

Deep Bingham Networks

Dealing with Uncertainty and Ambiguity in Pose Estimation

Haowen Deng^{1,2} · Mai Bui¹ · Nassir Navab¹ · Leonidas Guibas³ · Slobodan Ilic² · Tolga Birdal³

Abstract In this work, we introduce *Deep Bingham Networks (DBN)*, a generic framework that can naturally handle pose-related uncertainties and ambiguities arising in almost all real life applications concerning 3D data. While existing works strive to find a single solution to the pose estimation problem, we make peace with the ambiguities causing high uncertainty around which solutions to identify as the best. Instead, we report a *family of poses* which capture the nature of the solution space. DBN extends the state of the art direct pose regression networks by (i) a multi-hypotheses prediction head which can yield different distribution modes; and (ii) novel loss functions that benefit from Bingham distributions on rotations. This way, DBN can work both in unambiguous cases providing uncertainty information, and in ambiguous scenes where an uncertainty per mode is desired. On a technical front, our network regresses continuous *Bingham mixture models* and is applicable to both 2D data such as images and to 3D data such as point clouds. We proposed new training strategies so as to avoid mode or posterior collapse during training and to improve numerical

H. Deng*
E-mail: haowen.deng@tum.de

M. Bui*
E-mail: mai.bui@tum.de

N. Navab
E-mail: nassir.navab@tum.de

L. Guibas
E-mail: guibas@cs.stanford.edu

S. Ilic
E-mail: slobodan.ilic@tum.de

T. Birdal [0000-0001-7915-7964]
E-mail: t.birdal@stanford.edu

1. Informatics at Technische Universität München, Munich, Germany ·
 2. Corporate Technology Siemens AG, Munich, Germany ·
 3. Computer Science Department, Stanford University, CA USA ·
- * shared first authorship

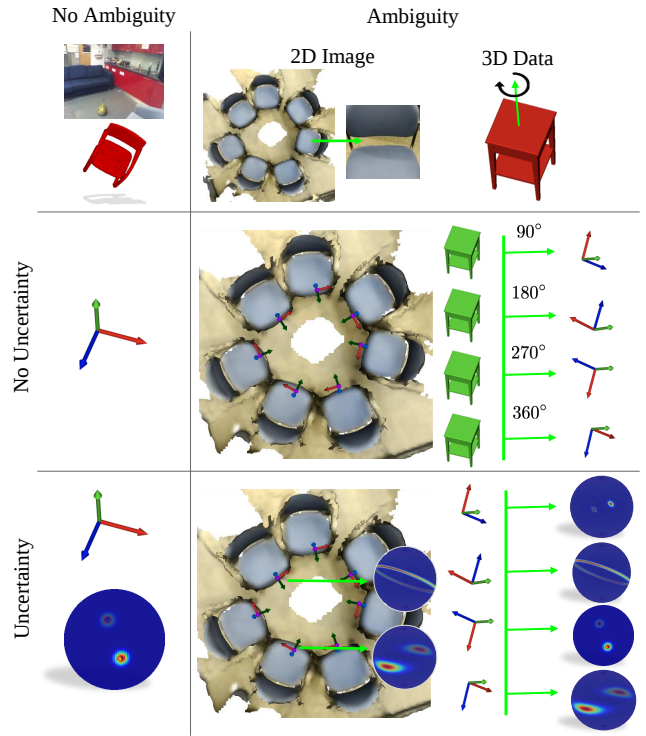


Fig. 1: DBM as a generic framework, addresses the following problems arising in pose prediction methods: Estimating 1) a single pose from non-ambiguous input data, 2) multiple pose hypotheses in case of ambiguities; and in addition, associating uncertainty both 3) to a single pose and 4) to all of the hypotheses in the multi-modal predictions.

stability. Our methods are thoroughly tested on two different applications exploiting two different modalities: (i) 6D camera relocalization from images; and (ii) object pose estimation from 3D point clouds, demonstrating decent advantages over the state of the art. For the former we contributed our own dataset composed of five indoor scenes where it is

unavoidable to capture images corresponding to views that are hard to uniquely identify. For the latter we achieve the top results especially for symmetric objects of ModelNet dataset [96]. The code and dataset accompanying this paper is provided under [multimodal3dvision.github.io](https://github.com/multimodal3dvision).

1 Introduction

A majority of tasks in computer vision can be interpreted as scene understanding problems conditioned on either 2D image or 3D scan modalities. Usually, these scenes are digitizations of our man made environments composed of *objects*. Hence, a fundamental piece of this perception problem is *pose estimation*, i.e. figuring out how these objects are positioned and oriented in 3D space. A *rigid transformation* is a six degrees of freedom (6-DoF) entity explaining the pose either of an acquisition device (e.g. Lidar or camera) or an object enclosed within the captured data. Solving for the former is known as *camera relocalization*, while the latter is related to *3D object pose estimation*. Both of these are now key technologies in enabling a multitude of applications such as augmented reality, autonomous driving, human computer interaction and robot guidance, thanks to their extensive integration in simultaneous localization and mapping (SLAM) [24, 32, 83], structure from motion (SfM) [86, 91], metrology [9], visual localization [75, 87] and 3D object detection [78, 105].

A myriad of papers have worked on finding the *unique* solution to the pose estimation problem [46, 47, 78, 84, 100, 101]: a pose per view/scan. However, this trend is now witnessing a fundamental challenge. A recent school of thought has begun to point out that for our highly complex and *ambiguous* real environments, obtaining a single solution i.e. the *correct pose*, is simply not sufficient. For example, an image of a scene with repeating structures can look similar even though the location and orientation of the capture device is drastically different. Likewise, objects with rotational symmetries lead to very similar point clouds when scanned from different viewpoints, say with a laser scanner. These observations have led to a paradigm shift that has opened a multitude of research directions focusing on these issues. Instead of estimating a single solution, methods now propose to predict a range of solutions providing multiple pose hypotheses [67], solutions that can associate *uncertainties* to their predictions [51, 67] or even solutions in the form of full probability distributions [2, 12, 13].

In this paper, we propose a generic data driven pose estimation algorithm that can handle non-ambiguous as well as ambiguous input data and is able to infer multiple solutions with their associated uncertainties. Specifically, depending on the input data given, we can estimate 1) a single pose, 2) multiple pose hypotheses, and in addition associating both 3) a single pose and 4) all hypotheses with a measure of

uncertainty in the prediction. Figure 1 summarizes the problems we address in this work. We further propose to capture the multiple plausible solutions of an ambiguous input in the form of a *continuous multimodal distribution* on the Riemannian manifold of poses, while explaining uncertainties by the entropy of the underlying distributions. In particular, we model rotations by a mixture of anisotropic Bingham distributions [7] that are well suited to capture the nature of quaternion parameterizations. We handle translations similarly using Gaussian distributions well suited to capture the variability in Euclidean spaces. To be more specific we extend our previous work on camera relocalization in ambiguous scenes [22] including object pose estimation in ambiguous 3D input. We begin by explaining our unimodal Bingham distribution based network, termed as *UBN* which can predict a single pose hypothesis and assign a measure of uncertainty to the prediction, but lacks the ability to model ambiguities. We then architect a multi-hypotheses prediction network similar to the one proposed by [67, 82], termed as *mixture Bingham networks* (MBN). This multi-headed network yields particle predictions spread across the posterior in order to capture different modes. Unlike [82], we additionally predict the mixture weights and variances, similar to *mixture density networks* (MDN), anchored on each mode resulting in a fully continuous Bingham mixture distribution. With a carefully designed training scheme, we largely alleviate issues such as the mode collapse attributed to MDNs, however, without resorting to a full particle scheme like [82]. We propose to train our networks with a multi-task loss that drives the network to a good optimum between capturing ambiguities and pose prediction itself. We extensively evaluated our methods on two fundamental applications – *6D camera relocalization* and *object pose estimation from point clouds*. We obtained superior results in comparison to the state-of-the-art especially when the data is inherently ambiguous. Our method is flexible in the sense that it can be used with a wide variety of backbone architectures, both for 2D images and for 3D data.

Having a continuous distribution of plausible solutions at hand is useful on multiple fronts: It allows for 1) the estimation of uncertainty [13] and provides a reliable confidence measure, 2) a direct use of the sampled solution space to characterize the 3D object symmetries or configuration of data acquisition, and 3) determining the best solution not through a naive conditional averaging but a scheme that is aware of multiple weighted modes. Note that while in 3D applications, the objects can possess many types of symmetries, unlike [26, 76] we avoid making a distinction and rather try to capture this nuance in the multimodal predictions, without explicit supervision. In conjunction with our earlier work [22], our contributions involve:

1. We provide a general framework for continuously modelling conditional density functions on quaternions us-

ing Bingham distributions, while explaining the translational uncertainty with multi-modal Gaussians when applicable. Both unimodal and multimodal models are proposed in our work and are extensively evaluated.

2. For this purpose, we devise novel ways of tailoring neural networks that fit our framework and propose effective multi-task training schemes that are well suited to the complex and non-convex posteriors we are aiming to predict. As a result, we enable an efficient optimization of the necessary distribution parameters while avoiding problems such as mode collapse and numeric instabilities existing in original Mixture Density Networks.
3. We exhaustively evaluate our methods on two fundamental problems where quantifying the pose is essential: camera relocalization and object pose estimation. We validate our approach, showing that uncertainties captured by our network correlate with the predicted rotation errors. Further, we show that ambiguities could be well handled by our deeply learned Bingham mixture model, both with regard to the quality of the single best prediction as well as the ability of capturing multiple ambiguous modes.

2 Related Work

Each of the problems summarized in Fig. 1 has posed an ongoing research question. In the following we will briefly outline recent findings for each of these categories.

Single Pose Estimation Pose estimation is a widely studied topic due to its fundamentality in many vision-based systems, such as CAD model pose estimation from images [10, 11, 49, 50, 99], object pose estimation from point clouds [78, 103], camera pose estimation [15, 17, 20, 21, 34, 52, 53, 55, 69] or pairwise pose alignment [28, 29, 94]. Apart from the correspondence based methods used in those applications [27, 100, 102], direct regression methods from a single input instance are becoming more and more popular due to their simplicity and fast deployment [29, 55, 97]. Particularly, with the advent of deep learning, powerful feature extraction methods from either images [30, 44] or 3D data [79, 80] have emerged that can pave the way to more accurate direct pose regression. Most related to our work, Kendall et al. [55] for example was the first to adopt a convolutional neural network to regress the 6D camera location and orientation from a single RGB image. Similarly, on 3D data, Deng et al. [29] learned to predict the relative poses between partial scans by regressing the rotations from pairs of local patches. Dealing with ambiguities in the context of pose estimation has so far often been handled by prior knowledge of object symmetries. The pose estimation network of Pitteri et al. [76] explicitly considered axis-symmetric objects whose pose cannot be uniquely determined. Likewise,

Corona et al. [26] addressed general rotational symmetries. All of these works require extensive knowledge about the object and cannot be extended to the scenario of localizing against a scene without having a 3D model. Note that they also cannot handle the case of self-symmetry and [26] additionally requires a dataset of symmetry-labeled objects, an assumption unlikely to be fulfilled in real applications.

Dealing with Uncertainty The above mentioned methods have shown promising results. However, so far most methods neglect that typical CNNs [44, 88] are over-confident in their predictions [41, 106]. Moreover, these networks tend to approximate the conditional averages of the target data [14]. These undesired properties render the immediate outputs of those networks unsuitable for the quantification of calibrated uncertainty, i.e. all predictions are assumed to be equally correct [29]. As a result no information of uncertainty can be provided to indicate how good or bad the predictions are. Initial attempts that address these issues and aim to capture the uncertainty of camera relocalization methods involved the use of random forests [19]. Valentin et al. [92] stored components of a Gaussian Mixture Model at the leaves of a scene coordinate regression forest [87]. The modes are obtained via a mean shift procedure on the scene coordinate samples, and the covariance is explained by a 3D Gaussian. A similar approach later considered the uncertainty in object coordinate labels [16]. A shortcoming of both of these approaches is the requirement of hand crafted depth features to train the regression forest. Moreover, their uncertainty is on the correspondences and not on the final camera pose. As a result a costly RANSAC [35] is required to propagate the uncertainty in the leaves to the camera pose.

In comparison, probabilistic methods can provide the means to directly capture the uncertainty [5] in the instance of interest. An initial attempt to capture the variability in the predictions has incorporated Monte Carlo Sampling into neural networks by activating the dropout layers commonly used in neural networks [36]. For instance, Kendall and Cipolla [52] augmented PoseNet [55] with uncertainty by sampling the posterior to approximate probabilistic inference. In comparison, Mixture Density Networks [14] directly learn to predict parameters of a Gaussian mixture distribution by using a neural network which in turn can be used to infer the uncertainty in a prediction based on the variance of the predicted distribution. Yet this method suffers from problems like mode collapse and numeric instability, and Gaussian distributions are not ideal for modeling directional data. VidLoc [25] adapted MDNs [14] to model and predict uncertainty for the 6D relocalization problem. Besides the reported issues of MDNs, VidLoc incorrectly modeled the rotation parameters using Gaussian distributions and lacked the demonstrations of uncertainty on rotations. Prokudin et al. [77] replaced the Gaussian distribution with von Mises

distribution to enable estimation of a continuous probability space applied to head pose orientations. However, only poses aligned with certain axis are able to be modeled by 2D von Mises distribution. The Bingham distribution [7], on the other hand, is found to be a good way of analyzing quaternion distributions in a full rotation space. Glover et al. [38, 39] estimated the parameters of a Bingham distribution via *Sample Consensus*. The closest to our work has been presented by Gilitschenski et al. [37] and proposes end-to-end orientation learning by incorporating the Bingham Distribution for object pose estimation from 2D images. However, the method does not yet provide any means of dealing with problems such as mode collapse commonly known to arise in MDNs.

Multiple Hypotheses Prediction In general, ambiguities arise due to the existence of multiple legit solutions. For example, in a 3D object pose estimation scenario, it can be caused by an object’s rotational symmetries [26, 76], or in a relocalization scenario, identical views acquired by cameras under different poses [54]. Many other prior works targeting ambiguities derive from the field of future prediction [63, 64]. [31, 42] proposed to generate multiple outputs as possible choices, and a *winner takes all* (WTA) strategy was proposed [42] and later widely adopted in other applications such as semantic segmentation [64]. Rupprecht et al. [82] provided a better way to understand the benefits of this branch of methods with a mathematical formulation, and a relaxation term that was introduced to WTA to facilitate convergence. In these literature, only discrete outputs are considered instead of a continuous space. To close the gap, Makansi et al. [66] learned to fit parameters of a Gaussian mixture model to the generated point hypotheses in a two-stage training scheme with a variant of WTA loss. In pose estimation, to deal with rotational symmetries and self-occlusion symmetries from visual data, Manhardt et al. [67] generate multiple quaternions as hypotheses for 6D pose estimation.

