge when training the mixtures is avoiding mode collapse, for which a winner-takes-it-all strategy can be used [11]. An alternative to the mixture models is to predict multiple pose hypotheses directly [41], but this does not share any of the benefits of a probabilistic representation.

A more general representation of the distribution is pro-

	SYMSOL I (log likelihood ↑)					
	cone	cyl.	tet.	cube	ico.	avg.
Deng et al. [11]	0.16	-0.95	0.27	-4.44	-2.45	1.48
Gilitschenski et al. [18]	3.84	0.88	-2.29	-2.29	-2.29	-0.43
Prokudin et al. [58]	-1.87	-3.34	-1.28	-1.86	-0.50	-2.39
Murphy et al. [46]	4.45	4.26	5.70	4.81	1.28	4.10
HyperPosePDF (Ours)	**5.74**	**4.73**	**7.04**	**6.77**	**5.10**	**5.78**

Table 1: A model was jointly trained for all of the SYMSOL I classes. We compare our results against multimodal mixture models [11, 18, 58] and Implicit-PDF [46] which we all outperform by a significant amount in the log likelihood metric. A value of -2.29 represents the minimal information of a uniform distribution on SO(3).

	SYMSOL I (Spread ↓)				
	cone	cyl.	tet.	cube	ico.
Deng et al. [11]	10.1	15.2	16.7	40.7	28.5
Murphy et al. [46]	1.4	1.4	4.6	4.0	8.4
HyperPosePDF (Ours)	**0.55**	**0.48**	**3.27**	**2.18**	**3.24**

Table 2: Similiar to Tab. 1 we train a joint model for all objects in the SYMSOL I dataset and compare it to the method of [11] and Implicit-PDF [46]. For the cone and cylinder, the spread of the probability prediction away from the rotational continuous symmetry has a value of less than one degree.

posed by [46], where they model the probability density function with a multilayer perceptron whose architecture is inspired by the field of INRs. Their works provide the challenging SYMSOL and SYMSOL II datasets focused on symmetries and can show superior performance to the above-introduced mixture models.

In our work we are going to make use of a hypernetwork to predict the weights of an implicit neural representation. This implicit neural representation is associated with the task of representing a probability distribution on SO(3). We then aim to fully utilize the theoretical findings on Fourier embeddings for pose estimation, which have been found to be crucial for the performance of INRS. We will introduce our approach in the following.

3. Method

Given an image x, our goal is to predict a probability density function

$$p(\cdot|x) : \mathrm{SO}(3) \to [0,1] \tag{1}$$

that incorporates not only a single rotation but the general information on the distribution of the rotation of an object in a given image. This is especially helpful in finding symmetry patterns of objects.

We give a general overview of our approach in Figure 1. The input image is first fed through a vision network to output a feature vector. This feature vector is then used as the weights of an MLP. The MLP then represents the probability density function on $\mathrm{SO}(3)$ by taking a Fourier-mapped rotation $\mathrm{SO}(3) \ni R \mapsto \gamma(R)$ as input, and outputting the corresponding probability $p(R|x) \in [0,1]$. With this formulation, it is possible to make single pose predictions by taking the mode of the pdf or to predict the full distribution to observe patterns of symmetries.

3.1. Fourier Transform on the Rotation Manifold

For an integrable function of the form $f : \mathbb{R} \to \mathbb{C}$ the Fourier transform of f is defined as

$$\mathcal{F}_f(l) = \int_{\mathbb{R}} f(x)\, \mathrm{e}^{-ilx}\, dx. \tag{2}$$

The Fourier transform is usually applied to periodic, and bounded functions, i.e. of the form $f : [0, 2\pi) \to \mathbb{C}$. Instead of defining f on the range $[0, 2\pi)$, we can also find a mapping between $\alpha \in [0, 2\pi)$ and the rotation matrices $R_\alpha \in \mathrm{SO}(2)$, where α is the rotation angle. This allows us to use the Fourier transform for complex valued functions defined on the rotation group $\mathrm{SO}(2)$. This indeed suggests that the Fourier transform can be generalized to work with various other groups, specifically $\mathrm{SO}(3)$.

