

# Deep Directional Statistics: Pose Estimation with Uncertainty Quantification

Sergey Prokudin<sup>1</sup>, Peter Gehler <sup>\*2</sup>, and Sebastian Nowozin<sup>3</sup>

<sup>1</sup> Max Planck Institute for Intelligent Systems, Tübingen, Germany,  
[sergey.prokudin@tuebingen.mpg.de](mailto:sergey.prokudin@tuebingen.mpg.de)

<sup>2</sup> Amazon, Tübingen, Germany

<sup>3</sup> Microsoft Research, Cambridge, UK

**Abstract.** Modern deep learning systems successfully solve many perception tasks such as object pose estimation when the input image is of high quality. However, in challenging imaging conditions such as on low resolution images or when the image is corrupted by imaging artifacts, current systems degrade considerably in accuracy. While a loss in performance is unavoidable, we would like our models to quantify their uncertainty in order to achieve robustness against images of varying quality. Probabilistic deep learning models combine the expressive power of deep learning with uncertainty quantification. In this paper we propose a novel probabilistic deep learning model for the task of angular regression. Our model uses *von Mises* distributions to predict a distribution over object pose angle. Whereas a single von Mises distribution is making strong assumptions about the shape of the distribution, we extend the basic model to predict a mixture of von Mises distributions. We show how to learn a mixture model using a finite and *infinite* number of mixture components. Our model allows for likelihood-based training and efficient inference at test time. We demonstrate on a number of challenging pose estimation datasets that our model produces calibrated probability predictions and competitive or superior point estimates compared to the current state-of-the-art.

**Keywords:** pose estimation, deep probabilistic models, uncertainty quantification, directional statistics.

## 1 Introduction

Estimating object pose is an important building block in systems aiming to understand complex scenes and has a long history in computer vision [1,2]. Whereas early systems achieved low accuracy, recent advances in deep learning and the collection of extensive data sets have led to high performing systems that can be deployed in useful applications [3,4,5].

---

\* This work has been done prior to Peter Gehler joining Amazon.

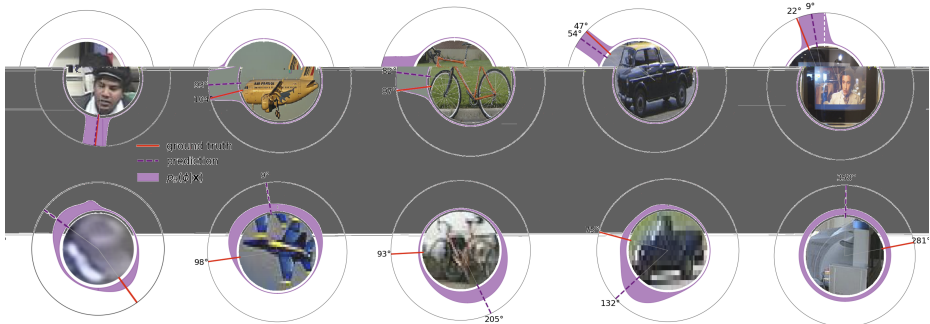


Fig. 1: Our model predicts complex multimodal distributions on the circle (truncated by the outer circle for better viewing). For difficult and ambiguous images our model report high uncertainty (bottom row). Pose estimation predictions (pan angle) on images from IDIAP, TownCentre and PASCAL3D+ datasets.

However, the reliability of object pose regression depends on the quality of the image provided to the system. Key challenges are low-resolution due to distance of an object to the camera, blur due to motion of the camera or the object, and sensor noise in case of poorly lit scenes (see Figure 1).

We would like to predict object pose in a way that captures uncertainty. *Probability* is the right way to capture the uncertainty [6] and in this paper we therefore propose a novel model for object pose regression whose predictions are fully probabilistic. Figure 1 depicts an output of the proposed system. Moreover, instead of assuming a fixed form for the predictive density we allow for very flexible distributions, specified by a deep neural network.

The value of quantified uncertainty in the form of probabilistic predictions is two-fold: *first*, a high prediction uncertainty is a robust way to diagnose poor inputs to the system; *second*, given accurate probabilities we can summarize them to improved point estimates using Bayesian decision theory.

More generally, accurate representation of uncertainty is especially important in case a computer vision system becomes part of a larger system, such as when providing an input signal for an autonomous control system. If uncertainty is not well-calibrated, or—even worse—is not taken into account at all, then the consequences of decisions made by the system cannot be accurately assessed, resulting in poor decisions at best, and dangerous actions at worst.

In the following we present our method and make the following contributions:

- We demonstrate the importance of probabilistic regression on the application of object pose estimation;
- We propose a novel efficient probabilistic deep learning model for the task of circular regression;
- We show on a number of challenging pose estimation datasets (including PASCAL 3D+ benchmark [7]) that the proposed probabilistic method outper-

forms purely discriminative approaches in terms of predictive likelihood and show competitive performance in terms of angular deviation losses classically used for the tasks.

## 2 Related Work

Estimation of object orientation arises in different applications and in this paper we focus on the two most prominent tasks: head pose estimation and object class orientation estimation. Although those tasks are closely related, they have been studied mostly in separation, with methods applied to exclusively one of them. We will therefore discuss them separately, despite the fact that our model applies to both tasks. We present results for our methods on the standard benchmarks from both domains.

*Head pose estimation* has been a subject of extensive research in computer vision for a long time [8,2] and the existing systems vary greatly in terms of feature representation and proposed classifiers. The input to pose estimation systems typically consists of 2D head images [9,10,11], and often one has to cope with low resolution images [12,13,14,8]. Additional modalities such as depth [15] and motion [14,16] information has been exploited and provides useful cues. However, these are not always available. Also, information about the full body image could be used for joint head and body pose prediction [17,18,19]. Notably the work of [18] also promotes a probabilistic view and fuse body and head orientation within a tracking framework. Finally, the output of facial landmarks can be used as an intermediate step [20,21].

Existing head pose estimation models are diverse and include manifold learning approaches [22,23,24,25], energy-based models [19], linear regression based on HOG features [26], regression trees [15,27] and convolutional neural networks [5]. A number of probabilistic methods for head pose analysis exist in the literature [28,29,18], but none of them combine probabilistic framework with learnable hierarchical feature representations from deep CNN architectures. At the same time, deep probabilistic models have shown an advantage over purely discriminative models in other computer vision tasks, e.g., depth estimation [30]. To the best of our knowledge, our work is the first to utilize deep probabilistic approach to angular orientation regression task.