Dealing with Uncertainty and Ambiguity Few works have yet attempted to capture both multiple solutions and uncertainty prediction in the context of pose estimation. Monte Carlo sampling for example has been used to create multiple pose predictions [51]. Unfortunately even for moderate dimensions these methods still face difficulties in capturing multiple modes. In theory these methods can generate discrete samples from the multi-modal distributions. In practice, as we will demonstrate, the Monte Carlo scheme tends to draw samples around a single mode. This method also suffers from the large errors associated to PoseNet [55] itself and can not provide a measure of uncertainty for each pose hypothesis. Manhardt et al. [67] infer a measure of uncertainty from generated multiple quaternion predictions, however not for each pose hypothesis either. Further, methods

predicting a mixture of distributions in theory can capture multiple predictions. Prokudin et al. [77] for instance learn a variational auto-encoder [57] to approximate the posterior of $SO(2)$ modeled by von Mises mixtures [68] and Gilitschenski et al. [37] show that learned mixture models can aid in handling rotational ambiguities. These approaches, however, do not yet provide any measure of dealing with mode collapse that commonly arises in these type of methods, and therefore are not yet fully able to capture multiple distinct pose predictions.

In comparison our work leverages the best properties from MDN [14], WTA [67, 82] and Bingham distribution [7] to avoid problems such as mode collapse. Each unimodal Bingham distribution is treated as a single hypothesis and we aim to capture ambiguities via multiple Bingham distributions predicted by the network, without full modeling of the object symmetries or repeated structures. Eventually, ambiguities can be explained by the modes of a Bingham mixture model while the uncertainty is captured in the concentration parameters or in the *entropy*. Furthermore, we extensively evaluate our method on two applications, namely camera localization from 2D images and object pose estimation from point clouds.

3 The Bingham Distribution

We now introduce the mathematical concepts our work is based on, starting with the foundation of our work, the Bingham distribution. The Bingham distribution [7] is an antipodally symmetric probability distribution derived from a zero-mean Gaussian. It is conditioned to lie on $\mathbb{S}^{d-1}$ and its probability density function $\mathcal{B} : \mathbb{S}^{d-1} \rightarrow \mathbb{R}$ is computed as follows:

$$\mathcal{B}(\mathbf{x}; \mathbf{\Lambda}, \mathbf{V}) = (1/F) \exp(\mathbf{x}^T \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T \mathbf{x}) \quad (1)$$

$$= (1/F) \exp \left(\sum_{i=1}^d \lambda_i (\mathbf{v}_i^T \mathbf{x})^2 \right) \quad (2)$$

where $\mathbf{V} \in \mathbb{R}^{d \times d}$ is an orthogonal matrix ($\mathbf{V} \mathbf{V}^T = \mathbf{V}^T \mathbf{V} = \mathbf{I}_{d \times d}$) describing the orientation, $\mathbf{\Lambda} \in \mathbb{R}^{d \times d}$ is called the *concentration matrix* and is constrained such that $0 \geq \lambda_1 \geq \dots \geq \lambda_{d-1}$:

$$\mathbf{\Lambda} = \text{diag}([0, \lambda_1, \lambda_2, \dots, \lambda_{d-1}]) \quad (3)$$

It is easy to show that adding a multiple of the identity matrix $\mathbf{I}_{d \times d}$ to $\mathbf{V}$ does not change the distribution [7]. Thus, we conveniently force the first entry of $\mathbf{\Lambda}$ to be zero. Moreover, since it is possible to swap columns of $\mathbf{\Lambda}$, we can build $\mathbf{V}$ in a sorted fashion. This allows us to obtain the *mode* very easily by taking the first column of $\mathbf{V}$. Due to its antipodally symmetric nature, the mean of the distribution

is always zero. F in Eq. (1) denotes the *normalization constant* dependent only on Λ and is of the form:

$$F \triangleq |S_{d-1}| \cdot {}_1F_1\left(1/2, d/2, \Lambda\right), \quad (4)$$

where $|S_{d-1}|$ is the surface area of the d -sphere and ${}_1F_1$ is a confluent hypergeometric function of matrix argument [45].

Relationship to quaternions The antipodal symmetry of the probability density function $\mathcal{B}(\cdot)$ makes it amenable to explain the topology of quaternions, i. e., $\mathcal{B}(\mathbf{x}; \cdot) = \mathcal{B}(-\mathbf{x}; \cdot)$ holds for all $\mathbf{x} \in \mathbb{S}^{d-1}$. In 4D when $\lambda_1 = \lambda_2 = \lambda_3$, one can write $\Lambda = \text{diag}([1, 0, 0, 0])$. In this case, the Bingham density relates to the dot product of two quaternions $\mathbf{q}_1 \triangleq \mathbf{x}$ and the mode of the distribution, say $\bar{\mathbf{q}}_2$. This induces a metric of the form

$$d_{\text{bingham}} = d(\mathbf{q}_1, \mathbf{q}_2) = (\mathbf{q}_1 \cdot \bar{\mathbf{q}}_2)^2 = \cos(\theta/2)^2, \quad (5)$$

that is closely related to the true Riemannian distance [13] given two rotation matrices $\mathbf{R}_1$ and $\mathbf{R}_2$

$$\begin{aligned} d_{\text{riemann}} &= \|\log(\mathbf{R}_1 \mathbf{R}_2^T)\| \triangleq 2\arccos(|\mathbf{q}_1 \bar{\mathbf{q}}_2|) \\ &\triangleq 2\arccos(\sqrt{d_{\text{bingham}}}). \end{aligned} \quad (6)$$

Bingham distributions have been extensively used to represent distributions on quaternions ($\mathbb{H}_1$) [8, 13, 38, 39, 59].

Relationship to other representations Note that geometric [4] or measure theoretic [33], there are multitudes of ways of defining probability distributions on the Lie group of 6D rigid transformations [43]. A naive choice would be to define Gaussian distribution on the Rodrigues vector (or exponential coordinates) [72] where the geodesics are straight lines [71]. However, as our purpose is not tracking but direct regression, in this work we favor quaternions as continuous and minimally redundant parameterizations without singularities [23, 40] and use the Bingham distribution that is well suited to their topology. We handle the redundancy $\mathbf{q} \equiv -\mathbf{q}$ by mapping all the rotations to the northern hemisphere.

3.1 Constructing Orientation Matrices $\mathbf{V}$

Unlike Gaussian distributions whose covariance is aligned with the standard basis, constructing a Bingham distribution requires a local frame estimation to establish the orientation matrix $\mathbf{V}$. While the first component of this matrix is the *mode* as explained above, it is not clear how the other components should be computed. Additionally, ensuring this matrix is orthonormal requires care as adding a regularization term such as $\|\mathbf{V}^T \mathbf{V} - \mathbf{I}\|_F$ during optimization cannot guarantee a valid orthonormal matrix. In this work, we investigate three different ways to construct $\mathbf{V}$:

1. **Gram-Schmidt process** A straightforward way is to first estimate an unconstrained Euclidean matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ and then ortho-normalize it into $\mathbf{V}$ via Gram-Schmidt (GS) process. In this case, the column vectors $\mathbf{v}_i$ of $\mathbf{V}$ are computed from the column vectors $\mathbf{m}_i$ as follows

$$\hat{\mathbf{v}}_i = \mathbf{m}_i - \sum_{k=1}^{i-1} \langle \mathbf{v}_k, \mathbf{m}_i \rangle \cdot \mathbf{v}_k, \text{ where } \mathbf{v}_i = \frac{\hat{\mathbf{v}}_i}{\|\hat{\mathbf{v}}_i\|}. \quad (8)$$

This GS procedure requires prediction of 16 values, an over-parametrization of the degrees of freedom in $\mathbf{V}$. In the following, we refer to this process as *Gram-Schmidt (GS) Strategy*.

2. **Matrix representation** To use the minimal degrees of freedom, an elegant way proposed by Birdal [13] is to estimate the mode $\mathbf{q} \in \mathbb{H}_1$ and subsequently find a set of vectors orthonormal to $\mathbf{q}$. Fortunately, the quaternions can be linearly represented by matrices. In other words, there exists an injective homomorphisms from $\mathbb{H}_1$ to the matrix ring $M(4, \mathbb{R})$. The result is a frame bundle $\mathbb{H}_1 \rightarrow \mathcal{FH}_1$ composed of four unit basis vectors: the mode and its orthonormals:

$$\mathbf{V}(\mathbf{q}) \triangleq \begin{bmatrix} q_1 & -q_2 & -q_3 & q_4 \\ q_2 & q_1 & q_4 & q_3 \\ q_3 & -q_4 & q_1 & -q_2 \\ q_4 & q_3 & -q_2 & -q_1 \end{bmatrix}. \quad (9)$$

It is easy to verify that the matrix valued function $\mathbf{V}(\mathbf{q})$ is orthonormal for every $\mathbf{q} \in \mathbb{H}_1$. $\mathbf{V}(\mathbf{q})$ further gives a convenient way to represent quaternions as matrices paving the way to linear operations, such as quaternion multiplication or orthonormalization without the Gram-Schmidt. We refer this one as *Birdal Strategy*.

3. **Cayley transformation** Utilizing the Cayley transform, which describes a mapping from skew-symmetric matrices to special orthogonal matrices, we propose a third way to construct $\mathbf{V}$: Given the mode $\mathbf{q}$ (not necessarily with unit norm), we compute $\mathbf{V}$ as:

$$\mathbf{V} = (\mathbf{I}_{d \times d} - \mathbf{S})^{-1}(\mathbf{I}_{d \times d} + \mathbf{S}), \quad (10)$$

where $\mathbf{I}_{d \times d}$ is the identity matrix and

$$\mathbf{S}(\mathbf{q}) \triangleq \begin{bmatrix} 0 & -q_1 & q_4 & -q_3 \\ q_1 & 0 & q_3 & q_2 \\ -q_4 & -q_3 & 0 & -q_1 \\ q_3 & -q_2 & q_1 & 0 \end{bmatrix} \quad (11)$$

is a skew-symmetric matrix parameterized by $\mathbf{q}$. Similar to *Birdal* this allows us to only estimate four values and even removes the need of normalization to obtain a valid quaternion during optimization. We term this construction as *Cayley Strategy*.

Note that, for *Birdal* and *Cayley* strategies, a reduced number of predictions (4) suffice to yield $\mathbf{V}$ compared to *Gram-Schmidt* (16). We will show later in our experiments that the former two also demonstrate better performance.

4 Deep Bingham Networks (DBNs)

The Bingham distribution establishes the foundation of modelling orientations while providing the means for uncertainty estimation as well. We now further elaborate on how such a distribution can be integrated into a neural network to provide both the means for uncertainty estimation as well as to handle ambiguity issues in pose estimation problems, without sacrificing the accuracy of the single prediction. For this aim, we adopt deep neural networks and predict the underlying posterior distribution of the target pose in an end-to-end style. We consider the situation where we observe an input image $\mathbf{X} \in \mathbb{R}^{W \times H \times 3}$ or a point cloud $\mathbf{X} \in \mathbb{R}^{N \times 3}$ and assume the availability of a predictor function $\mathbf{V} \triangleq \mathbf{V}_{\Gamma}(\mathbf{X})$ parameterized by $\Gamma = \{\Gamma_i\}$. $\mathbf{V}(\cdot)$ is an orthogonal orientation matrix computed using any of the strategies introduced in Section 3.1. Note that predicting entities that are non-Euclidean easily generalizes to prediction of Euclidean quantities such as translations e.g. $\mathbf{t} \in \mathbb{R}^3$ when the constraints are removed. To this end, we investigate two models:

Unimodal Bingham Network (UBN) models the pose by a single Bingham distribution. In addition to computing a single best prediction, the entropy of the resulting distribution can be used as a measure of the uncertainty. An overview of this model is shown in Fig. 2.

Multimodal Bingham Network (MBN) predicts a multimodal Bingham distribution, as shown in Fig. 3. It extends the advantage of UBN with extra capability of capturing different modes lying in the data, thus dissolving ambiguities.

Note the definitions of our Deep Bingham Network do not include any specific network architectures, i.e. they are agnostic of the backbone networks and can be combined with any existing networks other than the reported ones in our paper.

4.1 Unimodal Bingham Network (UBN)

We now start with describing our models in more detail, beginning with our Unimodal Bingham Network. UBN takes an observation in the form of either a point cloud $\mathbf{X} \in \mathbb{R}^{N \times 3}$ or a 2D image $\mathbf{X} \in \mathbb{R}^{W \times H \times 3}$, and predicts the essential parameters of a unimodal Bingham distribution of the target pose. It describes the orientation of the object (or the camera) of interest, i.e. a single pose prediction with associated

measure of uncertainty given by the entropy of the distribution.

The first column of $\mathbf{V}$ represents the correct values of the rotation $\mathbf{q}_i \in \mathbb{H}_1$, admitting a non-ambiguous prediction, hence a posterior of single mode. We use the predicted rotation to set the most likely value (mode) of a Bingham distribution:

$$p_{\Gamma}(\mathbf{q} | \mathbf{X}; \Lambda) = (1/F) \exp(\mathbf{q}^{\top} \mathbf{V} \Lambda \mathbf{V}^{\top} \mathbf{q}), \quad (12)$$

and let $\mathbf{q}_i$ differ from this value up to the extent determined by $\Lambda = \{\lambda_i\}$. For the sake of brevity we use $\mathbf{V} \equiv \mathbf{V}_{\Gamma}(\mathbf{X})$. In this work, we model Γ by a deep neural network.

While for certain applications, fixing Λ can work, in order to capture the variation in the input, it is recommended to adapt Λ [77]. Thus, we introduce it among the unknowns. To this end we define the function $\Lambda_{\Gamma}(\mathbf{X})$ or in short Λ for computing the concentration values depending on the current input $\mathbf{X}$ and the parameters Γ . Our final model for the unimodal case reads:

$$p_{\Gamma}(\mathbf{q} | \mathbf{X}) = \frac{\exp(\mathbf{q}^{\top} \mathbf{V}_{\Gamma}(\mathbf{X}) \Lambda_{\Gamma}(\mathbf{X}) \mathbf{V}_{\Gamma}(\mathbf{X})^{\top} \mathbf{q})}{F(\Lambda_{\Gamma}(\mathbf{X}))} \quad (13)$$

$$= \frac{\exp(\mathbf{q}^{\top} \mathbf{V} \Lambda \mathbf{V}^{\top} \mathbf{q})}{F(\Lambda)} \quad (14)$$

The second row follows from the short-hand notations and is included for clarity. For the rest of this section, we stick with this simplified notations.

Inferring Bingham Parameters To produce a Bingham probability model with a network, we need to propose an end-to-end inference paradigm for Λ and $\mathbf{V}$, conditioned only on the input $\mathbf{X}$. We need to ensure all values in Λ to be non-positive and in descending order and $\mathbf{V}$ to be an orthogonal matrix.

Assume a backbone feature network is available, the original output is unconstrained and does not satisfy the properties of $\mathbf{V}$ and Λ . Further processing of those raw values is necessary. In order to keep the efforts on modifications to the lowest level, we propose only to change the last layer without adapting the remaining part of the network.

Three values are needed in Λ . In order to keep the diagonal values $(\lambda_1, \lambda_2, \lambda_3)$ sorted in **descending order** and **non-positive**, we make use of softplus activation and accumulative sum, with an extra negating operation:

$$\lambda_1 = -\phi(o_1) \quad (15)$$

$$\lambda_2 = -\phi(o_1) - \phi(o_2) \quad (16)$$

$$\lambda_3 = -\phi(o_1) - \phi(o_2) - \phi(o_3). \quad (17)$$

$\phi(\cdot)$ is the *softplus* activation function and $o_i (i = 1, 2, 3)$ are three values taken from the original output of the network.