In fact, this is possible by introducing the Wigner-D matrices, which are from a technical point of view the irreducible representations of the rotation group SO(3) [55]. Leveraging this observation, it is possible to define the Fourier transform for a function

$$f\colon\ \mathrm{SO}(3) \longrightarrow \mathbb{R}. \tag{3}$$

	SYMSOL I (log likelihood ↑)					
	cone	cyl.	tet.	cube	ico.	avg.
Positional Encoding	5.74	4.73	7.04	6.77	5.10	5.78
Gaussian encoding	5.78	5.05	7.16	6.80	5.48	6.05
Siren encoding	5.66	4.71	8.06	7.34	4.01	5.96

Table 3: We evaluate the effect of an initial Fourier embedding being applied to our network. In this table we compare the effect of positional encoding [44] vs. Gaussian encoding [72] vs. a learnable sinusoidal layer [65]. While the positional encoding is the most spread embedding, it is possible to increase the performance by changing to a Gaussian embedding or a learnable sinusoidal layer. For the experiments reported in the other tables, we use a positional encoding.

	SYMSOL II (log likelihood ↑)			
	sphX	cylO	tetX	avg.
Deng et al. [11]	1.12	2.99	3.61	2.57
Gilitschenski et al. [18]	3.32	4.88	2.90	3.70
Prokudin et al. [58]	4.19	4.16	1.48	0.48
Murphy et al. [46]	7.30	6.91	8.49	7.57
HyperPosePDF (Ours)	7.73	7.12	8.53	7.72

Table 4: For this experiment, we trained a model for each object of the SYMSOL II dataset separately and compare our results against multimodal mixture models [11, 18, 58] and Implicit-PDF [46]. We are able to achieve better results than our competitors on all objects. These experiments were especially challenging due to the differing numbers of symmetries that are dependant on the visibility of the markers on the objects.

By using the Wigner-D functions $D_l^{m,n}$, which are an orthogonal basis for the rotation group $\mathrm{SO}(3)$, the Fourier transform is given as

$$f = \sum_{l=1}^{L} \sum_{m,n=-l}^{l} f_{l,m,n} D_l^{m,n} \tag{4}$$

with the integer L denoting the degree of freedom. It is possible to rewrite this into an ordinary Fourier transform by expanding the Wigner-D function to a Fourier sum. In literature, this derivation is usually given by using the Euler angles representation $R(\alpha,\beta,\gamma)$ of the respective rotation. Following [57] it turns out that

$$f(R(\alpha,\beta,\gamma))) = \sum_{l,m,n=-L}^{L} h_l^{m,n} e^{-i\left((m,n,l)(R(\alpha,\beta,\gamma)\right)}, \tag{5}$$

where the derivation of the Fourier coefficients $h_l^{m,n}$ can be found in the supplementary material. For ease of writing, we define $\mathbf{i} := (m,n,l)$. Using Euler's formula it is easy to show that (see supplementary material)

$$f(R) = \sum_{\mathbf{i}=-L, m\geq 0}^{L} a_{\mathbf{i}} \cos(2\pi \mathbf{i} R) + b_{\mathbf{i}} \sin(2\pi \mathbf{i} R), \tag{6}$$

where

$$\begin{aligned} a_{\mathbf{0}} &= h_{\mathbf{0}}, \\ a_{\mathbf{i}} &= \begin{cases} 0 & \exists j \in \{2,3\} : i_1 = i_{j-1} = 0 \wedge i_j < 0 \\ 2\mathrm{Re}(h_{\mathbf{i}}) & \text{otherwise}, \end{cases} \\ b_{\mathbf{i}} &= \begin{cases} 0 & \exists j \in \{2,3\} : i_1 = i_{j-1} = 0 \wedge i_j < 0 \\ -2\mathrm{Im}(h_{\mathbf{i}}) & \text{otherwise}. \end{cases} \end{aligned} \tag{7}$$

The main idea is to make the coefficients $\mathbf{a}$ and $\mathbf{b}$ trainable, by letting them act as weights of a neural network on an initial Fourier embedding. In [4] it was shown that for problems of dimension > 2, as it is in our case, memory problems arise on a modern Nvidia RTX 2080Ti GPU if all coefficients are jointly approximated as the size of the embedding simply gets too large. This introduces the need of finding appropriate Fouier embeddings that do not affect the performance and memory consumption of the method. The design choices of the embedding is discussed in the next section.