An early dataset for estimating the *object rotation for general object classes* was proposed in [31] along with an early benchmark set. Over the years the complexity of data increased, from object rotation [31] and images of cars in different orientations [32] to Pascal3D [33]. The work of [33] then assigned a separate Deformable Part Model (DPM) component to a discrete set of viewpoints. The work of [34,35] then proposed different 3D DPM extensions which allowed viewpoint estimation as integral part of the model. However, both [34] and [35] do not predict a continuous angular estimate but only a discrete number of bins.

More recent versions make use of CNN models but still do not take a probabilistic approach [3,4]. The work of [36] investigates the use of a synthetic

rendering pipeline to overcome the scarcity of detailed training data. The addition of synthetic and real examples allows them to outperform previous results. The model in [36] predicts angles, and constructs a loss function that penalizes geodesic and  $\ell_1$  distance. In this work we advocate the use of likelihood estimation as a principled probabilistic training objective. The recent work of [37] draws a connection between viewpoints and object keypoints. The viewpoint estimation is however again, framed as a classification problem in terms of Euler angles to obtain a rotation matrix from a canonical viewpoint.

Many works phrase angular prediction as a classification problem [3,37,36] which always limits the granularity of the prediction and also requires the design of a loss function and a means to select the number of discrete labels. A benefit of a classification model is that components like softmax loss can be re-used and also interpreted as an uncertainty estimate. In contrast, our model mitigate this problem: the likelihood principle suggests a direct way to train parameters, moreover ours is the only model in this class that conveys an uncertainty estimate.

### 3 Review of Biternion Networks

We build on the Biternion networks method for pose estimation from [5] and briefly review the basic ideas here. Biternion networks regress angular data and currently define the state-of-the-art model for a number of challenging head pose estimation datasets.

A key problem is to regress angular orientations which is periodic and prevents a straight-forward application of standard regression methods, including CNN models with common loss functions. Consider a ground truth value of  $0^\circ$ , then both predictions  $1^\circ$  and  $359^\circ$  should result in the same absolute loss. Applying the mod operator is no simple fix to this problem, since it results in a discontinuous loss function that complicates the optimization. A loss function needs to be defined to cope with this discontinuity of the target value. Biternion networks overcome this difficulty by using a different parameterization of angles and the cosine loss function between angles.

#### 3.1 Biternion Representation

Beyer et al. [5] propose an alternative representation of an angle  $\phi$  using the two-dimensional sine and cosine components  $\mathbf{y} = (\cos \phi, \sin \phi)$ .

This *biternion representation* is inspired by quaternions, which are popular in computer graphics systems. It is easy to predict a (cos, sin) pair with a fully-connected layer followed by a normalization layer, that is,

$$f_{BT}(\mathbf{x}; \mathbf{W}, \mathbf{b}) = \frac{\mathbf{W}\mathbf{x} + \mathbf{b}}{\|\mathbf{W}\mathbf{x} + \mathbf{b}\|}. \quad (1)$$

A Biternion network is then a convolutional neural network with a layer (1) as the final operation, outputting a two-dimensional vector  $\mathbf{y}_{pred}$ . We use VGG-style network [38] and InceptionResNet [39] networks in our experiments and provide

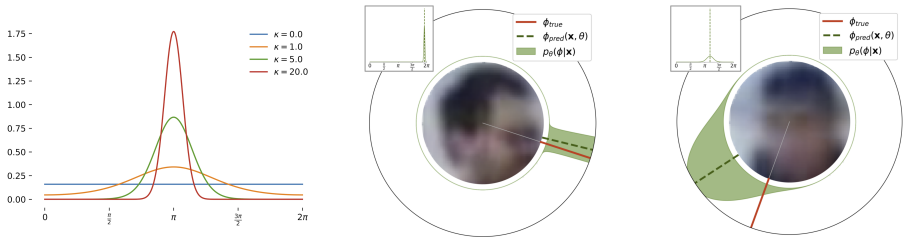


Fig. 2: Left: examples of the von Mises probability density function for different concentration parameters  $\kappa$ . Center, right: predicted  $\mathcal{VM}$  distributions for two images from the CAVIAR dataset. We plot the predicted density on the viewing circle. For comparison we also include the 2D plot (better visible in zoomed pdf version). The distribution on the center image is very certain, the one on the right more uncertain about the viewing angle.

a detailed description of the network architecture in Section 6.1. Given recent developments in network architectures it is likely that different network topologies may perform better than selected backbones. We leave this for future work, our contributions are orthogonal to the choice of the basis model.

### 3.2 Cosine loss function

The cosine distance is chosen in [5] as a natural candidate to measure the difference between the predicted and ground truth Biternion vectors. It reads

$$L_{cos}(\mathbf{y}_{pred}, \mathbf{y}_{true}) = 1 - \frac{\mathbf{y}_{pred} \cdot \mathbf{y}_{true}}{\|\mathbf{y}_{pred}\| \cdot \|\mathbf{y}_{true}\|} = 1 - \mathbf{y}_{pred} \cdot \mathbf{y}_{true}, \quad (2)$$

where the last equality is due to  $\|\mathbf{y}\| = \cos^2 \phi + \sin^2 \phi = 1$ .

The combination of a Biternion angle representation and a cosine loss solves the problems of regressing angular values, allowing for a flexible deep network with angular output. We take this state-of-the-art model and generalize it into a family of probabilistic models of gradually more flexibility.

## 4 Probabilistic models of circular data.

The von Mises (vM) distribution is the basic building block of our probabilistic framework for circular data [40]. We continue with a brief formal definition and in Section 4.1 describe a simple way to convert the output of Biternion networks into a  $\mathcal{VM}$  density, that does not require any network architecture change or re-training as it requires only selection of the model variance. We will then use this approach as a baseline for more advanced probabilistic models. Section 4.2 slightly extends the original Biternion network by introducing an additional

network output unit that models uncertainty of our angle estimation and allows optimization for the log-likelihood of the  $\mathcal{VM}$  distribution.

The von Mises distribution  $\mathcal{VM}(\mu, \kappa)$  is a close approximation of a normal distribution on the unit circle. Its probability density function is

$$p(\phi; \mu, \kappa) = \frac{\exp(\kappa \cos(\phi - \mu))}{2\pi I_0(\kappa)}, \quad (3)$$

where  $\mu \in [0, 2\pi)$  is the mean value,  $\kappa \in \mathbb{R}_+$  is a measure of concentration (a reciprocal measure of dispersion, so  $1/\kappa$  is analogous to  $\sigma^2$  in a normal distribution), and  $I_0(\kappa)$  is the modified Bessel function of order 0. We show examples of  $\mathcal{VM}$ -distributions with  $\mu = \pi$  and varying  $\kappa$  values in Figure 2 (left).