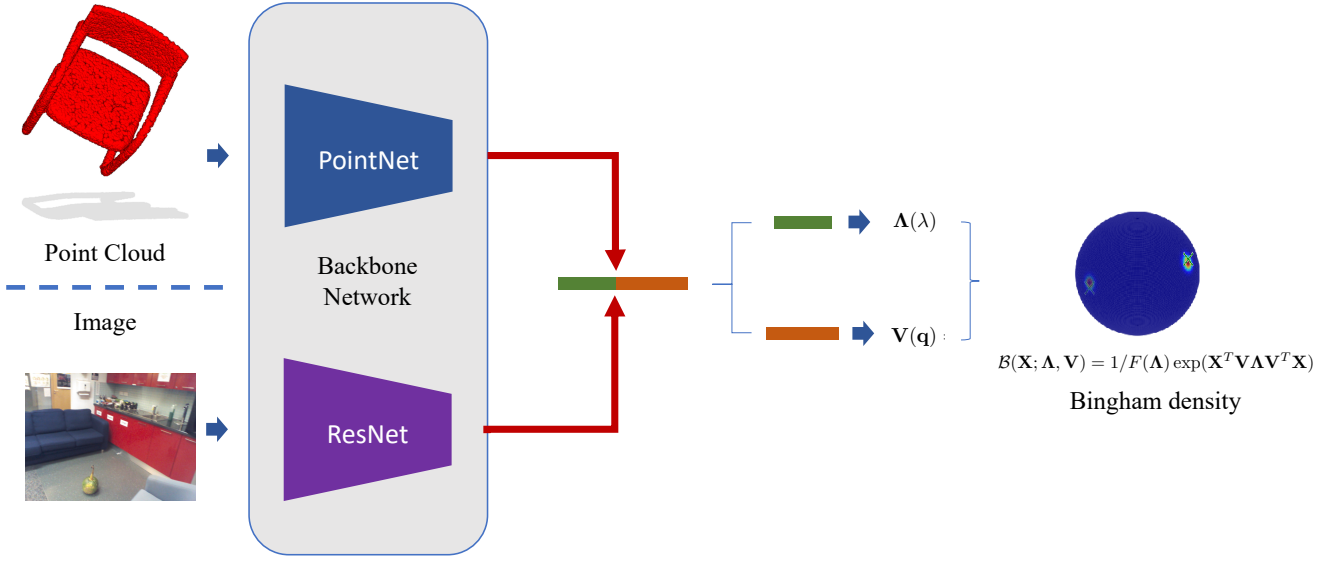


Fig. 2: The pipeline for Unimodal Bingham Network. The input data is processed by an adequate backbone network (PointNet for a rotated point cloud and ResNet for a 2D image here) to output a 7- d vector from the last layer, which is later used to form Λ and $\mathbf{V}$ of a Bingham distribution.

For constructing $\mathbf{V}$, based on the chosen strategy, 4 or 16 values are needed from the original network output. Also, for *Birdal Strategy*, we need to normalize the four values first to make it a legal quaternion to feed into Eq. (9).

The hard-to-compute entity F is shown in Eq. (4) to depend solely on Λ . To enable fast inference of F and gradient flow from Λ through F , we make use of a lookup table that is pre-computed based on a set of predefined range of values in Λ [58, 60].

Entropy as Uncertainty After obtaining the parameters of the probability function $\mathcal{B}$ of a Bingham distribution, its entropy can be computed.

$$E(\mathcal{B}) = \log(F) - \Lambda \frac{\nabla F(\Lambda)}{F} \quad (18)$$

Information theory [48] proves that higher entropy is an indicator of higher uncertainty.

Following information theory [48] and similar to [66], we treat the entropy of the predicted Bingham distribution as a practical measure of uncertainty [93]: the higher entropy the higher uncertainty. For an easier to interpret uncertainty score, we pass the entropy values through a sigmoid function mapping it to the range (0, 1):

$$U = \text{sigmoid}(E(\mathcal{B})) = \frac{1}{1 + e^{-E(\mathcal{B})}} \quad (19)$$

Bingham Loss Given a collection of observations $\mathcal{X} = \{\mathbf{X}_i\}$ and associated rotations $\mathcal{Q} = \{\mathbf{q}_i\}$, we learn the parameters Γ of our unimodal Bingham network by minimizing the negative log-likelihood:

$$\mathcal{L}(\mathbf{q}, \mathcal{B}(\Lambda, \mathbf{V})) = \log F(\Lambda) - \mathbf{q}^\top \mathbf{V} \Lambda \mathbf{V}^\top \mathbf{q} \quad (20)$$

As shown in Fig. 2, we compose $\mathbf{V} : \mathbf{V}_\Gamma(\mathbf{X})$ and $\Lambda : \Lambda_\Gamma(\mathbf{X})$ during training so as to evaluate the density. This maximizes the probability of ground truth $\mathbf{q}_i$ evaluated on the associated Bingham distribution $\mathcal{B}(\Lambda_i, \mathbf{V}_i)$. In short we obtain the optimal parameters Γ^* by

$$\Gamma^* = \arg \min_{\Gamma} \sum_{i=1}^N \mathcal{L}(\mathbf{q}_i, \mathcal{B}(\Lambda_i, \mathbf{V}_i) | \Gamma) \quad (21)$$

Note once again that Λ and $\mathbf{V}$ are dependant upon $\mathbf{X}$, thus $\Lambda_i \equiv \Lambda_\Gamma(\mathbf{X}_i)$ and $\mathbf{V}_i \equiv \mathbf{V}_\Gamma(\mu(\mathbf{X}_i))$.

4.2 Multimodal Bingham Network (MBN)

While UBN is able to grant the predictions with uncertainty information, it cannot handle ambiguities such as objects with rotational symmetries, or scenes with identical views under different camera locations. Inspired by MDN [14], we resort to multimodality and propose a Multimodal Bingham Network for predicting multiple Bingham distributions to explain a single observation $\mathbf{X}$, and eventually yielding a Bingham mixture model, as depicted in Fig. 3.

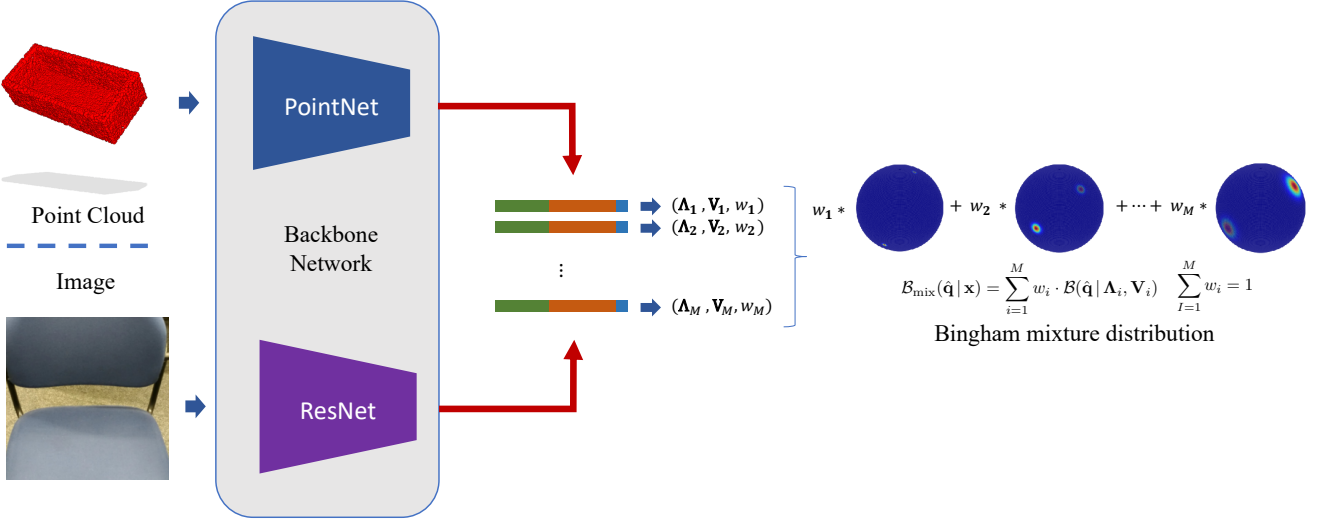


Fig. 3: The pipeline for Multimodal Bingham Network. Same network is used as in UBN, only the last layer is modified to output $M \times (7 + 1)$ units to form M groups of parameters and weights for different Bingham components. Different modes counting for ambiguity are captured by different Bingham distribution.

Bingham Mixture Model (BMM) A Bingham mixture model can be easily extended from a set of unimodal Bingham models by assigning each one a weight factor and combining them linearly to form a continuous distribution space with multimodality. Each unimodal *component* captures a *peak* presence of a valid solution. MBN then aims to predict a Bingham Mixture Model (BMM) for any given input $\mathbf{X}$, storing individual component predictions in different *branches*.

We build MBN on top of UBNs. Apart from the parameters $(\Lambda_i, \mathbf{V}_i)$ for each unimodal component, it also predicts a set of weights $\{w_i\}_{i=1}^M$. The probability density function $\mathcal{B}_{\text{mix}}$ can be written as:

$$\mathcal{B}_{\text{mix}}(\mathbf{q} | \mathbf{X}) = \sum_{i=1}^M w_i \cdot \mathcal{B}(\mathbf{q} | \Lambda_i, \mathbf{V}_i) : \sum_{i=1}^M w_i = 1 \quad (22)$$

M is the number of components. This way, different poses associated with the same input $\mathbf{X}$ can acquire high probabilities in different components.

Mixture Bingham Loss Similar to Eq. (21), we can define a *Mixture Bingham Loss* as the negative log-likelihood of the mixture distribution model, which can be viewed as a weighted sum of *Bingham losses* per each component.

$$\mathcal{L}_{\text{MB}}(\mathbf{q}, \mathcal{B}_{\text{mix}}) = -\log(\mathcal{B}_{\text{mix}}(\mathbf{q} | \mathbf{X})) \quad (23)$$

Previous work on Mixture Density Network [14, 66] has shown that directly optimizing all the parameters at the same time can lead to numerical instabilities and problems such as mode collapse might arise. Thus a proper solution to tackle these issues is critical.

RWTA Loss Our current model easily suffers from mode collapse, i.e. it is not able to capture multiple distinct modes well. However, to efficiently handle ambiguities predicting plausible distinct solutions is of importance. With this additional property included into our model, we would be able to capture multiple solutions as well as associated uncertainties for each prediction. We therefore incorporate a “Winner Takes All” training scheme that has been proven to be effective for coping with ambiguities e.g. multiple modes [67, 82]. In WTA, each iteration updates only the branch that generates the closest prediction to the ground truth. This provably leads to the Voronoi tessellation of the output space.

Let $\mathcal{B}_i$ be the i -th component of our Bingham Mixture Model. At each training iteration, we select the component giving the best prediction with regard to the current ground truth. We could select the best component by checking the l_1 distances of the predicted quaternion $\hat{\mathbf{q}}$ with regard to the given ground truth $\mathbf{q}$ [67].

$$i^* = \arg \min_i |\mathbf{q} - \hat{\mathbf{q}}_i| \quad (24)$$

An alternative way is to check the probabilities of the ground truth $\mathbf{q}$ on the set of predicted Bingham distributions, and keep the one with highest value.

$$i^* = \arg \max_i \mathcal{B}_i(\mathbf{q} | \Lambda_i, \mathbf{V}_i) \quad (25)$$

Then we optimize $(\Lambda_i, \mathbf{V}_i)$ of $\mathcal{B}_i$ by minimizing its corresponding Bingham Loss. Rupprecht et al. [82] find that allowing a small portion of gradients from the sum of losses would help avoid “dead” particles which never get updated because of their bad random initializations, which can be considered as a *relaxed* version of WTA.

$$\mathcal{L}_{\text{RWTa}}(\mathbf{q}, \mathcal{B}_{\text{mix}}) = \sum_{i=1}^M \pi_i \mathcal{L}(\mathbf{q}, \mathcal{B}_i(\mathbf{A}_i, \mathbf{V}_i)) \quad (26)$$

$$\pi_i = \begin{cases} 1 - \epsilon & \text{if } i\text{-th component is selected} \\ \frac{\epsilon}{M-1} & \text{else} \end{cases} \quad (27)$$

$\mathcal{L}_{\text{RWTa}}$ is able to guide the multiple unimodal components to cover different modes of the ground truth distribution. Yet, it cannot represent a continuous distribution due to the lack of variance predictions and the mixture weights $\{w_i\}$.

Cross Entropy Loss We provide an alternative way to explicitly train the weights for the components in MBN by forming it as a classification problem.

$$\mathcal{L}_{\text{CE}}(w_{\mathcal{B}_{\text{mix}}}) = \sum_{i=1}^M -(y_i \cdot \log(w_i) + (1 - y_i) \cdot \log(1 - w_i)), \quad (28)$$

w_i is the predicted weight for the i -th component and y_i is the associated label given by the selection results by either Eq. (24) or Eq. (25).

$$y_i = \begin{cases} 1, & \text{if } i = i^* \\ 0, & \text{otherwise} \end{cases}. \quad (29)$$

Based on those loss functions, we propose two independent training schemes:

- **Cross Entropy + RWTa.** Following [22], it relies on RWTa loss to train each individual components for capturing different modes and Cross Entropy loss to assign proper weights for each one with

$$\mathcal{L}_{\text{MBN-CE}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{RWTa}}. \quad (30)$$

- **Mixture Bingham + RWTa.** It follows the conventional MDN training [14] scheme, but imports extra RWTa loss to help guide the training to overcome the existing problems resulting in the following loss function

$$\mathcal{L}_{\text{MBN}} = \mathcal{L}_{\text{MB}} + \mathcal{L}_{\text{RWTa}}. \quad (31)$$

We will show later in our experiments that both schemes could facilitate the training process as well as further improve the performance. To differentiate, we name the MBNs trained with the two schemes as *MBN* and *MBN-CE* respectively. Note that by learning the weights of the mixture model, we can always also find a single best prediction by utilizing the mode associated to the most likely mixture component.

5 Application to Camera Re-localization

We first evaluate our method on the task of re-localizing a camera in a given reference scene, before demonstrating our method's performance in predicting the pose of an object from a given input point cloud.

5.1 Modelling translations

As a camera's pose is defined by its orientation as well as its position, we predict the rotation by our proposed Deep Bingham Network. Further, as they reside in the Euclidean space, we use mixture density networks [14] to incorporate translations. In more detail, for a sample input image $\mathbf{X} \in \mathbb{R}^{W \times H \times 3}$, we obtain a predicted translation $\hat{\mathbf{t}} \in \mathbb{R}^{c=3}$ from a neural network with parameters $\mathbf{\Gamma}$. This prediction is set to the most likely value of a multivariate Gaussian distribution with covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_c^2 \end{bmatrix}_{c \times c}, \quad (32)$$

where σ^2 is predicted by our model. As a result our model for a unimodal Gaussian is defined as:

$$p_{\mathbf{\Gamma}}(\mathbf{t} | \mathbf{X}) = \frac{\exp(-\frac{1}{2}(\mathbf{t} - \hat{\mathbf{t}})^{\top} \mathbf{\Sigma}^{-1}(\mathbf{t} - \hat{\mathbf{t}}))}{(2\pi)^{c/2} |\mathbf{\Sigma}|^{1/2}}, \quad (33)$$

where $c = 3$ and both $\hat{\mathbf{t}}$ as well as $\mathbf{\Sigma}$ are trained by minimizing its negative log-likelihood.