3.2. Fourier embedding

Inspired by the success for INRs, we compare the following three embeddings on a flattened rotation $R \in \mathrm{SO}(3)$, which we call r in the following.

- The positional encoding is defined as: $\gamma(r) = [\dots, \cos(\pi 2^{\frac{j}{m}} r), \sin(\pi 2^{\frac{j}{m}} r), \dots]$

for $j = 0, \dots, m-1$ where $m \in \mathbb{N}$, using a log-linear spacing for each dimension [44].

- The Gaussian embedding is defined as: $\gamma(r) = [\cos(2\pi \mathbf{B} r), \sin(2\pi \mathbf{B} r)]$, where $\mathbf{B} \in \mathbb{R}^{m \times d}$ is sampled from a normal distribution $\mathcal{N}(0, \sigma^2)$, while σ is the hyperparameter to be optimized [72].

- Instead of using an initial Fourier encoding, it is also possible to use a sinusoidal network. Contrary to classical MLPs, it consists of periodic activation functions. It has been shown that an additional initial sinusoidal layer acts as a learnable Fourier embedding layer, achieving similiar or better performance [4, 65].

4. Experiments

We conduct our experiments on the Symsol I, Symsol II and Pascal3D+ datasets. While we use the common Acc30° metric for Pascal3D+, we evaluate the SYMSOL datasets using two metrics: log likelihood and spread, which we will introduce in the following.

4.1. Evaluation Metrics

We assume the ground truth labels to be samples from an underlying but unknown distribution, which contains all information about symmetries, noise and ambiguities. As the output of our model is also a distribution, it is standard to compare the two distributions using maximum likelihood. More formally: Our test set consists of images $x \in I$, where each x has annotated poses $\mathbf{R^x} = (R_1^x, \dots, R_k^x)$ for some $k \in \mathbb{N}$ and $k > 1$ if there exist symmetries. We then calculate the averaged log likelihood as follows

$$\mathrm{LL} = \frac{1}{|I|} \sum_{x \in I} \frac{1}{|\mathbf{R}^x|} \sum_{R \in \mathbf{R^x}} \log(p(R|x)).$$

Another way of comparing two distributions is to calculate the spread Spr. It assumes a set of equivalent rotation annotations to be given. It uses the geodesic distance

$$\begin{aligned} d : \mathrm{SO}(3) \times \mathrm{SO}(3) &\to \mathbb{R}_+ \\ (R_1, R_2) &\mapsto || \log R_1 R_2^T ||_F \end{aligned}$$

using the Frobenius norm $|| \cdot ||_F$. Only the closest ground truth annotation is then taken into account

$$Spr = \mathrm{E}_{R \sim p(R|x)} [\min_{R' \in \mathbf{R}^x} d(R, R')].$$

4.2. SYMSOL I

The Symsol I dataset is publicly available as part of the Tensorflow datasets. This dataset is especially interesting as it consists of 5 objects with multiple symmetries, namely: *cone*, *cylinder*, *tetrahedron*, *cube* and *icosahedron*. Here, the *tetrahedron*, *cube* and *icosahedron* have countably many symmetries, i.e. 12, 24 and 60, respectively. As the *cone* and *cylinder* both have continuous symmetries, their annotations are made discrete with an equidistant 1-degree spacing. Each RGB image is of size 224×224. The associated labels per image are its class and the ground truth rotation including all equivalent rotations.

Our implementation specifics are as follows. For our vision module we use a pretrained ResNet-50 backbone. We predict the weights of a one-layer network with a width of 256. The number of coefficients used for the positional encoding is set to 4. A learning rate of $1e-4$ is used for the first 1000 iterations, then a cosine decay is applied. Using the Adam optimizer, we evaluate our model after 200k iterations using a batch size of 16.