#### 4.1 Von Mises Biternion Networks

A conceptually simple way to turn the Biternion networks from Section 3 into a probabilistic model is to take its predicted value as the center value of the  $\mathcal{VM}$  distribution,

$$p_\theta(\phi|\mathbf{x}; \kappa) = \frac{\exp(\kappa \cos(\phi - \mu_\theta(\mathbf{x})))}{2\pi I_0(\kappa)}, \quad (4)$$

where  $\mathbf{x}$  is an input image,  $\theta$  are parameters of the network, and  $\mu_\theta(\mathbf{x})$  is the network output. In order to arrive at a probability distribution we may regard  $\kappa > 0$  as a hyper-parameter. For fixed network parameters  $\theta$  we can select  $\kappa$  by maximizing the log-likelihood of the observed data,

$$\kappa^* = \operatorname{argmax}_\kappa \sum_{i=1}^N \log p_\theta(\phi^{(i)}|\mathbf{x}^{(i)}; \kappa). \quad (5)$$

The model (4) with  $\kappa^*$  will serve as the simplest probabilistic baseline in our comparisons.

#### 4.2 Maximizing the von Mises Log-likelihood

Using a single scalar  $\kappa$  for every possible input in the model (4) is clearly a restrictive assumption: model certainty should depend on factors such as image quality, light conditions, etc. For example, Figure 2 (center, right) depicts two low resolution images from a surveillance camera that are part of the CAVIAR dataset [13]. In the left image facial features like eyes and ears are distinguishable which allows a model to be more certain when compared to the more blurry image on the right.

We therefore extend the simple model by replacing the single constant  $\kappa$  with a function  $\kappa_\theta(\mathbf{x})$ , predicted by the Biternion network,

$$p_\theta(\phi|\mathbf{x}) = \frac{\exp(\kappa_\theta(\mathbf{x}) \cos(\phi - \mu_\theta(\mathbf{x})))}{2\pi I_0(\kappa_\theta(\mathbf{x}))}. \quad (6)$$

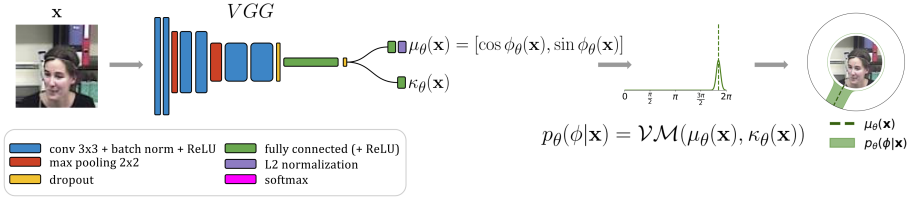


Fig. 3: **The single mode von Mises model (VGG backbone variation).** A BiternionVGG network regresses both mean and concentration parameter of a single vM distribution.

We train (6) by maximizing the log-likelihood of the data,

$$\log \mathcal{L}(\theta|\mathbf{X}, \Phi) = \sum_{i=1}^N \kappa_\theta(\mathbf{x}^{(i)}) \cos(\phi^{(i)} - \mu_\theta(\mathbf{x}^{(i)})) - \sum_{i=1}^N \log 2\pi I_0(\kappa_\theta(\mathbf{x}^{(i)})). \quad (7)$$

Note that when  $\kappa$  is held constant in (7), the second sum in  $\log \mathcal{L}(\theta|\mathbf{X}, \Phi)$  is constant and therefore we recover the Biternion cosine objective (2) up to constants  $C_1, C_2$ ,

$$\log \mathcal{L}(\theta|\mathbf{X}, \Phi, \kappa) = C_1 \sum_{i=1}^N \cos(\phi^{(i)} - \mu_\theta(\mathbf{x}^{(i)})) + C_2.$$

The sum has the equivalent form,

$$\sum_{i=1}^N \cos(\phi^{(i)} - \mu_\theta(\mathbf{x}^{(i)})) = \sum_{i=1}^N [\cos \phi^{(i)} \cos \mu_\theta(\mathbf{x}^{(i)}) + \sin \phi^{(i)} \sin \mu_\theta(\mathbf{x}^{(i)})] \quad (8)$$

$$= \sum_{i=1}^N \mathbf{y}_{\phi^{(i)}} \cdot \mathbf{y}_{\mu_\theta(\mathbf{x}^{(i)})}, \quad (9)$$

where  $\mathbf{y}_\phi = (\cos \phi, \sin \phi)$  is a Biternion representation of an angle. Note, that the above derivation shows that the loss function in [5] corresponds to optimizing the von Mises log-likelihood for the fixed value of  $\kappa = 1$ . This offers an interpretation of Biternion networks as a probabilistic model.

The additional degree of freedom to learn  $\kappa_\theta(\mathbf{x})$  as a function of  $\mathbf{x}$  allows us to capture the desired image-dependent uncertainty as can be seen in Figure 2.

However, like the Gaussian distribution the von Mises distribution makes a specific assumption regarding the shape of the density. We now show how to overcome this limitation by using a mixture of von Mises distributions.

## 5 Mixture of von Mises Distributions

The model described in Section 4.2 is only unimodal and can not capture ambiguities in the image. However, in case of blurry images like the ones in Figure 2

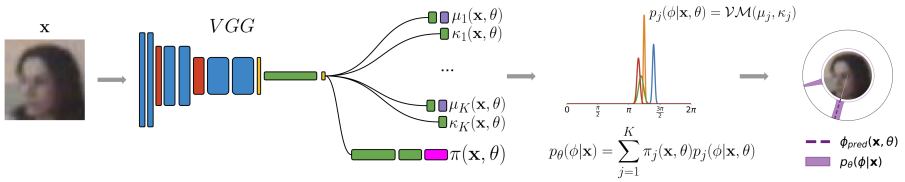


Fig. 4: **The finite  $\mathcal{VM}$  mixture model.** A VGG network predicts  $K$  mean and concentration values and the mixture coefficients  $\pi$ . This allows to capture multimodality in the output.

we could be interested in distributing the mass around a few potential high probability hypotheses, for example, the model could predict that a person is looking sideways, but could not determine the direction, left or right, with certainty. In this section we present two models that are able to capture multimodal beliefs while retaining a calibrated uncertainty measure.