Similar to forming a Bingham Mixture Model, we can equally compute a Gaussian Mixture Model with M components and corresponding weights $\{w_i\}$, such that $\sum_{i=1}^M w_i = 1$, to obtain a multi-modal solution. Again both $\hat{\mathbf{t}}$ and $\mathbf{\Sigma}$ as well as $\{w_i\}$ are learned by the network and trained by minimizing the negative log-likelihood of the mixture model. Note that, in this case, the components of $\hat{\mathbf{t}}$ are assumed to be statistically independent within each distribution component. However, it has been shown that any density function can be approximated up to a certain error by a multivariate Gaussian mixture model with underlying kernel function as defined in Eq. (33) [14, 70]. In practice we first train our network for the translation and its variance. We then train for the rotation and its distribution parameters, which intuitively, after knowing the position, should be an easier task. Finally we fine-tune the entire network for all distribution parameters.

Similar to our Bingham model we use the entropy of the resulting distribution as a measure of uncertainty:

$$E(G) = \frac{c}{2} + \frac{c}{2} \log(2\pi) + \frac{1}{2} \log(|\mathbf{\Sigma}|), \quad (34)$$

respectively, where $c = 3$ the dimension of the mean vector of the Gaussian. For a given image we first normalize the entropy values over all pose hypothesis, and finally obtain a measure of (un)certainly for a camera pose hypothesis as the sum of both rotational ($E(B)$, see Eq. (18)) and translational ($E(G)$) normalized entropy.

5.2 Implementation Details

We implement our method in Python using the PyTorch library [73]. Following the current state-of-the-art direct camera pose regression methods, we use a *ResNet-34* [44] as our backbone network architecture, followed by fully-connected layers for rotation and translation, respectively. In the following we use the strategy of *Birdal* to construct $\mathbf{V}$ and normalize the predicted quaternions during training. Ablation studies on further construction methods are presented in section 7. All models are trained with the ADAM optimizer with an exponential learning rate decay and trained for 300 epochs with a batch size of 20 images.

5.3 Experimental Setup for 6D Relocalization

When evaluating our method we consider two cases: (1) camera relocalization in non-ambiguous scenes, where our aim is to not only predict the camera pose, but the posterior of both rotation and translation that can be used to associate each pose with a measure of uncertainty; (2) we create a highly ambiguous environment, where similar looking images are captured from very different viewpoints. We show the problems current regression methods suffer from in handling such scenarios and in contrast show the merit of our proposed method.

Error metrics Given a ground truth camera pose, consisting of a rotation, represented by a quaternion $\mathbf{q}$, and its translation, $\mathbf{t}$, we evaluate the performance of our models with respect to the accuracy of the predicted camera poses by computing the recall of ours and the baseline models. We consider a camera pose estimate to be correct if both rotation and translation are below a pre-defined threshold and compute the angular error between ground truth, $\mathbf{q}$, and predicted quaternion, $\hat{\mathbf{q}}$, as

$$d_q(\mathbf{q}, \hat{\mathbf{q}}) = 2 \arccos(|\mathbf{q} \circ \hat{\mathbf{q}}|). \quad (35)$$

For translations we use the norm of the difference between ground truth $\mathbf{t}$, and predicted translation $\hat{\mathbf{t}}$: $d_t(\mathbf{t}, \hat{\mathbf{t}}) = \|\mathbf{t} - \hat{\mathbf{t}}\|_2$ to compute the error in position of the camera.

We obtain a single prediction from our network by taking the weighted mode, the mode of the distribution with highest mixture coefficient. Note that, under ambiguities a best mode is unlikely to exist. In those cases, as long as we can generate a hypothesis that is close to the ground truth, our network is considered successful. For this reason, in addition to the standard metrics and the weighted mode, we will also speak of the so called *Oracle* error, assuming an oracle that is able to choose the best of all predictions: the one closest to the ground truth. In addition, we report the *Self-EMD* (SEMD) [66], the earth movers distance [81] of

turning a multi-modal distribution into a unimodal one. With this measure we can evaluate the diversity of predictions. We choose the predicted mode, the unimodal distribution of the weighted mode, as reference for this measure. Note that this metric itself does not give any indication about the accuracy of the prediction, but is used as a measure of diversity in our predictions.

Datasets We first evaluate on the standard datasets of 7-Scenes [87] and Cambridge Landmarks [55] that have both been widely used to evaluate camera localization methods. Both datasets consist of RGB frames with associated ground truth camera poses and provide training as well as test sequences. In addition, we created synthetic as well as real datasets, that are specifically designed to contain repetitive structures and allow us to assess the real benefits of our approach in ambiguous environments. For synthetic data we render table models from 3D Warehouse¹ and create camera trajectories, such that ambiguous views are ensured to be included in our dataset. In particular we create a circular movement around the object. Specifically we use a *dining table* and a *round table* model with discrete modes of ambiguities. In addition, we create highly ambiguous real scenes using Google Tango and the graph-based SLAM approach RTAB-Map [61]. We acquire RGB and depth images as well as distinct ground truth camera trajectories for training and testing. We also obtain a reconstruction of those scenes. However, note that only the RGB images and corresponding camera poses are required to train our model and the reconstructions are used for visualization only. In particular, our training and test sets consist of 2414 and 1326 frames, respectively. Note that our network sees a single pose label per image.

Baselines and state of the art We compare our approach to current direct camera pose regression methods, PoseNet [53] and MapNet [18], that regress a single pose prediction with neural networks. More importantly, we assess our performance against two state-of-the-art approaches, namely BayesianPoseNet [51] and VidLoc [25], that are most related to our work and predict a distribution over the pose space by using dropout and mixture density networks, respectively. We further include the unimodal predictions of UBN, as well as BMMs, MBN-MB, trained using mixture density networks [14, 37] as baseline models.

5.4 Evaluation in non-ambiguous scenes

We first evaluate our method on the publicly available 7-Scenes [87] and Cambridge Landmarks [55] datasets. As most of the scenes contained in these datasets do not show

¹ <https://3dwarehouse.sketchup.com/>

Table 1: Evaluation in non-ambiguous scenes, displayed is the median rotation and translation error. (Numbers for MapNet on the Cambridge Landmarks dataset are taken from [85]). BPN depicts Bayesian-PoseNet [18]. *UBN* and *MBN-MB* refer to our unimodal version and mixture model respectively.

Dataset [° / m]	7-Scenes							Cambridge Landmarks				
	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Kings	Hospital	Shop	St. Marys	Street
PoseNet	4.48/0.13	11.3/0.27	13.0/0.17	5.55/0.19	4.75/0.26	5.35/0.23	12.4/0.35	1.04/0.88	3.29/3.2	3.78/0.88	3.32/1.57	25.5/20.3
MapNet	3.25/0.08	11.69/0.27	13.2/0.18	5.15/0.17	4.02/0.22	4.93/0.23	12.08/0.3	1.89/1.07	3.91/1.94	4.22/1.49	4.53/2.0	-
BPN	7.24/0.37	13.7/0.43	12.0/0.31	8.04/0.48	7.08/0.61	7.54/0.58	13.1/0.48	4.06/1.74	5.12/2.57	7.54/1.25	8.38/2.11	-
VidLoc	-0.18	-0.26	-0.14	-0.26	-0.36	-0.31	-0.26	-	-	-	-	-
UBN	4.97/0.1	12.87/0.27	14.05/0.12	7.52/0.2	7.11/0.23	8.25/0.19	13.1/0.28	1.77/0.88	3.71/1.93	4.74/0.8	6.19/1.84	24.08/16.8
MBN-MB	4.35/0.1	11.86/0.28	12.76/0.12	6.55/0.19	6.9/0.22	8.08/0.21	9.98/0.31	2.08/0.83	3.64/2.16	4.93/0.92	6.03/1.37	36.88/9.69

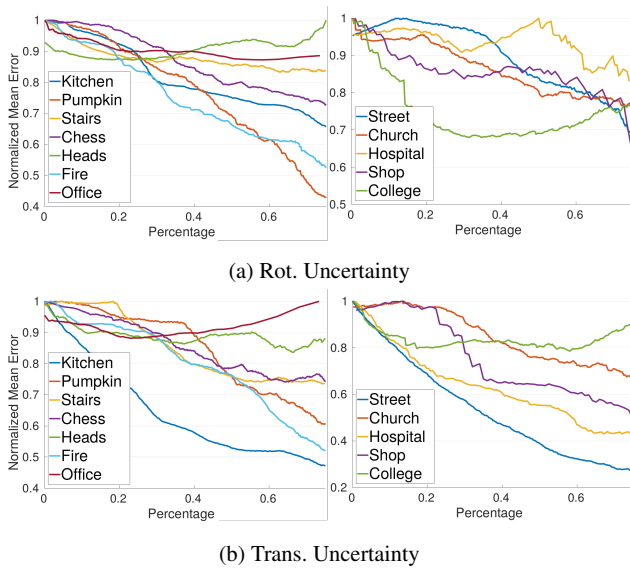


Fig. 4: Uncertainty evaluation on the 7-Scenes and Cambridge Landmarks datasets, showing the correlation between predicted uncertainty and pose error. Based on the entropy of our predicted distribution uncertain samples are gradually removed. We observe that as we remove the uncertain samples the overall error drops indicating a correlation between our predictions and the actual erroneous estimations on most scenes.

highly ambiguous environments, we consider them to be non-ambiguous. Though, we can not guarantee that some ambiguous views might arise in these datasets as well, such as in the *Stairs* scene of the 7-Scenes dataset. Both datasets have extensively been used to evaluate camera pose estimation methods. Following the state of the art, we report the median rotation and translation errors, the results of which can be found in Table 1. In comparison to methods that output a single pose prediction (e.g. PoseNet [53] and MapNet [18]), our methods achieves similar results. Yet, our network provides an additional piece of information that is the uncertainty. On the other hand, especially in translation our

method outperforms uncertainty methods, namely Bayesian-PoseNet [51] and VidLoc [25], on most scenes.

Uncertainty evaluation One benefit of our method is that we can use the resulting variance of the predicted distribution as a measure of uncertainty in our predictions. The resulting correlation between pose error and uncertainty can be seen in Fig. 4, where we gradually remove the most uncertain predictions and plot the mean error for the remaining samples. The strong inverse correlation between the actual errors vs our confidence on most scenes shows that whenever our algorithm labels a prediction as uncertain it is also likely to be a bad estimate.

It has been shown that current direct camera pose regression methods still have difficulties in generalizing to views that differ significantly from the camera trajectories seen during training [85]. However, as we will show in the experiments section of our paper, these methods in addition suffer from ambiguities arising in the scene. Therefore, we analyze the performance of direct regression methods in a highly ambiguous environment. In this scenario even similar trajectories can confuse the network and easily lead to wrong predictions, for which our method proposes a solution.

5.5 Evaluation in ambiguous scenes

We start with quantitative evaluations on our synthetic as well as real scenes before showing qualitative results of our and the baseline methods. As recent results suggest that the ResNet network architecture is more effective in direct camera pose regression methods [18], we exchange the initially used GoogleNet architecture with a ResNet for the state-of-the-art methods. In the following, we thus refer to Bayesian-PoseNet as MC-Dropout.

5.5.1 Explicit mixture coefficient learning

We first evaluate explicit learning of the mixture coefficients and present results of MBN-CE, earlier introduced in [22],

Table 2: Ratio of correct poses on our ambiguous scenes for several thresholds. We report the results of our MBN as MBN- M , where M is the number of hypotheses used.

	Threshold	PoseNet [55]	MC-Dropout [51]	UBN	MBN-MB	MBN-CE	MBN-5	MBN-10	MBN-25	MBN-50
Blue Chairs (A)	10° / 0.1m	0.19	0.39	0.29	0.24	0.35	0.40	0.48	0.35	0.39
	15° / 0.2m	0.69	0.78	0.73	0.75	0.81	0.85	0.92	0.80	0.79
	20° / 0.3m	0.90	0.88	0.86	0.80	0.82	0.89	0.96	0.87	0.85
Meeting Table (B)	10° / 0.1m	0.0	0.04	0.02	0.01	0.05	0.03	0.07	0.08	0.03
	15° / 0.2m	0.05	0.13	0.12	0.07	0.28	0.26	0.33	0.31	0.32
	20° / 0.3m	0.10	0.22	0.19	0.10	0.39	0.34	0.42	0.38	0.41
Staircase (C)	10° / 0.1m	0.14	0.13	0.11	0.04	0.18	0.20	0.18	0.18	0.17
	15° / 0.2m	0.45	0.32	0.48	0.15	0.50	0.47	0.47	0.47	0.49
	20° / 0.3m	0.60	0.49	0.62	0.25	0.68	0.64	0.66	0.64	0.64
Staircase Extended (D)	10° / 0.1m	0.07	0.02	0.06	0.06	0.09	0.10	0.06	0.09	0.11
	15° / 0.2m	0.31	0.14	0.26	0.21	0.39	0.43	0.38	0.44	0.46
	20° / 0.3m	0.49	0.31	0.41	0.32	0.58	0.59	0.60	0.62	0.64
Seminar Room (E)	10° / 0.1m	0.37	0.18	0.11	0.06	0.35	0.39	0.38	0.35	0.37
	15° / 0.2m	0.81	0.58	0.36	0.23	0.83	0.77	0.78	0.84	0.80
	20° / 0.3m	0.90	0.78	0.57	0.40	0.95	0.88	0.93	0.94	0.92
Average	10° / 0.1m	0.15	0.15	0.12	0.08	0.20	0.23	0.24	0.21	0.22
	15° / 0.2m	0.46	0.39	0.39	0.28	0.56	0.56	0.58	0.57	0.57
	20° / 0.3m	0.60	0.54	0.53	0.37	0.68	0.67	0.71	0.69	0.69

Table 3: SEMD of our method and MC-Dropout indicating highly diverse predictions by our method in comparison to the baseline. Capital letter refer to the scenes of our ambiguous relocation dataset with A: Blue Chairs, B: Meeting Table, C: Staircase, D: Staircase Extended and E: Seminar Room.

Method/Scene	A	B	C	D	E
MC-Dropout	0.06	0.11	0.13	0.26	0.10
MBN-CE	1.19	2.13	2.04	3.81	1.70
MBN	1.20	2.53	2.24	4.35	2.22

before providing additional evaluations of MBN, as proposed in this paper.

Quantitative evaluations We specifically created our synthetic dataset to contain a discrete set of modes such that we can easily identify ambiguous views. In particular, we know that there are two and four possible modes for each image in the *dining* and *round* table scenes respectively. Hence, to analyze the predictions of our model and its ability to avoid mode collapse we compute the accuracy of correctly detected modes of the true posterior. A mode is considered as found if there exists one pose hypothesis that falls into a certain rotational (5°) and translational (10% of the diameter of ground truth camera trajectory) threshold of it. In the dining-table scene, we observe that MC-Dropout obtains an accuracy of 50%, finding one mode for each image, whereas the accuracy of MBN-CE on average achieves

Table 4: Ratio of correctly detected modes for various translational thresholds (in meters). A refers to the *Blue Chairs* scene, whereas B stands for the *Meeting Table* scene.