The model learns jointly all object classes of the SYMSOL I dataset. Table 1 shows the log likelihood results. In this metric, we can demonstrate superior results to competing methods on all objects individually and on average. This is particularly visible for the objects *cone, tetrahedron, cube* and *icosahedron*. We were able to rerun the experiments of the competing methods and receive numbers closely to their official numbers, still, we show their reported values in our table. Note that we used the positional encoding in this experiment. We can further improve the performance by switching to Gaussian or Siren encodings. Table 2 shows the spread results. We compare against reported values from [11] and [46]. The metric values are in degrees and show how well the method is able to capture the ground truths. For the cone and cylinder, the spread of the probability prediction away from the rotational continuous symmetry has a value of less than one degree. The spread experiments have only been conducted on the SYMSOL I dataset as it is the only one with full symmetry annotations. If only a single ground truth is known this metric would be misleading as it penalizes correct predictions if no corresponding annotation is available.

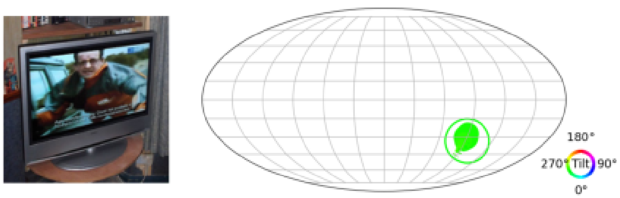
We compare the different Fourier embeddings as introduced in section 3.2. For the Gaussian embedding, we found a scale of 2 to perform best. Likewise, the performance of the sinusoidal embedding heavily depends on the chosen bias, which we found to perform best with a value of 1. Table 3 shows that, in general, an embedding is helpful, and with accurate parameters, it is possible for the Gaussian and Siren embedding to outperform the positional encoding.

4.3. SYMSOL II

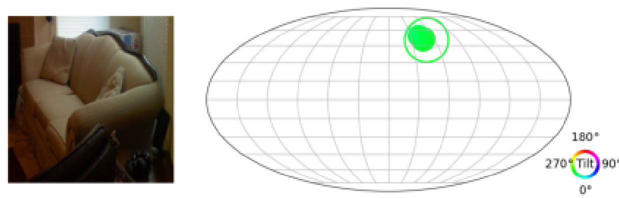
The Symsol II dataset is also publicly available as part of the Tensorflow datasets. This dataset consists of three objects: a *tetrahedron* (*tetX*) with a marked red area , a *cylinder* (*cylO*) with a marked off-center point and a *sphere* (*sphX*) with an X and a marked point. Depending on the visibility, these markings affect the number of symmetries sig-

	PASCAL3D+ (Acc30° ↑)						PASCAL3D+ (Median ↓)					
	bottle	bus	table	sofa	tv	avg	bottle	bus	table	sofa	tv	avg
Liao et al. [35]	0.93	0.95	0.61	0.95	0.82	0.852	10.3	4.8	12.0	12.3	14.3	10.74
Mohlin et al. [45]	0.94	0.95	0.62	0.85	0.84	0.840	7.8	3.3	12.5	13.8	11.7	9.82
Prokudin et al. [58]	0.96	0.93	0.76	0.90	0.91	0.892	5.4	2.9	12.6	9.1	12.0	8.4
Tulsianiet al. [73]	0.93	0.98	0.62	0.82	0.80	0.830	12.9	5.8	15.2	13.7	15.4	12.6
Mahendran et al. [40]	0.96	0.97	0.67	0.97	0.88	0.890	7.0	3.1	11.3	10.2	11.7	8.66
Murphy et al. [46]	0.93	0.95	0.78	0.88	0.86	0.880	8.8	3.4	7.3	9.5	12.3	8.26
HyperPosePDF (Ours)	0.83	0.92	0.97	0.89	0.88	0.898	11.7	3.9	4.2	5.8	6.5	6.42

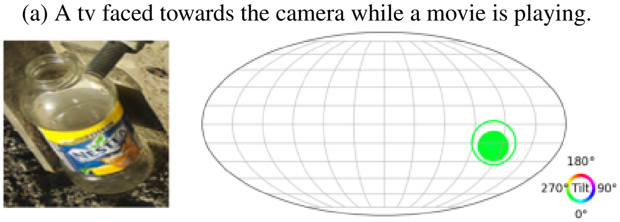
Table 5: Results on objects from the Pascal3D+ dataset. A single model was jointly trained on all classes. We compare our results in the Acc30° and the median in degrees. We are able to achieve similar or slightly better results than the competing methods.