### 5.1 Finite Mixture of von Mises Distributions

One common way to generate complex distributions is to sum multiple distributions into a *mixture distribution*. We introduce  $K$  different component distributions and a  $K$ -dimensional probability vector representing the mixture weights. Each component is a simple von Mises distribution. We can then represent our density function as

$$p_{\theta}(\phi|\mathbf{x}) = \sum_{j=1}^K \pi_j(\mathbf{x}, \theta) p_j(\phi|\mathbf{x}, \theta), \quad (10)$$

where  $p_j(\phi|\mathbf{x}, \theta) = \mathcal{VM}(\phi|\mu_j, \kappa_j)$  for  $j = 1, \dots, K$  are the  $K$  component distributions and the mixture weights are  $\pi_j(\mathbf{x}, \theta)$  so that  $\sum_j \pi_j(\mathbf{x}, \theta) = 1$ . We denote all parameters with the vector  $\theta$ , it contains component-specific parameters as well as parameters shared across all components.

To predict the mixture in a neural network framework, we need  $K \times 3$  output units for modeling all von Mises component parameters (two for modeling the Bivernion representation of the mean,  $\mu_j(\mathbf{x}, \theta)$  and one for the  $\kappa_j(\mathbf{x}, \theta)$  value), as well as  $K$  units for the probability vector  $\pi_j(\mathbf{x}, \theta)$ , defined by taking the **softmax** operation to get a positive mixture weights.

The finite von Mises density model then takes form

$$p_{\theta}(\phi|\mathbf{x}) = \sum_{j=1}^K \pi_j(\mathbf{x}, \theta) \frac{\exp\left(\kappa_j(\mathbf{x}, \theta) \cos(\phi - \mu_j(\mathbf{x}, \theta))\right)}{2\pi I_0(\kappa_j(\mathbf{x}, \theta))}. \quad (11)$$

Similarly to the single von Mises model, we can train by maximizing the log-likelihood of the observed data,  $\sum_{i=1}^N \log p_{\theta}(\phi^{(i)}|\mathbf{x}^{(i)})$ . We show an overview of the model in Figure 4.



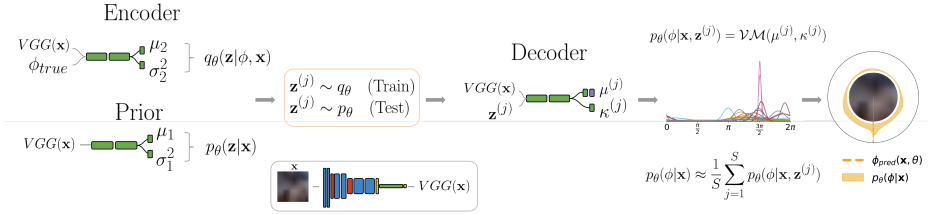


Fig. 5: **The infinite mixture model (CVAE)**. An encoder network predicts a distribution  $q(\mathbf{z}|\mathbf{x})$  over latent variables  $\mathbf{z}$ , and a decoder network  $p(\phi|\mathbf{x}, \mathbf{z})$  defines individual mixture components. Integrating over  $\mathbf{z}$  yields an infinite mixture of von Mises distributions. In practice we approximate this integration using a finite number of Monte Carlo samples  $\mathbf{z}^{(j)} \sim q(\mathbf{z}|\mathbf{x})$ .

## 5.2 Infinite Mixture (CVAE)

To extend the model from a finite to an infinite mixture model, we follow the variational autoencoder (VAE) approach [41,42], and introduce a vector-valued latent variable  $\mathbf{z}$ . The resulting model is depicted in Figure 5. The continuous latent variable becomes the input to a decoder network  $p(\phi|\mathbf{x}, \mathbf{z})$  which predicts the parameters—mean and concentration—of a single von Mises component. We define our density function as the infinite sum (integral) over all latent variable choices, weighted by a learned distribution  $p(\mathbf{z}|\mathbf{x})$ ,

$$p_{\theta}(\phi|\mathbf{x}) = \int p(\phi|\mathbf{x}, \mathbf{z}) p(\mathbf{z}|\mathbf{x}) d\mathbf{z}, \quad (12)$$

where  $p_{\theta}(\phi|\mathbf{x}, \mathbf{z}) = \mathcal{VM}(\mu(\mathbf{x}, \theta), \kappa(\mathbf{x}, \theta))$ , and  $p_{\theta}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu_1(\mathbf{x}, \theta), \sigma_1^2(\mathbf{x}, \theta))$ . The log-likelihood  $\log p_{\theta}(\phi|\mathbf{x})$  for this model is not longer tractable, preventing simple maximum likelihood training. Instead we use the variational autoencoder framework of [41,42] in the form of the conditional VAE (CVAE) [43]. The CVAE formulation uses an auxiliary *variational* density  $q_{\theta}(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mu_2(\mathbf{x}, \phi, \theta), \sigma_2^2(\mathbf{x}, \phi, \theta))$  and instead of the log-likelihood optimizes a *variational lower bound*,

$$\log p_{\theta}(\phi|\mathbf{x}) = \log \int p_{\theta}(\phi|\mathbf{x}, \mathbf{z}) p_{\theta}(\mathbf{z}|\mathbf{x}) d\mathbf{z} \quad (13)$$

$$\geq \mathbb{E}_{\mathbf{z} \sim q_{\theta}(\mathbf{z}|\mathbf{x}, \phi)} \left[ \log \frac{p_{\theta}(\phi|\mathbf{x}, \mathbf{z}) p_{\theta}(\mathbf{z}|\mathbf{x})}{q_{\theta}(\mathbf{z}|\mathbf{x}, \phi)} \right] =: \mathcal{L}_{\text{ELBO}}(\theta|\mathbf{x}, \phi). \quad (14)$$

We refer to [41,43,44,42] for more details on VAEs.

The CVAE model is composed of multiple deep neural networks: an *encoder network*  $q_{\theta}(\mathbf{z}|\mathbf{x}, \phi)$ , a *conditional prior network*  $p_{\theta}(\mathbf{z}|\mathbf{x})$ , and a *decoder network*  $p_{\theta}(\phi|\mathbf{x}, \mathbf{z})$ . Like before, we use  $\theta$  to denote the entirety of trainable parameters of all three model components. We show an overview of the model in Figure 5. The model is trained by maximizing the variational lower bound (14) over the training set  $(\mathbf{X}, \Phi)$ , where  $\mathbf{X} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)})$  are the images and  $\Phi = (\phi^{(1)}, \dots, \phi^{(N)})$

are the ground truth angles. We maximize

$$\hat{\mathcal{L}}_{\text{CVAE}}(\theta|\mathbf{X}, \Phi) = \frac{1}{N} \sum_{i=1}^N \hat{\mathcal{L}}_{\text{ELBO}}(\theta|\mathbf{x}^{(i)}, \phi^{(i)}), \quad (15)$$

where we use  $\hat{\mathcal{L}}_{\text{ELBO}}$  to denote the Monte Carlo approximation to (14) using  $S$  samples. We can optimize (15) efficiently using stochastic gradient descent.