Scene	Method	0.1	0.2	0.3	0.4
A	MC-Dropout	0.11	0.15	0.16	0.16
	MBN-CE	0.36	0.79	0.80	0.80
	MBN	0.42	0.76	0.77	0.77
B	MC-Dropout	0.04	0.07	0.09	0.11
	MBN-CE	0.10	0.43	0.63	0.73
	MBN	0.12	0.41	0.60	0.73

96%. On the round-table scene, our model shows an average detection rate of 99.1%, in comparison to 24.8% of MC-Dropout, which suffers from mode collapse and even though it is able to predict multiple hypotheses, they tend to reside around one particular mode. On our real scenes, we report the recall, where a pose is considered to be correct if both the rotation and translation errors are below a pre-defined threshold. Thanks to the diverse mode predictions of our MBN-CE, which is indicated by the high SEMD shown in Table 3, we are able to improve upon our baselines predictions. Further, by a semi-automatic labeling procedure, we are able to extract ground truth modes for the *Blue Chairs* and *Meeting Table* scenes. For that aim, we train an autoencoder on reconstructing the input images and use the resulting feature descriptors to obtain the nearest neighbor camera poses. Then, we cluster the resulting camera poses using a Riemannian Mean Shift algorithm [89] and use the

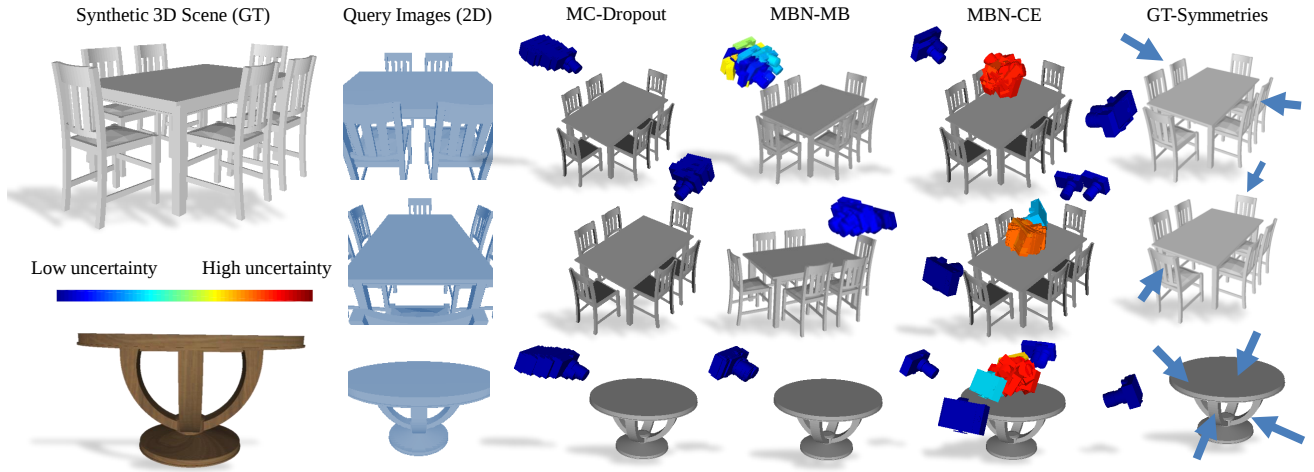


Fig. 5: Qualitative results on our synthetic *dining* and *round table* datasets. Camera poses are colored by uncertainty. View-points are adjusted for best perception.

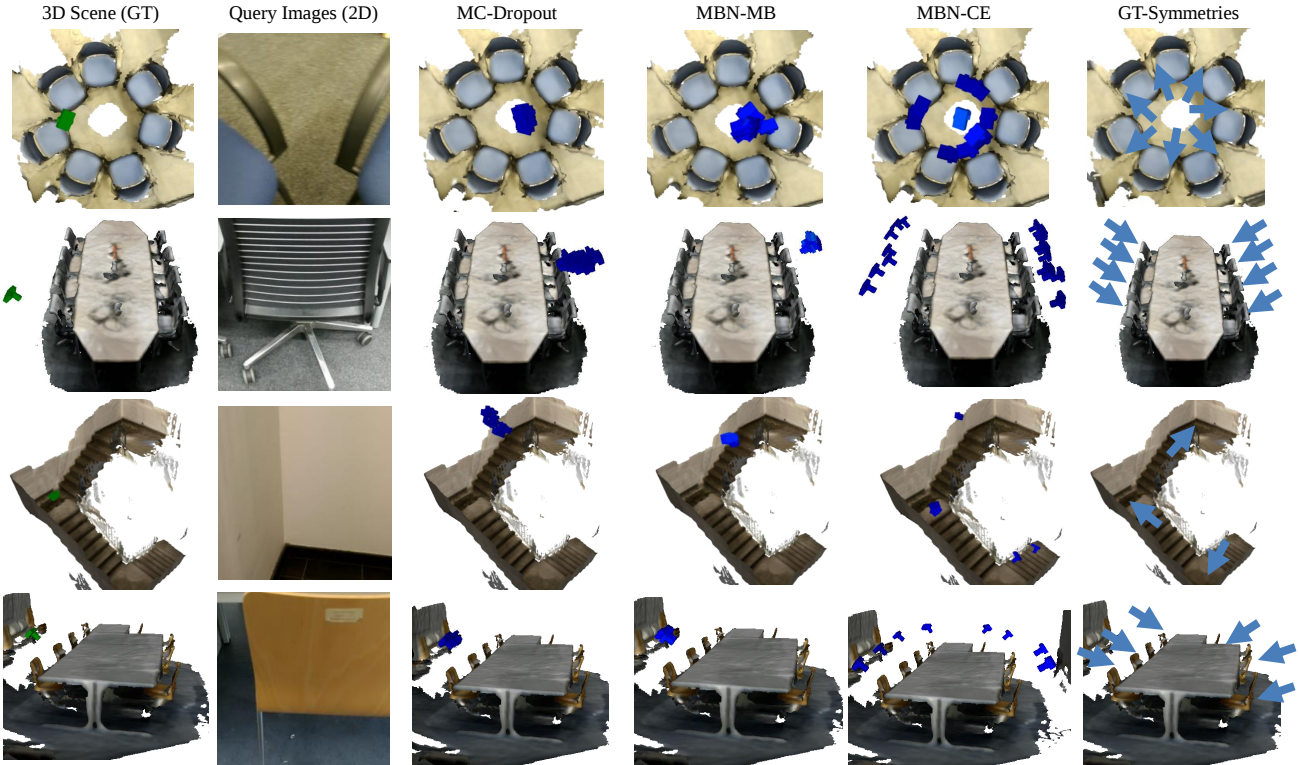


Fig. 6: Qualitative results in our ambiguous relocation dataset. For better visualization, camera poses have been pruned by their uncertainty values.

centroids of the resulting clusters as "ground truth" modes. We visually verify the results. This way, we can evaluate the entire set of predictions against the ground truth. Table 4 shows the percentage of correctly detected modes for our method in comparison to MC-Dropout when evaluating with these ground truth modes. The results again support our

findings, that MC-Dropout suffers from mode collapse, such that even with increasing threshold the number of detected modes does not increase significantly.

Qualitative evaluations Qualitative results of our proposed model on our synthetic table datasets are shown in Fig. 5.

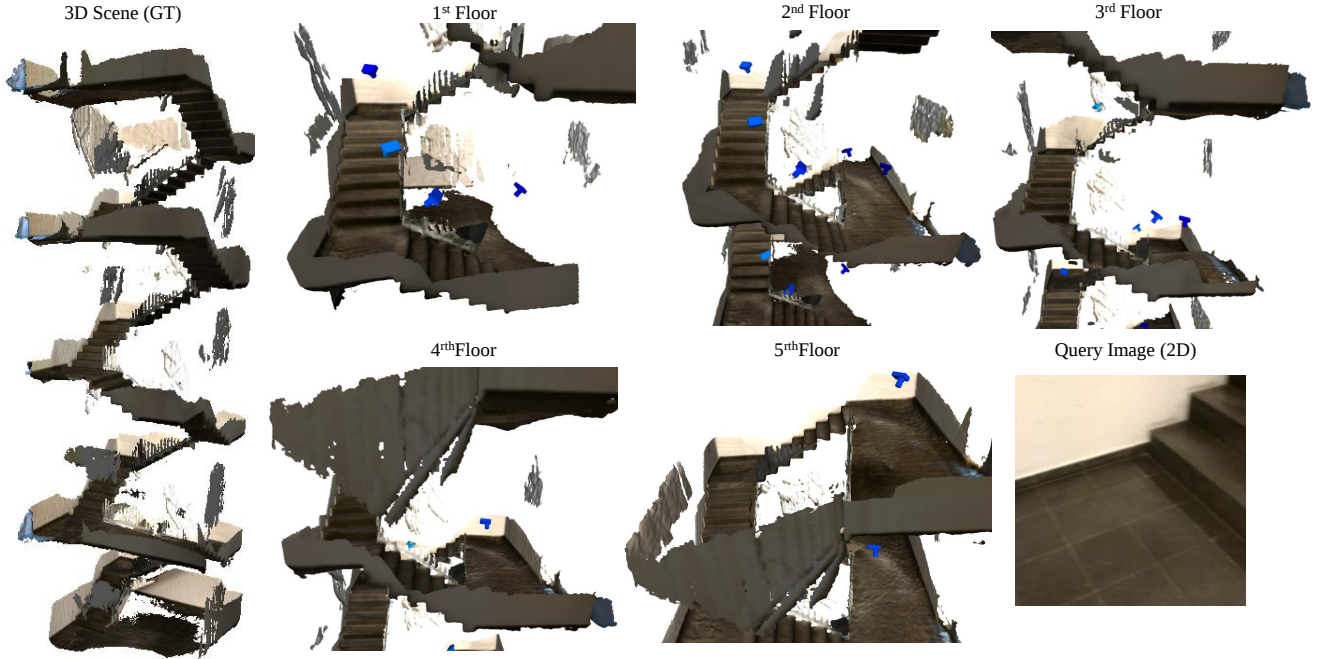


Fig. 7: Qualitative results of our model on the *Staircase Extended* scene. We show a reconstruction of the scene as well as predicted camera poses for a given query image.

MC-Dropout as well as our finite mixture model, *MBN-MB*, suffer from mode collapse. In comparison, the proposed MHP model is able to capture plausible, but diverse modes as well as associated uncertainties. In contrast to other methods that obtain an uncertainty value for one prediction, we obtain uncertainty values for each hypothesis. This way, we could easily remove non-meaningful predictions, that for example can arise in the WTA and RWTa training schemes.

Fig. 6 shows qualitative results on our ambiguous real scenes. Again, MC-Dropout and MBN-MB suffer from mode collapse. Moreover, these methods are unable to predict reasonable poses given highly ambiguous query images. That is most profound in our *Meeting Table* scene, where due to its symmetric structure the predicted camera poses fall on the opposite side of the ground truth one.

5.5.2 Implicit mixture coefficients learning

We now evaluate our extended method, MBN, that allows for implicit learning of the mixture coefficients without succumbing to the pitfalls of ordinary mixture density networks such as mode collapse. Table 2 shows the accuracy of our baseline methods in comparison to ours for various thresholds. Especially on our *Meeting Table* scene, it can be seen that the performance of direct camera pose regression methods that suffer from mode collapse drops significantly due to the presence of ambiguities in the scene. In addition, we are able to improve upon our baseline model, MBN-CE, in overall performance, as well as in SEMD as reported in Table 3.

Table 5: Average ratio of correct oracle poses on our ambiguous relocalization scenes for several thresholds and numbers of predicted pose hypothesis M , indicated as Oracle- M .

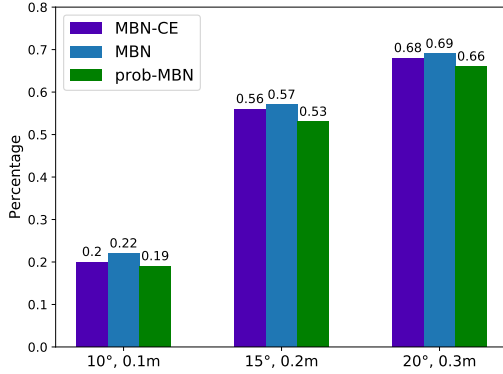
Threshold	MC-Dropout Oracle	MBN- Oracle-5	MBN- Oracle-10	MBN- Oracle-25	MBN- Oracle-50
10° / 0.1m	0.28	0.26	0.27	0.27	0.26
15° / 0.2m	0.60	0.56	0.60	0.67	0.67
20° / 0.3m	0.70	0.69	0.75	0.80	0.84

Further, in comparison to MC-Dropout our model is able to provide diverse predictions and capture multiple modes, which is indicated by the high Oracle accuracy, see Table 5.

Backbone network To evaluate the effect of different network architectures on our model, we change the backbone network of ours and the state-of-the-art baseline methods. As most of the recent state-of-the-art image based localization methods [3, 18, 74] use a version of ResNet, we compare between ResNet variants: ResNet-18, ResNet-34 and ResNet-50 and Inception-v3 [90]. All networks are initialized from an ImageNet [30] pre-trained model. We report our findings in Table 6. When comparing the performance of different ResNet variants all networks showed on average similar accuracy. Overall, naturally all methods are slightly dependant on the features that serve as input to the final pose regression layers. However, regardless of the backbone net-

Table 6: Averaged ratio of correct poses for different backbone networks over all scenes of our ambiguous relocalization dataset.

	Threshold	PoseNet	UBN	MBN-MB	MC-Dropout	MBN-CE	MBN
ResNet-34	10° / 0.1m	0.15	0.12	0.08	0.15	0.20	0.22
	15° / 0.2m	0.46	0.39	0.28	0.39	0.56	0.57
	20° / 0.3m	0.60	0.53	0.37	0.54	0.68	0.69
ResNet-18	10° / 0.1m	0.15	0.16	0.09	0.15	0.19	0.20
	15° / 0.2m	0.47	0.42	0.29	0.39	0.52	0.53
	20° / 0.3m	0.60	0.54	0.39	0.54	0.66	0.64
ResNet-50	10° / 0.1m	0.20	0.15	0.10	0.15	0.20	0.20
	15° / 0.2m	0.49	0.36	0.30	0.40	0.55	0.52
	20° / 0.3m	0.62	0.53	0.38	0.53	0.69	0.67
Inception-v3	10° / 0.1m	0.11	0.10	0.11	0.08	0.18	0.17
	15° / 0.2m	0.38	0.33	0.38	0.31	0.49	0.50
	20° / 0.3m	0.55	0.53	0.52	0.49	0.63	0.63

Fig. 8: Influence on the choice of the best hypothesis on our MHP training scheme. We compare between MBN-CE and MBN (l_1 loss) and prob-MBN, i.e. choosing the best branch according to its probability (see Eq. (25)).

work used, MBN-CE and MBN show, on average, superior performance over the baseline methods.

5.6 Choice of best branch

The choice of the best branch in multiple hypothesis predictions depends on the chosen distance function comparing the prediction to the ground truth label.

In this work, we compare between the l_1 norm (see Eq. (24)) and choosing the branch with highest probability density, as described in Eq. (25), and report the results in Fig. 8. For our specific application, camera re-localization, we have found l_1 to outperform probability based choices.