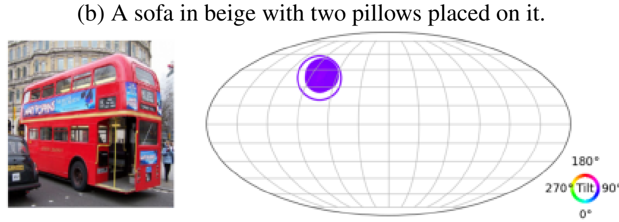
(a) A tv faced towards the camera while a movie is playing.



(b) A sofa in beige with two pillows placed on it.



(c) A bottle with yellow plastic wrapped around it.



(d) A red bus with the front faced to the right.

Figure 3: Results on the Pascal3D+ dataset. As all objects in these images are standing upright and are faced towards us, the rotations are closely related. With the presence of texture, symmetries are not existent and hence, we predict only a single rotation for the objects.

nificantly. For example, the sphere without visible markings would have all orientations with the marks on the back possible, but if both markings are visible the orientation would be unique.

We took the same implementation specifics as for the SYMSOL I dataset. Following [46], we trained a network for each object separately. Table 4 shows that we are able to achieve promising results on this challenging dataset. In particular, we show in the experiments that our method is able to represent distributions that cannot be well approximated by mixture-based models. That is mainly because of the changing amount of present symmetries due to the visibility of the given markings.

In general, it is not clear how to visualize a pose. Reporting the values of a 3×3 rotation matrix will not help the reader to check whether the predicted pose is correct or not. Just recently in [46] a new method for visualizing poses was introduced. With the help of Hopf fibrations, they project circles of poses from SO(3) to the 2-sphere and then use the color to indicate the location on the circle. Because of the projection to a lower dimension, limitations do exist. Still, we are happy to use the visualization tool to demonstrate the performance of our model. Figure 2 shows qualitative results of a model trained on SYMSOL II. We plot the ground truth and predicted poses of *cylO* object. The plots illustrate that the model has successfully learned the pose distribution of this object. When the red dot is visible, the model successfully collapses the distribution to predict a small range of poses. When it is not visible, the model outputs a smooth distribution of all possible poses given how the object is visible in the figure.

4.4. Pascal3D+

To analyze whether our approach is applicable to single pose estimation, we conduct additional experiments on a subset of the Pascal3D+ dataset [75]. It consists of a subset of the object categories from the well known PASCAL VOC dataset [15], where 3D annotations are added. Furthermore, the dataset has been enlarged by adding more images from the ImageNet dataset [12]. The annotation of an object consists of the elevation, azimuth, and distance of the camera position in 3D. With at least 3000 instances per category, it is a challenging dataset of real world objects, like *planes, trains, bicycles* and more. The choice of a subset is due to the unavailability of an official dataloader and existing invalid bounding box annotations in the dataset that we handled individually, e.g. manually adding the missing annotations or skipping elements with incorrect annotations. This leads us to exclude quantitative results of the objects where we can not guarantee alignment with publicly available results on this dataset. Still, we show qualitative results in the supplementary material. For the train and test splits, we follow the split provided by [35].

Our implementation specifics are as follows. As the complexity of the images in the Pascal3D+ dataset is higher than in the SYMSOL dataset we choose a larger pretrained ResNet-101 backbone for our vision module. We predict the weights of a one-layer network with a width of 256. Using the Adam optimizer, we evaluate our model after 150k iterations using a batch size of 16. A learning rate of $1e-5$ is used for the first 1000 iterations, and then a cosine decay is applied.

Table 5 shows our evaluations in the standard Acc30° metric and the median angular error. While our method is specifically designed to account for present symmetries, the table shows that we are also competitive in the task of single-pose prediction. In [35] the authors reported values that are incorrectly lowered by a factor of $\sqrt{2}$. Hence we report the corrected values in our experiments. Visualizations can be found in Figure 3, where we display four objects: a bottle, a sofa, a bus, and a tv monitor. With the presence of textures, the pose predictions are unique.