To evaluate the log-likelihood during testing, we use the importance-weighted sampling technique proposed in [45] to derive a stronger bound on the marginal likelihood,

$$\log p_{\theta}(\phi|\mathbf{x}) \geq \log \frac{1}{S} \sum_{j=1}^S \frac{p_{\theta}(\phi|\mathbf{x}, \mathbf{z}^{(j)}) p_{\theta}(\mathbf{z}^{(j)}|\mathbf{x})}{q_{\theta}(\mathbf{z}^{(j)}|\mathbf{x}, \phi)}, \quad (16)$$

$$\mathbf{z}^{(j)} \sim q_{\theta}(\mathbf{z}^{(j)}|\mathbf{x}, \phi) \quad j = 1, \dots, S. \quad (17)$$

**Simplified CVAE.** In our experiments we also investigate a variant of the aforementioned model where  $p_{\theta}(\mathbf{z}|\mathbf{x}) = q_{\theta}(\mathbf{z}|\mathbf{x}, \phi) = p(\mathbf{z}) = \mathcal{N}(0, I)$ . Compared to the full CVAE framework, this model, which we refer to as *simplified CVAE* (sCVAE) in the experiments, sacrifices the adaptive input-dependent density of the hidden variable  $\mathbf{z}$  for faster training and test inference as well as optimization stability. In that case the KL-divergence  $KL(q_{\theta} \parallel p_{\theta})$  term in  $\hat{\mathcal{L}}_{\text{ELBO}}$  becomes zero, and we train for a Monte Carlo estimated log-likelihood of the data:

$$\hat{\mathcal{L}}_{\text{sCVAE}}(\theta|\mathbf{X}, \Phi) = \frac{1}{N} \sum_{i=1}^N \log \left( \frac{1}{S} \sum_{j=1}^S p_{\theta}(\phi^{(i)}|\mathbf{x}^{(i)}, \mathbf{z}^{(j)}) \right), \quad (18)$$

$$\mathbf{z}^{(j)} \sim p(\mathbf{z}) = \mathcal{N}(0, I), j = 1, \dots, S. \quad (19)$$

In some applications it is necessary to make a single best guess about the pose, that is, to summarize the posterior  $p(\phi|\mathbf{x})$  to a single point prediction  $\hat{\phi}$ . We now discuss an efficient way to do that.

### 5.3 Point Prediction

To obtain an optimal single point prediction we utilize Bayesian decision theory [6,46,47] and minimize the expected loss,

$$\hat{\phi}_{\Delta} = \operatorname{argmin}_{\phi \in [0, 2\pi)} \mathbb{E}_{\phi' \sim p(\phi|\mathbf{x})} [\Delta(\phi, \phi')], \quad (20)$$

where  $\Delta : [0, 2\pi) \times [0, 2\pi) \rightarrow \mathbb{R}_+$  is a loss function. We will use the  $\Delta_{\text{AAD}}(\phi, \phi')$  loss which measures the absolute angular deviation (AAD). To approximate (20) we use the empirical approximation of [46] and draw  $S$  samples  $\{\phi_j\}$  from  $p_{\theta}(\phi|\mathbf{x})$ . We then use the empirical approximation

$$\hat{\phi}_{\Delta} = \operatorname{argmin}_{j=1, \dots, S} \frac{1}{S} \sum_{k=1}^S \Delta(\phi_j, \phi_k). \quad (21)$$

We now evaluate our models both in terms of uncertainty as well as in terms of point prediction quality.

## 6 Experiments

This section presents the experimental results on several challenging head and object pose regression tasks. Section 6.1 introduces the experimental setup including used datasets, network architecture and training setup. In Section 6.2 we present and discuss qualitative and quantitative results on the datasets of interest.

### 6.1 Experimental Setup

**Network architecture.** We build our probabilistic framework on top of the Biternion network approach. Therefore, for all the experiments on the head pose regression, we use the same deep batch-normalized VGG-style network [38] architecture as in [5]. The architecture consists of six convolutional layers with 24, 24, 48, 48, 64 and 64 feature channels, respectively, followed by a fully-connected layer of a variable length. The final layer was set to match the number of required parameters of the probabilistic model. For the CVAE, the same architecture is used for both the encoder and decoder network.

For the experiments on the PASCAL3D+ dataset [33] we use InceptionResNet [39] as a backbone architecture and jointly predict distributions over three angles (azimuth, elevation and tilt). We use a separate model for each class of objects and consider constructing a single shot probabilistic object detector and pose estimator a future work. All models were implemented in Keras [48] using the TensorFlow [49] back-end. Code and data for all experiments are available at [https://github.com/sergeyprokudin/deep\\_direct\\_stat](https://github.com/sergeyprokudin/deep_direct_stat).

**Training.** We optimize using Adam [50] and perform random search proposed in [51] for finding the best values of hyper-parameters such as dropout values, batch size, fully connected layers sizes, learning rate, and other optimizer parameters. For the headpose estimation tasks we train all networks for 1000 epochs, with an early stopping in case of no improvement of validation loss after 200 consecutive step. For PASCAL3D+ we train for 200 epochs with an early stopping after 10 epochs of no improvement.