6 Application to Point Cloud Pose Estimation

We continue to apply our Deep Bingham Networks on the task of class-level point cloud pose estimation. We assume that the class of the test object is known and an individual network is trained for each class. The pose of a point cloud can be expressed by a combination of rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$. For 3D objects, translation can be canceled out using the centroid or a common anchor, whereas the rotation has to be estimated, which is also the main source of ambiguities in this application. Similarly, the latter quantity is parameterized by a unit quaternion $\hat{\mathbf{q}} \in \mathbb{H}_1$.

6.1 Implementation and Training Details

Our Deep Bingham Networks and losses are implemented using Pytorch [73] and we use PointNet [79] as the backend to process point clouds. We train each class for 500 epochs and in each epoch 100 quaternions are randomly sampled to rotate the training objects. We set the learning rate at 0.001 and use Adam solver [56] to optimize the network parameters.

Birdal Strategy is used as the default way of constructing $\mathbf{V}$. The default training loss for MBN is a combination of *Mixture Bingham Loss* and *RWTA Loss* and the best component for computing *RWTA Loss* is chosen by probability.

6.2 Experiment Setup

Baselines We extensively studied the state-of-the-art methods on pose/rotation estimation, however, due to the differences in the specific targeted applications, such as camera localization [53], point cloud registration [94] or object pose

Table 7: Point cloud pose estimation results on ModelNet10. The values are scaled by 10^2 .

	L1	Cosine	Ploss	PointNetLK	IT-Net	MC-Dropout	UBN	MBN-5	MBN-10	MBN-25	MBN-50	MBN-CE	MBN-MB
Bathtub	4.126	7.141	2.262	10.110	11.015	5.994	1.064	0.728	0.557	0.490	0.504	0.790	0.805
Bed	1.815	3.049	1.610	12.330	7.106	1.907	0.918	0.331	0.235	0.267	0.332	0.790	0.656
Chair	0.653	1.108	0.859	8.280	3.272	0.866	0.999	0.734	0.600	0.727	0.663	0.790	0.970
Desk	6.101	8.190	4.529	10.730	11.299	6.068	3.190	2.129	1.953	2.615	2.656	2.620	2.510
Dresser	4.524	6.753	2.888	7.260	8.500	5.124	2.285	1.423	1.372	1.669	1.771	2.620	2.166
Monitor	3.005	5.547	2.172	13.230	11.184	2.980	2.038	1.009	0.988	0.984	1.169	1.150	1.243
Night Stand	3.661	4.144	2.987	5.700	6.348	3.535	1.943	1.602	1.282	1.248	1.281	1.600	2.066
Sofa	0.727	1.368	0.786	12.460	3.763	0.820	0.620	0.314	0.327	0.352	0.408	0.300	0.444
Table	10.825	16.820	1.253	16.550	15.537	7.063	0.871	0.428	0.519	0.506	0.566	0.780	0.740
Toilet	0.609	1.389	0.582	7.430	3.746	0.730	0.846	0.576	0.544	0.401	0.377	0.570	0.769
Average	3.605	5.551	1.993	10.410	8.180	3.509	1.477	0.927	0.838	0.926	0.973	1.201	1.237

estimation [67], and the accordingly varying network selections, it is very difficult to obtain a fair direct comparison. Therefore, we evaluated the loss functions commonly employed by the state of the art:

- **L1 loss.** The most popular one is L_p ($p = 1, 2$) loss, and it has been most widely used in recent work for rotation regression [1, 53, 62, 94, 95]. Previous work has found that L_1 outperforms L_2 in general [53], so we mainly take L_1 loss for comparison: $\mathcal{L}_1 = \|\mathbf{q} - \hat{\mathbf{q}}\|_1$.
- **Cosine loss.** A metric often used for measuring distance on the spherical manifold, cosine loss respects the vectorial nature of quaternions [65]: $\mathcal{L}_{\cos} = 1 - |\mathbf{q} \cdot \hat{\mathbf{q}}|$.
- **Ploss.** Defined on the points rather than the rotations themselves, Ploss [97, 98] yields point-wise Euclidean distances: $\mathcal{L}_{\text{Ploss}} = \|\mathbf{q} \circ \mathbf{x} - \hat{\mathbf{q}} \circ \mathbf{x}\|_2$.

We unify the above loss functions under a common framework for a fair comparison with our Bingham losses. The same network, dataset, training schemes and tasks are used to compute different losses. Note, when training with those losses, only the quaternion parts of our DBN are trained. It is also possible to incorporate our Bingham losses in their tasks to train their networks, simply by adding more outputs to the last layer without changing the overall architecture.

To further showcase the power of the proposed algorithm, we include some extra independent state-of-the-art baselines. In particular, PointNetLK [1] and IT-Net [98] are two state-of-the-art deep learning-based algorithms for iteratively estimating the relative poses between pairs of point clouds. For a fair comparison, we pair the canonical and rotated point clouds to compose the input and try to predict the relative rotation. This notion is identical to what our MBN tries to predict. The remaining configurations are kept the same as in the original work.

Metrics Due to the potential ambiguities in the point clouds, it is inappropriate to use angular error to measure the qualities of the predictions with regard to the ground truths as different poses might align the point clouds equally well. Instead, we measure the *Chamfer distance* (CD) between the point clouds rotated by the ground truth and predicted

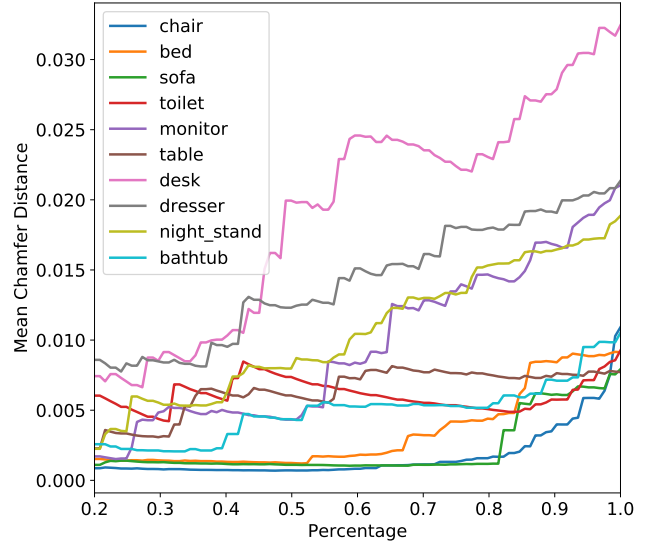


Fig. 9: As uncertainty threshold increases, the average CD of predictions whose uncertainties are below a threshold increases accordingly.

pose. In order to show how multiple modes are captured by our mixture Bingham networks, we also measure *Self-EMD* (SEMD) [66], the earth movers distance [81] of turning a multi-modal distribution into a unimodal one.

Dataset We choose ModelNet10 [96] as the benchmark to conduct our evaluations. This dataset contains 10 classes, and objects from each class possess unique geometries as well as different level of symmetries. This creates an ideal situation for validating our method in terms of uncertainty and ambiguity. We conduct class-level object pose estimation on this dataset, and follow the original train/test split.

6.3 Pose Estimation Evaluation

UBN and MBN output continuous distributions instead of discrete poses. To evaluate their performance on the task

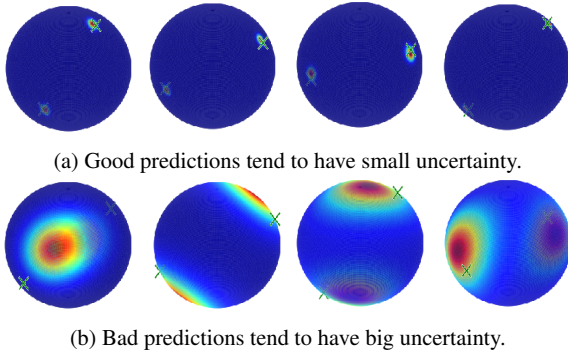


Fig. 10: Bingham distributions generated by Unimodal Bingham Network. The ground truth poses are marked with “x”. Predictions of UBN are centered on the mode (the red-most region).

of pose estimation, a single pose prediction for UBN is decided by taking the mode of the predicted continuous Bingham distribution, which equals the pose with highest chance in the according probability space. Similarly, for MBN, the component with the largest weight is selected and its mode serves as the single prediction in this evaluation.

Quantitative evaluations Table 7 showcases the average *Chamfer Distance* results of all the baselines as well as our Deep Bingham Network, including both UBN and MBN.

In comparison to all the baselines, our UBN not only achieves best performances across all the classes, but also provides an extra uncertainty information. We attribute the improvement to the fact that Bingham distribution enables an anisotropic distance taking into account the uncertainty in directions. This better reflects the real distance between the predictions and ground truth, thus lead to a better convergence.

However, like other baselines, UBN is incapable of dealing with ambiguities from point clouds with rotational symmetries. This is powerfully demonstrated by the further improvements obtained by our MBNs, where different modes triggering ambiguities are captured by different components of the underlying multimodal Bingham model.

PointNetLK [1] and IT-Net [98] work in a similar way as the celebrated iterative closest point (ICP) algorithm [6] and they are proven to be robust methods. Yet as we can see from Table 7, they could not cope with the drastic changes in the poses that well and this leads to inferior performances on this evaluation. Different to all the baseline competitors, our MBN does not require a canonical point set to estimate the orientation of the input. Instead, it relies on a high-level abstracted/implicit canonical notion for the entire class. Moreover, this canonical form is learned from the training data, making the algorithm more robust and efficient.

Table 8: Filtering hypotheses for registration using uncertainty information. The first row is the threshold value below which the hypotheses survive. The second row is the registration recall. It reaches 77.7% if no hypotheses are filtered, resulting in the method of [29]. The third row is the percentage of filtered hypotheses. The lower the uncertainty threshold, the more hypotheses are dropped. With the aid of our uncertainty, more than 20% of the hypotheses could be neglected without harming the performance. Even when 54.1% hypotheses are dropped, the performance decreases only by 4.7%.

Threshold	0.30	0.25	0.20	0.15	0.10
Ave. Recall	0.777	0.773	0.769	0.742	0.730
Drop Rate	0.184	0.221	0.280	0.388	0.541

To see how number of components would impact the performance of MBNs, we provide results for several versions of MBNs with 5, 10, 25 and 50 components respectively. As the number of components increases from 5 to 10, a further improvement can be clearly observed. However, the increase in the number of components (up to 25 and 50) does not translate to an increase in the performances. The existence of such sweet spot indicates the saturation of the mode diversity, i.e. excessive over-parameterization of the modes causes complexity in learning. Nevertheless, overall, all our MBNs outperform the other baselines which lack the ability to handle ambiguity.

Uncertainty Evaluation To better understand how our uncertainty gauge works, we plotted Fig. 9 by computing the average CDs of predictions with uncertainties less than a varying number of thresholds. As the uncertainty threshold decreases, the average CDs also decrease accordingly, which indicates that predictions with less uncertainty tend to align the point clouds better with the ground truth. We visualize the predicted Bingham distributions along with the ground truths in Fig. 10. In general, good predictions signal clustered and peaked distributions, while bad ones tend to spread and result in higher uncertainties.

To further demonstrate that our uncertainty information could be useful in practical 3D applications, we take the Partial Scan Registration task from [29]. In their work, local patches from two adjacent partial scans are first matched by learned descriptors. Each pair of patches is fed into another regression network to predict the relative rigid transformation between them (rotation only, as the translation part can be computed later by the distances between centers of the local patches). All the local patches are ideally aligned under the same relative pose. In the last stage of their pipeline, a pool of pose hypotheses is generated and exhaustively evaluated. The single best hypothesis is in the end picked as the final pose prediction to register the two partial scans.

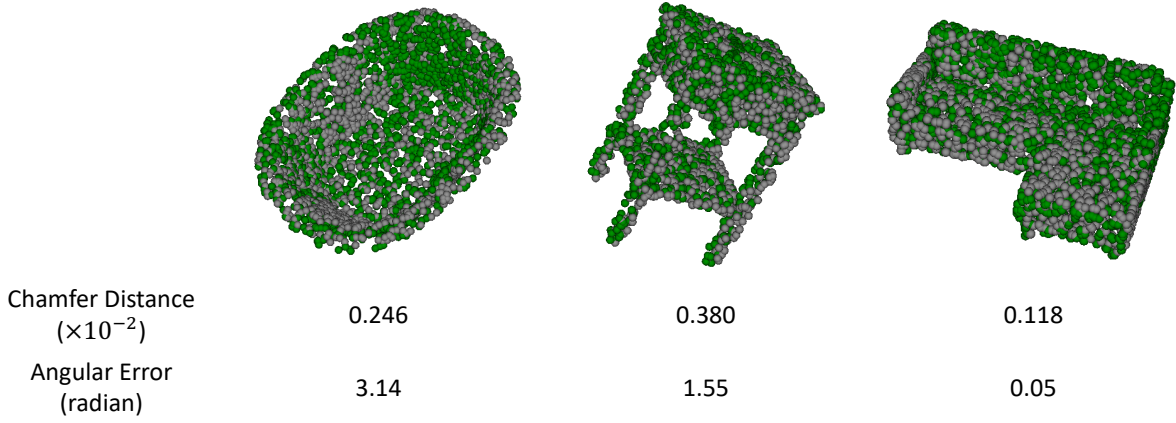


Fig. 11: The commonly used angular error would fail to be a good metric for the predictions which are different from the ground truth pose but still result in a good alignment for objects with symmetries. In this case, the chamfer distance better reflects the quality of the predictions. Gray points are from the ground truth point cloud and green ones from predicted point cloud.

Table 9: Comparison between best branch selection criteria for MBN and the task of object pose estimation. Reported is the average Chamfer distance on the ModelNet10 dataset.

Selection	L1	Prob
CD	1.057	0.973

For our purpose, we add extra output units to their network to predict *concentration parameters* for Λ and then train it with our Bingham loss. With our extra uncertainty information, it is possible to filter out some of the generated hypotheses that now are associated with high uncertainties and as a result accelerate the registration, without harming the performance much, as shown in Table 8.

Qualitative evaluations Fig. 11 shows the overlaps of point clouds rotated by ground truth poses and predicted poses. In all the displayed cases, the predictions achieve good alignment with the ground truths, which can be well indicted by the small chamfer distances. However, if we use angular error as the metric to qualify the results, even though it could still obtain small values for unambiguous objects (e.g. sofa), but it might disregard the first two predictions with big errors due to the ambiguities in the objects.

Fig. 12 demonstrates how MBN is able to capture different modes in the data when ambiguities are presented. For objects with rotational symmetries, different poses which could equivalently align them are captured by different components. It is possible to further derive the symmetric axis of those objects based on the set of various predictions.

In cases where an object does not carry ambiguity, we can see from Fig. 13 that all the components tend to agree on the predictions. It shows that extra components are not bur-

donsome for non-ambiguous objects. This kind of predictions could be further utilized as a sign to indicate whether the given point cloud is rotational symmetric or not.

6.4 Choice of the best branch

In this application, we use probability as the default metric in RWTA to select the best branch according to Eq. (25). To validate this decision, we compare the two strategies as described in Eq. (24) and Eq. (25). As we can see from Table 9, performance-wise the two criteria are close to each other. However, as the probabilities are anyway needed for the final RWTA loss, it reduces the total amount of computation by using them for the purpose of branch selection as well.