5. Conclusion

Previous works demonstrated that hypernetworks can be used to predict implicit neural representations for the task of 2D and 3D shape reconstruction. To the best of our knowledge we are the first to show that hypernetworks are able to predict the weights of an implicit neural representation associated with the task of pose estimation. HyperPosePDF is able to predict a non-parametric distribution on the rotation manifold, designed to incorporate uncertainty of symmetry, noise and ambiguities. Additionally, we could show that the commonly used Fourier embedding for INRs is also capable of boosting the pose estimation results. Furthermore, we achieve superior performance on the challenging SYMSOL datasets that consist of objects with varying symmetries. Besides that, we are able to maintain comparable performance on single-pose estimation evaluated on the Pascal3D+ dataset.

This work demonstrated promising results in pose estimation tasks. In future works, it would be interesting to see how these insights generalize to new application domains, such as spin detection in table tennis robots or visual-inertial odometry in flying robots.

References

[1] Wim Abbeloos and Toon Goedemé. Point pair feature based object detection for random bin picking. In *2016 13th Conference on Computer and Robot Vision (CRV)*, pages 432–439. IEEE, 2016.

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020.

[3] Jimmy Ba, Geoffrey E Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. *Advances in neural information processing systems*, 29, 2016.

[4] Nuri Benbarka, Timon Höfer, Andreas Zell, et al. Seeing implicit neural representations as fourier series. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2041–2050, 2022.

[5] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[6] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Smash: one-shot model architecture search through hypernetworks. *arXiv preprint arXiv:1708.05344*, 2017.

[7] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision*, pages 608–625. Springer, 2020.

[8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[9] Enric Corona, Kaustav Kundu, and Sanja Fidler. Pose estimation for objects with rotational symmetry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7215–7222. IEEE, 2018.

[10] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *European Conference on Computer Vision*, pages 612–628. Springer, 2020.

[11] Haowen Deng, Mai Bui, Nassir Navab, Leonidas Guibas, Slobodan Ilic, and Tolga Birdal. Deep bingham networks:

Dealing with uncertainty and ambiguity in pose estimation. *International Journal of Computer Vision*, pages 1–28, 2022.

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

[13] Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando De Freitas. Predicting parameters in deep learning. *Advances in neural information processing systems*, 26, 2013.

[14] Vikranth Dwaracherla, Xiuyuan Lu, Morteza Ibrahimi, Ian Osband, Zheng Wen, and Benjamin Van Roy. Hypermodels for exploration. *arXiv preprint arXiv:2006.07464*, 2020.

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[16] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020.

[17] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7154–7164, 2019.

[18] Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep orientation uncertainty learning based on a bingham loss. In *International Conference on Learning Representations*, 2019.

[19] Faustino Gomez and Jürgen Schmidhuber. Evolving modular fast-weight networks for control. In *International Conference on Artificial Neural Networks*, pages 383–389. Springer, 2005.

[20] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

[21] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020.

[22] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[23] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Learning a neural 3d texture space from 2d exemplars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8356–8364, 2020.

[24] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011.

[25] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. 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, 2017.

[26] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2020 on 6d object localization. In *European Conference on Computer Vision*, pages 577–594. Springer, 2020.

[27] Timon Höfer, Faranak Shamsafar, Nuri Benbarka, and Andreas Zell. Object detection and autoencoder-based 6d pose estimation for highly cluttered bin picking. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 704–708. IEEE, 2021.

[28] Yizhou Huang, Kevin Xie, Homanga Bharadhwaj, and Florian Shkurti. Continual model-based reinforcement learning with hypernetworks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 799–805. IEEE, 2021.

[29] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.

[30] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.

[31] Di Kang, Debarun Dhar, and Antoni Chan. Incorporating side information by adaptive convolution. *Advances in Neural Information Processing Systems*, 30, 2017.

[32] Benjamin Klein, Lior Wolf, and Yehuda Afek. A dynamic convolutional layer for short range weather prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4840–4848, 2015.