**Head pose datasets.** We evaluate all methods together with the non-probabilistic BiternionVGG baseline on three diverse (in terms of image quality and precision of provided ground truth information) headpose datasets: IDIAP head pose [9], TownCentre [52] and CAVIAR [13] coarse gaze estimation. The IDIAP head pose dataset contains 66295 head images stemmed from a video recording of a few people in a meeting room. Each image has a complete annotation of a head pose orientation in form of pan, tilt and roll angles. We take 42304, 11995 and 11996 images for training, validation, and testing, respectively. The TownCentre and CAVIAR datasets present a challenging task of a coarse gaze estimation of pedestrians based on low resolution images from surveillance camera

Table 1: Quantitative results on the IDIAP head pose estimation dataset [9] for the three head rotations pan, roll and tilt. In the situation of fixed camera pose, lightning conditions and image quality, all methods show similar performance (methods are considered to perform on par when the difference in performance is less than *standard error of the mean*).

estimated pose component	pan		tilt		roll	
	MAAD	log-likelihood	MAAD	log-likelihood	MAAD	log-likelihood
Beyer et al. ([5]), fixed $\kappa$	<b>5.8° ± 0.1*</b>	0.37 ± 0.01	<b>2.4° ± 0.1</b>	1.31 ± 0.01	<b>3.1° ± 0.1</b>	<b>1.13 ± 0.01</b>
Ours (single von Mises)	6.3° ± 0.1	<b>0.56 ± 0.01</b>	<b>2.3° ± 0.1</b>	<b>1.56 ± 0.01</b>	3.4° ± 0.1	<b>1.13 ± 0.01</b>
Ours (mixture-CVAE)	6.4° ± 0.1	≈ 0.52 ± 0.02	2.9° ± 0.1	≈ 1.35 ± 0.01	3.5° ± 0.1	≈ 1.05 ± 0.02

\*standard error of the mean (SEM).

Table 2: Quantitative results on the CAVIAR-o [13] and TownCentre [52] coarse gaze estimation datasets. We see clear improvement in terms of quality of probabilistic predictions for both datasets when switching to mixture models that allow to output multiple hypotheses for gaze direction.

	CAVIAR-o		TownCentre	
	MAAD	log-likelihood	MAAD	log-likelihood
Beyer et al. ([5]), fixed $\kappa$	5.74° ± 0.13	0.262 ± 0.031	22.8° ± 1.0	-0.89 ± 0.06
Ours (single von Mises)	5.53° ± 0.13	0.700 ± 0.043	22.9° ± 1.1	-0.57 ± 0.05
Ours (mixture-finite)	<b>4.21° ± 0.16</b>	<b>1.87 ± 0.04</b>	23.5° ± 1.1	<b>-0.50 ± 0.04</b>

videos. In case of the CAVIAR dataset, we focus on the part of the dataset containing occluded head instances (hence referred to as CAVIAR-o in the literature). We use (10802, 5444, 5445) and (6916, 874, 904) images for the training-validation-testing split for the CAVIAR and TownCentre datasets, respectively.

**PASCAL3D+ object pose dataset.** The Pascal 3D+ dataset [33] consists of images from the Pascal [53] and ImageNet [54] datasets that have been labeled with both detection and continuous pose annotations for the 12 rigid object categories that appear in Pascal VOC12 [53] train and validation set. With nearly 3000 object instances per category, this dataset provide a rich testbed to study general object pose estimation. In our experiments on this dataset we follow the same protocol as in [36,37] for viewpoint estimation: we use ground truth detections for both training and testing, and use Pascal validation set to evaluate and compare the quality of our predictions.

## 6.2 Results and Discussion

**Quantitative results.** We evaluate our methods using both discriminative and probabilistic metrics. We use discriminative metrics that are standard for the dataset of interest in order to be able to compare our methods with previous work. For headpose tasks we use the mean absolute angular deviation (MAAD), a widely used metric for angular regression tasks. For PASCAL3D+ we use the metrics advocated in [37]. Probabilistic predictions are measured in terms of log-likelihood [55,56], a widely accepted scoring rule for assessing the quality

Table 3: Results on PASCAL3D+ viewpoint estimation with ground truth bounding boxes. First two evaluation metrics are defined in [37], where  $Acc_{\frac{\pi}{6}}$  measures accuracy (the higher the better) and  $MedErr$  measures error (the lower the better). Additionally, we report the log-likelihood estimation  $\log \mathcal{L}$  of the predicted angles (the higher the better). We can see clear improvement on all metrics when switching to probabilistic setting compared to training for a purely discriminative loss (fixed  $\kappa$  case).

	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
$Acc_{\frac{\pi}{6}}$ (Tulsiani et al.[37])	0.81	0.77	0.59	0.93	<b>0.98</b>	0.89	<b>0.80</b>	0.62	0.88	0.82	0.80	0.80	0.81
$Acc_{\frac{\pi}{6}}$ (Su et al.[36])	0.80	0.82	<b>0.62</b>	0.95	0.93	0.83	0.75	<b>0.86</b>	0.86	0.85	0.82	0.89	0.83
$Acc_{\frac{\pi}{6}}$ (Ours, fixed $\kappa$ )	0.83	0.75	0.54	0.95	0.92	0.90	0.77	0.71	<b>0.90</b>	0.82	0.80	0.86	0.81
$Acc_{\frac{\pi}{6}}$ (Ours, single v.Mises)	0.87	0.78	0.55	<b>0.97</b>	0.95	<b>0.91</b>	0.78	0.76	<b>0.90</b>	0.87	<b>0.84</b>	<b>0.91</b>	<b>0.84</b>
$Acc_{\frac{\pi}{6}}$ (Ours, mixture-sCVAE)	<b>0.89</b>	<b>0.83</b>	0.46	0.96	0.93	0.90	<b>0.80</b>	0.76	<b>0.90</b>	<b>0.90</b>	0.82	<b>0.91</b>	<b>0.84</b>
$MedErr$ (Tulsiani et al.[37])	13.8	17.7	21.3	12.9	5.8	9.1	14.8	15.2	14.7	13.7	8.7	15.4	13.6
$MedErr$ (Su et al.[36])	10.0	<b>12.5</b>	<b>20.0</b>	<b>6.7</b>	4.5	6.7	12.3	<b>8.6</b>	13.1	11.0	5.8	13.3	<b>10.4</b>
$MedErr$ (Ours, fixed $\kappa$ )	11.4	18.1	28.1	6.9	4.0	6.6	14.6	12.1	12.9	16.4	7.0	12.9	12.6
$MedErr$ (Ours, single v.Mises)	<b>9.7</b>	17.7	26.9	<b>6.7</b>	<b>2.7</b>	4.9	12.5	8.7	13.2	10.0	4.7	<b>10.6</b>	10.7
$MedErr$ (Ours, mixture-sCVAE)	<b>9.7</b>	15.5	45.6	<b>5.4</b>	2.9	<b>4.5</b>	13.1	12.6	<b>11.8</b>	<b>9.1</b>	<b>4.3</b>	12.0	12.2
$\log \mathcal{L}$ (Ours, fixed $\kappa$ )	-0.89	<b>-0.73</b>	-1.21	0.18	2.09	1.43	-0.08	0.69	<b>-0.50</b>	-0.75	0.06	-1.02	-0.07 $\pm$ 0.15
$\log \mathcal{L}$ (Ours, single v.Mises)	0.19	-1.12	-0.30	2.40	<b>4.87</b>	<b>2.85</b>	<b>0.42</b>	<b>0.79</b>	-0.72	<b>-0.54</b>	<b>2.52</b>	0.52	<b>1.17 <math>\pm</math> 0.07</b>
$\log \mathcal{L}$ (Ours, mixture-sCVAE)	<b>0.60</b>	<b>-0.73</b>	<b>-0.26</b>	<b>2.71</b>	4.45	2.52	-0.58	0.08	-0.62	-0.64	2.05	<b>1.14</b>	<b>1.15 <math>\pm</math> 0.07</b>