7 Ablation Studies

We now provide further ablation studies that have been conducted for both applications, relocalization and object pose estimation. First, we evaluate the effect of various methods of constructing $\mathbf{V}$ and winner-takes-all training schemes proposed in current literature. We then include another rotation parameterization in our framework that has been specifically proposed for training with neural networks. We summarize our findings in common tables where the two subtables refer to a) relocalization and b) object pose estimation.

7.1 Variants of Constructing $\mathbf{V}$

As described in section 3.1 we compare between three ways of incorporating orientation learning into neural networks,

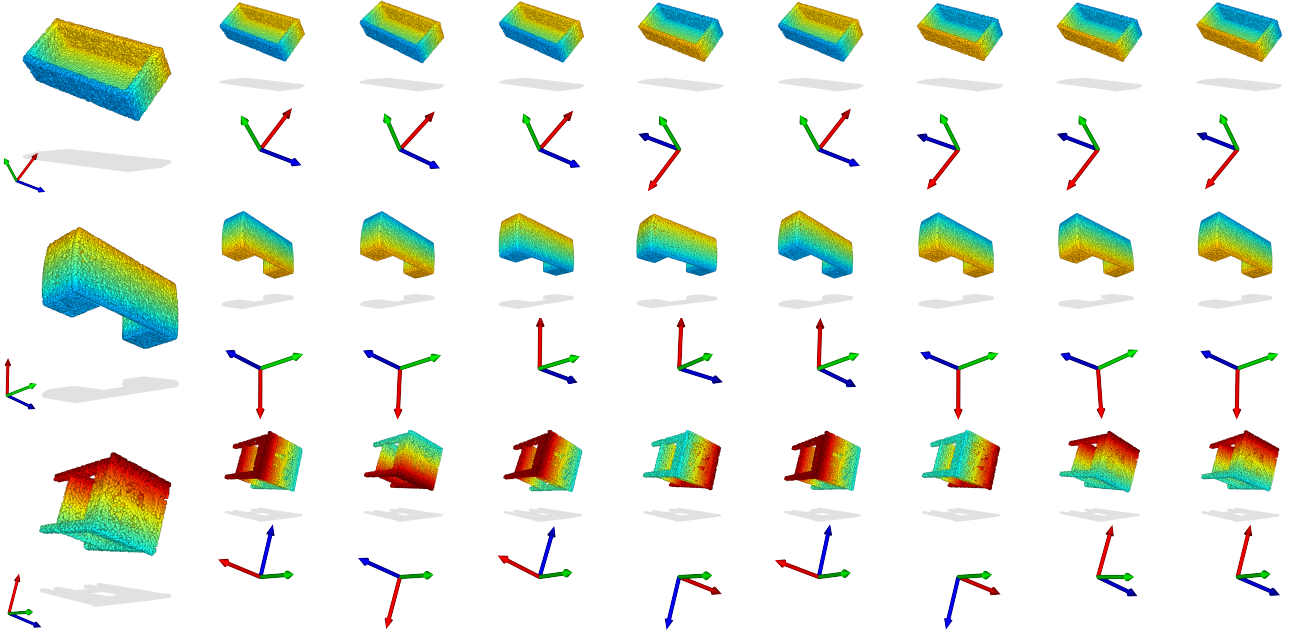


Fig. 12: Ambiguous poses could be well captured by different components of our Multimodal Bingham Network. The first column shows the object under ground truth poses. The rest of the columns show predicted poses (represented as local reference frames) and the correspondent rotated object. Point clouds are colored by the coordinates in the non-rotated version.

namely Gram-Schmidt (G), Cayley Transform (C) and the method of Birdal et. al [13] (B). We report our findings in Table 10. In comparison to Gram Schmidt orthonormalization the remaining methods only require four parameters to be estimated instead of the 16 entries of the matrix $\mathbf{V}$. In the case of camera localization for our UBN as well as multimodal MBN-MB we found the Skew-Symmetric construction to outperform both Gram-Schmidt and the employed method of Birdal et al. However, overall we have found the method of Birdal and our MBN network to achieve the best performance. This is further validated for the task of object pose estimation where our MBN model clearly outperforms the remaining construction methods.

7.2 Variants of Multiple Hypotheses Prediction Strategies

In our main experiments, we stick with the relaxed version of WTA, termed as RWTA. Recently, [66] proposed EWTA, an evolving version of WTA, to further alleviate the collapse problems of the RWTA training schemes proposed in [82]. Updating the top k hypotheses instead of only the best one, EWTA increases the number of hypotheses that are actually used during training, resulting in fewer wrong mode predictions that do not match the actual distribution.

We compared different versions of MHP training schemes for our applications, including WTA, RWTA and EWTA. The results can be found in Table 11a and Table 11b.

Table 10: Results on different strategies of constructing $\mathbf{V}$, including Gram-Schmidt (G), Cayley Transform (C) and Birdal et al. [13] (B) for camera localization 10a and point cloud pose estimation 10b.

	UBN	MBN-MB	MBN
Threshold	G / C / B	G / C / B	G / C / B
10° / 0.1m	0.15 / 0.16 / 0.11	0.13 / 0.13 / 0.08	0.21 / 0.18 / 0.22
15° / 0.2m	0.42 / 0.43 / 0.36	0.30 / 0.38 / 0.28	0.51 / 0.46 / 0.57
20° / 0.3m	0.54 / 0.54 / 0.50	0.39 / 0.49 / 0.37	0.63 / 0.56 / 0.69

(a) Ratio of correct poses for several thresholds, averaged over our ambiguous relocalization dataset.

	UBN	MBN-MB	MBN
	G / C / B	G / C / B	G / C / B
	4.654 / 2.821 / 1.477	3.233 / 2.458 / 1.237	2.867 / 1.597 / 0.973

(b) Average chamfer distances ($\times 10^{-2}$) on point cloud pose estimation, averaged over ModelNet10 dataset.

As it is not straightforward how k should be chosen in EWTA, we 1) start with $k = K$, where K is the number of hypotheses and gradually decrease k until $k = 1$ (as proposed in [66]) and 2) start with the best half hypotheses, i.e. $k = 0.5 \cdot K$. We set $K = 50$ in our experiments. We have found k to strongly influence the accuracy of our model. In our applications, however, we have found the wrong predictions to have high uncertainty so that, if desired, they can easily be removed. Overall, RWTA results in the highest per-

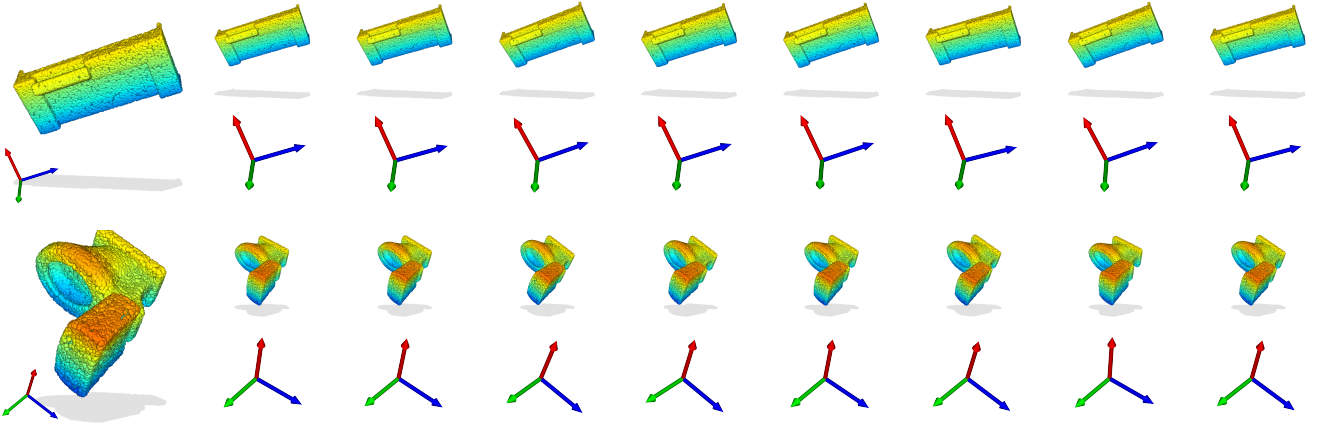


Fig. 13: For non-ambiguous objects, different components would generate similar predictions, where all modes correctly collapse. This can also be used to check the existence of ambiguities.

formance across all three, in both of the application scenarios. Therefore, we chose to remain with RWTA to train our models. This implicitly admits $k = 1$.

7.3 Impact of RWTA Loss

To show how the RWTA loss helps overcome problems of mode collapse as well as facilitate the training process for MBN, we introduce a training instance following the MDN [14] using only Mixture Bingham Loss, dubbed as *MBN-MB*. From Table 2 and Table 7, we can see for both camera relocalization and point cloud pose estimation, MBN-50 demonstrates constant improved performances over MBN-MB with the same configuration. Table 11 lists a complete comparison with MBN trained with only MB loss and the other versions combined with one of the WTA-based loss. We can see WTA-based losses not only improve the performance, but also increases the diversity in the multiple predictions which is illustrated by larger SEMD values.

7.4 Variants of Rotation Parameterization

The best choice of rotation parameterization for training deep learning models is an open question. PoseNet [55] proposed to use quaternions due to the ease of normalization. The ambiguities can be resolved by mapping the predictions to one hemisphere. MapNet [18] further showed improvements in using the axis angle representation. Recently it has been shown that any representation with four or less degrees of freedom suffers from discontinuities in mapping to $SO(3)$. This might harm the performance of deep learning models. Instead, [104] proposed a continuous 6D or 5D representation. We ablate in this context by mapping all predictions to the proposed 6D representation and model them using

Table 11: Comparison between different MHP variants, including WTA, EWTA[66] with RWTA[82] for camera localization 11a and point cloud pose estimation 11b.

Threshold	MB	WTA	EWTA (k=50)	EWTA (k=25)	RWTA (k=1, used)
10° / 0.1m	0.08	0.21	0.20	0.20	0.22
15° / 0.2m	0.28	0.56	0.51	0.51	0.57
20° / 0.3m	0.37	0.69	0.62	0.66	0.69

(a) Percentage of correct poses for several thresholds, averaged over all scenes of the ambiguous relocalization dataset.

	MB	WTA	EWTA(25)	EWTA(50)	RWTA
SEMD	0.425	0.777	0.639	0.513	0.729
CD	1.237	1.105	1.090	1.160	0.973

(b) Average SEMD and chamfer distances ($\times 10^{-2}$) on ModelNet10 across all the classes.

a Gaussian mixture models (GMM), similar to a MDN. In Table 12a, 'Geo + L1' refers to a direct regression using the geodesic loss proposed in [104] and an l_1 loss on the translation. Table 12b lists results for point cloud pose estimation. In both cases, we can see there is a clear improvement in the results when the model is lifted from a single prediction to multiple predictions, which further validates ambiguities could be well handled with multimodality. Also, in both MBN variants, when RWTA loss is incorporated, the performance could be further boosted from pure MDN, which qualifies our proposed training schemes as well.

8 Conclusion

We have proposed an elegant solution in this paper to enable end-to-end modeling of pose distributions for 3D rota-

Table 12: Results using continuous 6D representation [104] to model rotations instead of a Bingham distribution on the quaternion for camera localization 12a and point cloud pose estimation 12b

6D + 3D	Threshold	Geo+L1	Uni.	MDN	MC-Dropout	MBN-CE	MBN
A	10° / 0.1m	0.41	0.48	0.01	0.26	0.38	0.38
	15° / 0.2m	0.90	0.89	0.14	0.83	0.81	0.79
	20° / 0.3m	0.96	0.92	0.23	0.91	0.84	0.81
B	10° / 0.1m	0.03	0.03	0.02	0.02	0.06	0.07
	15° / 0.2m	0.16	0.16	0.11	0.13	0.29	0.33
	20° / 0.3m	0.22	0.23	0.14	0.21	0.38	0.42
C	10° / 0.1m	0.17	0.19	0.12	0.12	0.18	0.20
	15° / 0.2m	0.46	0.51	0.36	0.36	0.44	0.49
	20° / 0.3m	0.62	0.67	0.47	0.56	0.56	0.61
D	10° / 0.1m	0.07	0.01	0.01	0.04	0.08	0.08
	15° / 0.2m	0.30	0.06	0.09	0.18	0.35	0.40
	20° / 0.3m	0.48	0.13	0.14	0.36	0.55	0.60
E	10° / 0.1m	0.34	0.24	0.30	0.21	0.34	0.42
	15° / 0.2m	0.74	0.63	0.65	0.65	0.76	0.77
	20° / 0.3m	0.84	0.79	0.76	0.82	0.88	0.90
Average	10° / 0.1m	0.20	0.19	0.09	0.13	0.21	0.23
	15° / 0.2m	0.51	0.45	0.27	0.43	0.53	0.56
	20° / 0.3m	0.63	0.55	0.35	0.57	0.64	0.67

(a) Ratio of correct camera poses on our ambiguous relocation dataset. Camera poses are modelled with six dimensions for rotation and three for translations resulting in a nine dimensional representation for a camera pose. A: Blue Chairs, B: Meeting Table, C: Staircase, D: Staircase Extended and E: Seminar Room.

6D	Geo	Uni	MDN	MC-Dropout	MBN-CE	MBN
Bathtub	6.246	7.228	0.513	4.048	0.445	0.427
Bed	1.533	1.730	0.517	1.407	0.277	0.346
Chair	0.559	0.702	0.515	0.774	0.543	0.537
Desk	4.404	4.650	3.144	4.050	3.023	3.193
Dresser	4.792	3.799	2.216	2.590	2.121	2.333
Monitor	1.694	2.124	1.076	2.136	1.178	1.161
Night Stand	3.756	3.656	1.559	2.502	1.445	1.434
Sofa	0.313	0.374	0.319	0.503	0.324	0.312
Table	9.913	8.685	0.473	3.161	0.654	0.621
Toilet	0.427	0.873	1.065	0.750	0.266	0.423
Average	3.364	3.382	1.140	2.192	1.028	1.079

(b) Point cloud pose estimation results on ModelNet10 across all the classes using the 6D representation [104] to model rotations.

tion estimation via deep networks. We cover both unimodal (UBN) and multimodal (MBN) cases that are able to infer single as well as mixture distributions. We illustrate the feasibility to train neural networks to regress parameters of a Bingham distribution for obtaining pose predictions as well as uncertainty information. A MDN-like Multimodal Bingham network is designed targeting ambiguity issues which cannot be handled by Unimodal Bingham Network. Novel training schemes which resort to WTA strategy to facilitate the success of training a Bingham Mixture model are evaluated, avoiding mode collapse, and as a result providing multiple plausible pose predictions as well as associated uncertainty values in each hypothesis. We exhaustively evaluated our methods on two fundamental pose-related vision tasks, namely point cloud pose estimation and camera localization. For the latter we extend our framework to addition-

ally model the translation of a camera’s pose using Gaussian Mixture Models. We demonstrated our model’s superiority over the state-of-the-art on both tasks, obtaining consistently better mode predictions. We believe those solutions can be easily incorporated into other neural network-based pose estimation applications to improve their performances without heavy modifications.

Acknowledgements This project is supported by Bavaria California Technology Center (BaCaTeC), Stanford-Ford Alliance, NSF grant IIS-1763268, Vannevar Bush Faculty Fellowship, Samsung GRO program, the Stanford SAIL Toyota Research, and the PRIME programme of the German Academic Exchange Service (DAAD) with funds from the German Federal Ministry of Education and Research (BMBF).