[33] Jan Koutnik, Faustino Gomez, and Jürgen Schmidhuber. Evolving neural networks in compressed weight space. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 619–626, 2010.

[34] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020.

[35] Shuai Liao, Efstratios Gavves, and Cees GM Snoek. Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9767, 2019.

[36] Gidi Littwin and Lior Wolf. Deep meta functionals for shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1824–1833, 2019.

[37] Diyi Liu, Shogo Arai, Jiaqi Miao, Jun Kinugawa, Zhao Wang, and Kazuhiro Kosuge. Point pair feature-based pose estimation with multiple edge appearance models (ppf-meam) for robotic bin picking. *Sensors*, 18(8):2719, 2018.

[38] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. *Advances in Neural Information Processing Systems*, 32, 2019.

[39] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2019–2028, 2020.

[40] Siddharth Mahendran, Haider Ali, and Rene Vidal. A mixed classification-regression framework for 3d pose estimation from 2d images. *arXiv preprint arXiv:1805.03225*, 2018.

[41] Fabian Manhardt, Diego Martin Arroyo, Christian Rupprecht, Benjamin Busam, Tolga Birdal, Nassir Navab, and Federico Tombari. Explaining the ambiguity of object detection and 6d pose from visual data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6841–6850, 2019.

[42] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[43] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019.

[44] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[45] David Mohlin, Josephine Sullivan, and Gérald Bianchi. Probabilistic orientation estimation with matrix fisher distributions. *Advances in Neural Information Processing Systems*, 33:4884–4893, 2020.

[46] Kieran Murphy, Carlos Esteves, Varun Jampani, Srikumar Ramalingam, and Ameesh Makadia. Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold. *arXiv preprint arXiv:2106.05965*, 2021.

[47] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *International Conference on Computer Vision*, Oct. 2019.

[48] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[49] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4531–4540, 2019.

[50] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *International Conference on Computer Vision*, Oct. 2019.

[51] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[52] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020.

[53] V Peretroukhin, M Giamou, D Rosen, and WN Greene. A smooth representation of belief of so (3) for deep rotation learning with uncertainty. RSS, 2020.

[54] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[55] Jialun Ping, Fan Wang, and Jin-Quan Chen. *Group representation theory for physicists*. World Scientific Publishing Company, 2002.

[56] Giorgia Pitteri, Michaël Ramamonjisoa, Slobodan Ilic, and Vincent Lepetit. On object symmetries and 6d pose estimation from images. In *2019 International Conference on 3D Vision (3DV)*, pages 614–622. IEEE, 2019.

[57] Daniel Potts, Jürgen Prestin, and Antje Vollrath. A fast fourier algorithm on the rotation group. *Preprint A-07-06, Univ. zu Lübeck*, 2007.

[58] Sergey Prokudin, Peter Gehler, and Sebastian Nowozin. Deep directional statistics: Pose estimation with uncertainty quantification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 534–551, 2018.

[59] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019.

[60] Elad Sarafian, Shai Keynan, and Sarit Kraus. Recomposing the reinforcement learning building blocks with hypernetworks. In *International Conference on Machine Learning*, pages 9301–9312. PMLR, 2021.

[61] Pedro Savarese and Michael Maire. Learning implicitly recurrent cnns through parameter sharing. *arXiv preprint arXiv:1902.09701*, 2019.

[62] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pages 9355–9366. PMLR, 2021.

[63] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

[64] Juergen Schmidhuber. Steps towardsself-referential'neural learning: A thought experiment. 1992.

[65] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

[66] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

[67] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10753–10764, 2021.

[68] Kenneth O Stanley, David B D'Ambrosio, and Jason Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial life*, 15(2):185–212, 2009.

[69] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, and Rudolph Triebel. Augmented autoencoders: Implicit 3d orientation learning for 6d object detection. *International Journal of Computer Vision*, 128(3):714–729, 2020.

[70] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *ICML*, 2011.

[71] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020.

[72] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 2020.

[73] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015.

[74] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. Metaavatar: Learning animatable clothed human models from few depth images. *Advances in Neural Information Processing Systems*, 34:2810–2822, 2021.

[75] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[76] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[77] Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. *arXiv preprint arXiv:1810.05749*, 2018.