of probabilistic predictions. We summarize the results in Tables 1, 2 and 3. It can be seen from results on IDIAP dataset presented in Table 1 that when camera pose, lightning conditions and image quality are fixed, all methods perform similarly. In contrast, for the coarse gaze estimation task on CAVIAR we can see a clear improvement in terms of quality of probabilistic predictions for both datasets when switching to mixture models that allow to output multiple hypotheses for gaze direction. Here low resolution, pure light conditions and presence of occlusions create large diversity in the level of head pose expressions. Finally, on a challenging PASCAL3D+ dataset we can see clear improvement on all metrics and classes when switching to a probabilistic setting compared to training for a purely discriminative loss (fixed  $\kappa$  case). Our methods also show competitive or superior performance compared to state-of-the-art methods on discriminative metrics advocated in [37]. Method of [36] uses large amounts of synthesized images in addition to the standard training set that was used by our method. Using this data augmentation technique can also lead to an improved performance of our method and we consider this future work.

**Qualitative results.** Examples of probabilistic predictions for PASCAL3D+ dataset are shown in Figure 6. The first row highlights the effect we set out to achieve: to correctly quantify the level of uncertainty of the estimated pose. For easier examples we observe sharp peaks and a highly confident detection, and more spread-out densities otherwise. The examples on the second row highlight the advantage of mixture models, which allow to model complex densities with multiple peaks corresponding to more than one potential pose angle. Failure scenarios are highlighted in the last row: high confidence predictions in case if the model confuses head and tail of an object and tendency to uniform distributions ( $\kappa \rightarrow 0$ ) for hard classes.

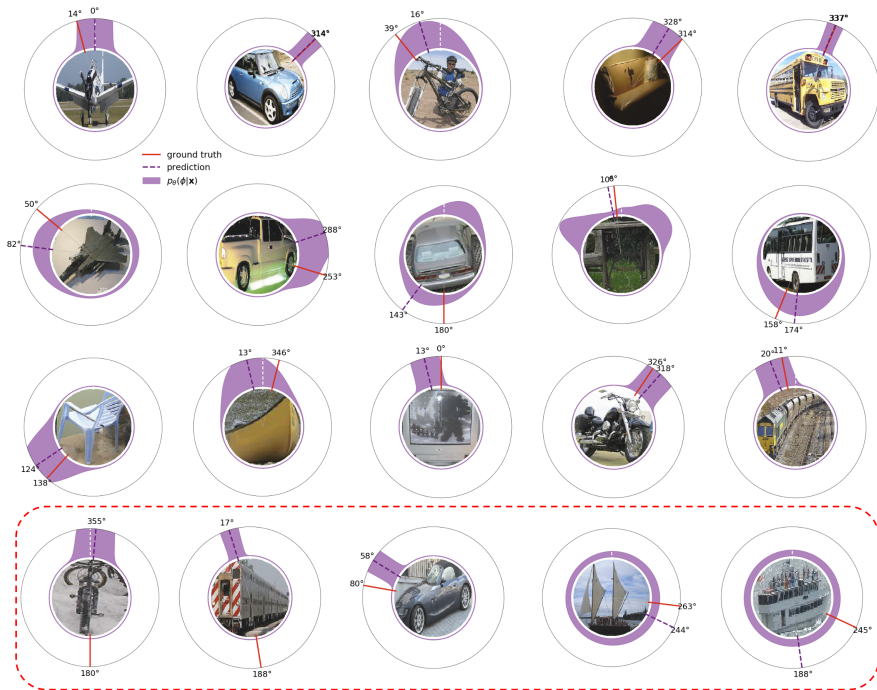


Fig. 6: Qualitative results of our simplified CVAE model on the PASCAL3D+ dataset. Our model correctly quantifies the uncertainty of pose predictions and is able to model ambiguous cases by predicting complex multimodal densities (second row). Last row shows failure cases (confusing head and tail of the object with high confidence, uniform densities in hard cases).

## 7 Conclusion

We demonstrated a new probabilistic model for object pose estimation that is robust to variations in input image quality and accurately quantifies its uncertainty. More generally our results confirm that our approach is flexible enough to accommodate different output domains such as angular data and enables rich and efficient probabilistic deep learning models. We train all models by maximum likelihood but still find it to be competitive with other works from the literature that explicitly optimize for point estimates even under point estimate loss functions. In the future, in order to improve our predictive performance and robustness, we would also like to handle uncertainty of model parameters [30] and to use the Fisher-von Mises distribution to jointly predict a distribution of azimuth-elevation-tilt [40].

We hope that as intelligent systems increasingly rely on perception abilities, future models in computer vision will be robust and probabilistic.

## Acknowledgments

This work was supported by Microsoft Research through its PhD Scholarship Programme.

## References

1. Marchand, E., Uchiyama, H., Spindler, F.: Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics* **22**(12) (2016) 2633–2651 [1](#)
2. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence* **31**(4) (2009) 607–626 [1](#), [3](#)
3. Poirson, P., Ammirato, P., Fu, C.Y., Liu, W., Kosecka, J., Berg, A.C.: Fast single shot detection and pose estimation. In: *3D Vision (3DV), 2016 Fourth International Conference on, IEEE* (2016) 676–684 [1](#), [3](#), [4](#)
4. Massa, F., Marlet, R., Aubry, M.: Crafting a multi-task cnn for viewpoint estimation. *arXiv preprint arXiv:1609.03894* (2016) [1](#), [3](#)
5. Beyer, L., Hermans, A., Leibe, B.: Biternion nets: Continuous head pose regression from discrete training labels. In: *German Conference on Pattern Recognition, Springer International Publishing* (2015) 157–168 [1](#), [3](#), [4](#), [5](#), [7](#), [11](#), [12](#)
6. Berger, J.O.: *Statistical Decision Theory and Bayesian Analysis*. Springer (1980) [2](#), [10](#)
7. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on, IEEE* (2014) 75–82 [2](#)
8. Siriteerakul, T.: Advance in head pose estimation from low resolution images: A review. *International Journal of Computer Science Issues* **9**(2) (2012) [3](#)
9. Odobez, J.M.: IDIAP Head Pose Database. <https://www.idiap.ch/dataset/headpose> [3](#), [11](#), [12](#)
10. Gourier, N., Hall, D., Crowley, J.L.: Estimating face orientation from robust detection of salient facial structures. In: *FG Net Workshop on Visual Observation of Deictic Gestures. Volume 6*. (2004) [3](#)
11. Demirkus, M., Clark, J.J., Arbel, T.: Robust semi-automatic head pose labeling for real-world face video sequences. *Multimedia Tools and Applications* **70**(1) (2014) 495–523 [3](#)
12. Murphy-Chutorian, E., Doshi, A., Trivedi, M.M.: Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In: *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, IEEE* (2007) 709–714 [3](#)
13. Fisher, R., Santos-Victor, J., Crowley, J.: Caviar: Context aware vision using image-based active recognition (2005) [3](#), [6](#), [11](#), [12](#)
14. Benfold, B., Reid, I.: Unsupervised learning of a scene-specific coarse gaze estimator. In: *Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE* (2011) 2344–2351 [3](#)
15. Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE* (2011) 617–624 [3](#)

16. Chamveha, I., Sugano, Y., Sugimura, D., Siriteerakul, T., Okabe, T., Sato, Y., Sugimoto, A.: Head direction estimation from low resolution images with scene adaptation. *Computer Vision and Image Understanding* **117**(10) (2013) 1502–1511 [3](#)
17. Chen, C., Odobez, J.M.: We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance video. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE* (2012) 1544–1551 [3](#)
18. Flohr, F., Dumitru-Guzu, M., Kooij, J.F.P., Gavrila, D.: A probabilistic framework for joint pedestrian head and body orientation estimation. *IEEE Transactions on Intelligent Transportation Systems* **16** (2015) 1872–1882 [3](#)
19. Osadchy, M., Cun, Y.L., Miller, M.L.: Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research* **8**(May) (2007) 1197–1215 [3](#)
20. Dantone, M., Gall, J., Fanelli, G., Van Gool, L.: Real-time facial feature detection using conditional regression forests. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE* (2012) 2578–2585 [3](#)
21. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE* (2012) 2879–2886 [3](#)
22. Lu, J., Tan, Y.P.: Ordinary preserving manifold analysis for human age and head pose estimation. *IEEE Transactions on Human-Machine Systems* **43**(2) (2013) 249–258 [3](#)
23. Huang, D., Storer, M., De la Torre, F., Bischof, H.: Supervised local subspace learning for continuous head pose estimation. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE* (2011) 2921–2928 [3](#)
24. Tosato, D., Spera, M., Cristani, M., Murino, V.: Characterizing humans on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(8) (2013) 1972–1984 [3](#)
25. BenAbdelkader, C.: Robust head pose estimation using supervised manifold learning. *Computer Vision–ECCV 2010* (2010) 518–531 [3](#)
26. Geng, X., Xia, Y.: Head pose estimation based on multivariate label distribution. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* (2014) 1837–1842 [3](#)
27. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* (2014) 1867–1874 [3](#)
28. Ba, S.O., Odobez, J.M.: A probabilistic framework for joint head tracking and pose estimation. In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. Volume 4., IEEE* (2004) 264–267 [3](#)
29. Demirkus, M., Precup, D., Clark, J.J., Arbel, T.: Probabilistic temporal head pose estimation using a hierarchical graphical model. In: *European Conference on Computer Vision, Springer* (2014) 328–344 [3](#)
30. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977* (2017) [3](#), [14](#)
31. Savarese, S., Fei-Fei, L.: 3d generic object categorization, localization and pose estimation. In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE* (2007) 1–8 [3](#)
32. Ozuysal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition.* (2009) 778–785 [3](#)



33. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: IEEE Winter Conference on Applications of Computer Vision (WACV). (2014) [3](#), [11](#), [12](#)
34. Pepik, B., Gehler, P., Stark, M., Schiele, B.: 3d2pm – 3d deformable part models. In: Proceedings of the European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science, Firenze, Springer (October 2012) 356–370 [3](#)
35. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3d geometry to deformable part models. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, IEEE (June 2012) 3362–3369 oral presentation. [3](#)
36. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: 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. (2015) 2686–2694 [3](#), [4](#), [12](#), [13](#)
37. Tulsiani, S., Malik, J.: Viewpoints and keypoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1510–1519 [4](#), [12](#), [13](#)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) [4](#), [11](#)
39. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. Volume 4. 12 [4](#), [11](#)
40. Mardia, K.V., Jupp, P.E.: Directional statistics. Volume 494. John Wiley & Sons (2009) [5](#), [14](#)
41. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) [9](#)
42. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082 (2014) [9](#)
43. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: Advances in Neural Information Processing Systems. (2015) 3483–3491 [9](#)
44. Doersch, C.: Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908 (2016) [9](#)
45. Burda, Y., Grosse, R., Salakhutdinov, R.: Importance weighted autoencoders. arXiv preprint arXiv:1509.00519 (2015) [10](#)
46. Premachandran, V., Tarlow, D., Batra, D.: Empirical minimum Bayes risk prediction: How to extract an extra few % performance from vision models with just three more parameters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1043–1050 [10](#)
47. Bouchacourt, D., Mudigonda, P.K., Nowozin, S.: DISCO Nets: DISSimilarity COefficients Networks. In: Advances in Neural Information Processing Systems. (2016) 352–360 [10](#)
48. Chollet, F., et al.: Keras. <https://github.com/fchollet/keras> (2015) [11](#)
49. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467 (2016) [11](#)
50. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) [11](#)
51. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research **13**(Feb) (2012) 281–305 [11](#)
52. Benfold, B., Reid, I.: Stable multi-target tracking in real-time surveillance video. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 3457–3464 [11](#), [12](#)

53. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *International journal of computer vision* **88**(2) (2010) 303–338 [12](#)
54. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009)* 248–255 [12](#)
55. Good, I.J.: Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)* (1952) 107–114 [12](#)
56. Gneiting, T., Raftery, A.E.: Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* **102**(477) (2007) 359–378 [12](#)