References

1. Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: Pointnetlk: Robust & efficient point cloud registration using pointnet. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7163–7172 (2019) **16, 17**
2. Arun Srivatsan, R., Xu, M., Zavallos, N., Choset, H.: Probabilistic pose estimation using a bingham distribution-based linear filter. The International Journal of Robotics Research **37**(13-14), 1610–1631 (2018) **2**
3. Balntas, V., Li, S., Prisacariu, V.: Relocnet: Continuous metric learning relocalisation using neural nets. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 751–767 (2018) **14**
4. Barfoot, T.D., Furgale, P.T.: Associating uncertainty with three-dimensional poses for use in estimation problems. IEEE Transactions on Robotics **30**(3), 679–693 (2014) **5**
5. Berger, J.O.: Statistical decision theory and Bayesian analysis. Springer Science & Business Media (2013) **3**
6. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures, vol. 1611, pp. 586–606. International Society for Optics and Photonics (1992) **17**
7. Bingham, C.: An antipodally symmetric distribution on the sphere. The Annals of Statistics pp. 1201–1225 (1974) **2, 4**
8. Birdal, T., Arbel, M., Şimşekli, U., Guibas, L.: Synchronizing probability measures on rotations via optimal transport. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020) **5**
9. Birdal, T., Bala, E., Eren, T., Ilic, S.: Online inspection of 3d parts via a locally overlapping camera network. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–10. IEEE (2016) **2**
10. Birdal, T., Ilic, S.: Point pair features based object detection and pose estimation revisited. In: 2015 Inter-

- national Conference on 3D Vision, pp. 527–535. IEEE (2015) [3](#)
11. Birdal, T., Ilic, S.: Cad priors for accurate and flexible instance reconstruction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 133–142 (2017) [3](#)
 12. Birdal, T., Simsekli, U.: Probabilistic permutation synchronization using the riemannian structure of the birkhoff polytope. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 11105–11116 (2019) [2](#)
 13. Birdal, T., Simsekli, U., Eken, M.O., Ilic, S.: Bayesian pose graph optimization via bingham distributions and tempered geodesic mcmc. In: Advances in Neural Information Processing Systems, pp. 308–319 (2018) [2](#), [5](#), [19](#)
 14. Bishop, C.M.: Mixture density networks (1994) [3](#), [4](#), [7](#), [8](#), [9](#), [10](#), [20](#)
 15. Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., Rother, C.: Dsac-differentiable ransac for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6684–6692 (2017) [3](#)
 16. Brachmann, E., Michel, F., Krull, A., Ying Yang, M., Gumhold, S., et al.: Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016) [3](#)
 17. Brachmann, E., Rother, C.: Learning less is more-6d camera localization via 3d surface regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4654–4662 (2018) [3](#)
 18. Brahmabhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J.: Geometry-aware learning of maps for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2616–2625 (2018) [10](#), [11](#), [14](#), [20](#)
 19. Breiman, L.: Random forests. *Machine learning* **45**(1), 5–32 (2001) [3](#)
 20. Bui, M., Albarqouni, S., Ilic, S., Navab, N.: Scene coordinate and correspondence learning for image-based localization. In: British Machine Vision Conference (BMVC) (2018) [3](#)
 21. Bui, M., Baur, C., Navab, N., Ilic, S., Albarqouni, S.: Adversarial networks for camera pose regression and refinement. In: International Conference on Computer Vision Workshops (ICCVW) (2019) [3](#)
 22. Bui, M., Birdal, T., Deng, H., Albarqouni, S., Guibas, L., Ilic, S., Navab, N.: 6d camera relocalization in ambiguous scenes via continuous multimodal inference. In: European Conference on Computer Vision (ECCV) (2020) [2](#), [9](#), [11](#)
 23. Busam, B., Birdal, T., Navab, N.: Camera pose filtering with local regression geodesics on the riemannian manifold of dual quaternions. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 2436–2445 (2017) [5](#)
 24. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics* **32**(6), 1309–1332 (2016) [2](#)
 25. Clark, R., Wang, S., Markham, A., Trigoni, N., Wen, H.: Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6856–6864 (2017) [3](#), [10](#), [11](#)
 26. Corona, E., Kundu, K., Fidler, S.: Pose estimation for objects with rotational symmetry. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7215–7222. IEEE (2018) [2](#), [3](#), [4](#)
 27. Deng, H., Birdal, T., Ilic, S.: Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 602–618 (2018) [3](#)
 28. Deng, H., Birdal, T., Ilic, S.: Ppfnet: Global context aware local features for robust 3d point matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 195–205 (2018) [3](#)
 29. Deng, H., Birdal, T., Ilic, S.: 3d local features for direct pairwise registration. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) [3](#), [17](#)
 30. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. Ieee (2009) [3](#), [14](#)
 31. Dey, D., Ramakrishna, V., Hebert, M., Andrew Bagnell, J.: Predicting multiple structured visual interpretations. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2947–2955 (2015) [4](#)
 32. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine* **13**(2), 99–110 (2006) [2](#)
 33. Falorsi, L., de Haan, P., Davidson, T.R., Forré, P.: Reparameterizing distributions on lie groups. *arXiv preprint arXiv:1903.02958* (2019) [5](#)
 34. Feng, W., Tian, F.P., Zhang, Q., Sun, J.: 6d dynamic camera relocalization from single reference image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4049–4057 (2016) [3](#)

35. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**(6), 381–395 (1981) [3](#)
36. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: *international conference on machine learning*, pp. 1050–1059 (2016) [3](#)
37. Gilitschenski, I., Sahoo, R., Schwarting, W., Amini, A., Karaman, S., Rus, D.: Deep orientation uncertainty learning based on a bingham loss. In: *International Conference on Learning Representations* (2020). URL <https://openreview.net/forum?id=ryloogSKDS> [4](#), [10](#)
38. Glover, J., Bradski, G., Rusu, R.B.: Monte carlo pose estimation with quaternion kernels and the bingham distribution. In: *Robotics: science and systems*, vol. 7, p. 97 (2012) [4](#), [5](#)
39. Glover, J., Kaelbling, L.P.: Tracking the spin on a ping pong ball with the quaternion bingham filter. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4133–4140 (2014) [4](#), [5](#)
40. Grassia, F.S.: Practical parameterization of rotations using the exponential map. *Journal of graphics tools* **3**(3), 29–48 (1998) [5](#)
41. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. *JMLR.org* (2017) [3](#)
42. Guzman-Rivera, A., Batra, D., Kohli, P.: Multiple choice learning: Learning to produce multiple structured outputs. In: *Advances in Neural Information Processing Systems*, pp. 1799–1807 (2012) [4](#)
43. Haarbach, A., Birdal, T., Ilic, S.: Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation. In: *3D Vision (3DV), 2018 Sixth International Conference on*, pp. 381–389. *IEEE* (2018). DOI [10.1109/3DV.2018.00051](https://doi.org/10.1109/3DV.2018.00051) [5](#)
44. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (2016) [3](#), [10](#)
45. Herz, C.S.: Bessel functions of matrix argument. *Annals of Mathematics* **61**(3), 474–523 (1955). URL <http://www.jstor.org/stable/1969810> [5](#)
46. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: *Asian conference on computer vision*, pp. 548–562. *Springer* (2012) [2](#)
47. Horaud, R., Conio, B., Leboulleux, O., Lacolle, B.: An analytic solution for the perspective 4-point problem. In: *Proceedings CVPR'89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 500–507. *IEEE* (1989) [2](#)
48. Jaynes, E.T.: Information theory and statistical mechanics. *Physical review* **106**(4), 620 (1957) [7](#)
49. Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5010–5019 (2018) [3](#)
50. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1521–1529 (2017) [3](#)
51. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: *2016 IEEE international conference on Robotics and Automation (ICRA)*, pp. 4762–4769. *IEEE* (2016) [2](#), [4](#), [10](#), [11](#), [12](#)
52. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)* (2016) [3](#)
53. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5974–5983 (2017) [3](#), [10](#), [11](#), [15](#), [16](#)
54. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: *Advances in Neural Information Processing Systems (NIPS)* (2017) [4](#)
55. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: *Proceedings of the International Conference on Computer Vision (ICCV)* (2015) [3](#), [4](#), [10](#), [12](#), [20](#)
56. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014) [15](#)
57. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013) [4](#)
58. Kume, A., Wood, A.T.: Saddlepoint approximations for the bingham and fisher-bingham normalising constants. *Biometrika* **92**(2), 465–476 (2005) [7](#)
59. Kurz, G., Gilitschenski, I., Julier, S., Hanebeck, U.D.: Recursive estimation of orientation based on the bingham distribution. In: *Information Fusion (FUSION), 2013 16th International Conference on*, pp. 1487–1494. *IEEE* (2013) [5](#)

60. Kurz, G., Gilitschenski, I., Pfaff, F., Drude, L., Hanebeck, U.D., Haeb-Umbach, R., Siegwart, R.Y.: Directional statistics and filtering using libdirectional. arXiv preprint arXiv:1712.09718 (2017) [7](#)
61. Labbé, M., Michaud, F.: Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics* **36**(2), 416–446 (2019) [10](#)
62. Liao, S., Gavves, E., Snoek, C.G.: Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9759–9767 (2019) [16](#)
63. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection—a new baseline. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6536–6545 (2018) [4](#)
64. Luc, P., Neverova, N., Couprie, C., Verbeek, J., LeCun, Y.: Predicting deeper into the future of semantic segmentation. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 648–657 (2017) [4](#)
65. Mahendran, S., Ali, H., Vidal, R.: 3d pose regression using convolutional neural networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2174–2182 (2017) [16](#)
66. Makansi, O., Ilg, E., Cicek, O., Brox, T.: Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7144–7153 (2019) [4](#), [7](#), [8](#), [10](#), [16](#), [19](#), [20](#)
67. Manhardt, F., Arroyo, D.M., Rupperecht, C., Busam, B., Birdal, T., Navab, N., Tombari, F.: Explaining the ambiguity of object detection and 6d pose from visual data. In: *International Conference of Computer Vision (ICCV)*. IEEE/CVF (2019) [2](#), [4](#), [8](#), [16](#)
68. Mardia, K.V., Jupp, P.E.: *Directional statistics*, vol. 494. John Wiley & Sons (2009) [4](#)
69. Massiceti, D., Krull, A., Brachmann, E., Rother, C., Torr, P.H.: Random forests versus neural networks—what’s best for camera localization? In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5118–5125. IEEE (2017) [3](#)
70. McLachlan, G.J., Basford, K.E.: *Mixture models: Inference and applications to clustering*, vol. 84. M. Dekker New York (1988) [9](#)
71. Morawiec, A., Field, D.: Rodrigues parameterization for orientation and misorientation distributions. *Philosophical Magazine A* **73**(4), 1113–1130 (1996) [5](#)
72. Murray, R.M.: *A mathematical introduction to robotic manipulation*. CRC press (1994) [5](#)
73. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: *NIPS Autodiff Workshop* (2017) [10](#), [15](#)
74. Peretroukhin, V., Wagstaff, B., Giamou, M., Kelly, J.: Probabilistic regression of rotations using quaternion averaging and a deep multi-headed network. arXiv preprint arXiv:1904.03182 (2019) [14](#)
75. Piasco, N., Sidibé, D., Demonceaux, C., Gouet-Brunet, V.: A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition* **74**, 90–109 (2018) [2](#)
76. Pitteri, G., Ramamonjisoa, M., Ilic, S., Lepetit, V.: On object symmetries and 6d pose estimation from images. In: *2019 International Conference on 3D Vision (3DV)*, pp. 614–622. IEEE (2019) [2](#), [3](#), [4](#)
77. Prokudin, S., Gehler, P., Nowozin, S.: Deep directional statistics: Pose estimation with uncertainty quantification. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 534–551 (2018) [3](#), [4](#), [6](#)
78. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9277–9286 (2019) [2](#), [3](#)
79. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660 (2017) [3](#), [15](#)
80. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: *Advances in neural information processing systems*, pp. 5099–5108 (2017) [3](#)
81. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover’s distance as a metric for image retrieval. *International journal of computer vision* **40**(2), 99–121 (2000) [10](#), [16](#)
82. Rupperecht, C., Laina, I., DiPietro, R., Baust, M., Tombari, F., Navab, N., Hager, G.D.: Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3591–3600 (2017) [2](#), [4](#), [8](#), [19](#), [20](#)
83. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H., Davison, A.J.: Slam++: Simultaneous localisation and mapping at the level of objects. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1352–1359 (2013) [2](#)
84. Sattler, T., Havlena, M., Radenovic, F., Schindler, K., Pollefeys, M.: Hyperpoints and fine vocabularies for large-scale location recognition. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2102–2110 (2015) [2](#)

85. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of cnn-based absolute camera pose regression. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3302–3312 (2019) [11](#)
86. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113 (2016) [2](#)
87. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937 (2013) [2](#), [3](#), [10](#)
88. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *CoRR* **abs/1409.1556** (2014) [3](#)
89. Subbarao, R., Meer, P.: Nonlinear mean shift over riemannian manifolds. *International journal of computer vision* **84**(1), 1 (2009) [12](#)
90. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826 (2016) [14](#)
91. Ullman, S.: The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences* **203**(1153), 405–426 (1979) [2](#)
92. Valentin, J., Nießner, M., Shotton, J., Fitzgibbon, A., Izadi, S., Torr, P.H.: Exploiting uncertainty in regression forests for accurate camera relocalization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4400–4408 (2015) [3](#)
93. Wang, Q.A.: Probability distribution and entropy as a measure of uncertainty. *Journal of Physics A: Mathematical and Theoretical* **41**(6), 065004 (2008) [7](#)
94. Wang, Y., Solomon, J.M.: Deep closest point: Learning representations for point cloud registration. *arXiv preprint arXiv:1905.03304* (2019) [3](#), [15](#), [16](#)
95. Wang, Y., Solomon, J.M.: Prnet: Self-supervised learning for partial-to-partial registration. *Advances in Neural Information Processing Systems* (2019) [16](#)
96. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920 (2015) [2](#), [16](#)
97. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In: *Robotics: Science and Systems (RSS)* (2018) [3](#), [16](#)
98. Yuan, W., Held, D., Mertz, C., Hebert, M.: Iterative transformer network for 3d point cloud. *arXiv preprint arXiv:1811.11209* (2018) [16](#), [17](#)
99. Zakharov, S., Kehl, W., Planche, B., Hutter, A., Ilic, S.: 3d object instance recognition and pose estimation using triplet loss with dynamic margin. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 552–559. *IEEE* (2017) [3](#)
100. Zakharov, S., Shugurov, I., Ilic, S.: Dpod: Dense 6d pose object detector in rgb images. *Proceedings of the IEEE International Conference on Computer Vision* (2019) [2](#), [3](#)
101. Zeisl, B., Sattler, T., Pollefeys, M.: Camera pose voting for large-scale image-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2704–2712 (2015) [2](#)
102. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1802–1811 (2017) [3](#)
103. Zhao, Y., Birdal, T., Lenssen, J.E., Menegatti, E., Guibas, L., Tombari, F.: Quaternion equivariant capsule networks for 3d point clouds. In: *European Conference on Computer Vision*, pp. 1–19. *Springer* (2020) [3](#)
104. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5745–5753 (2019) [20](#), [21](#)
105. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499 (2018) [2](#)
106. Zolfaghari, M., Çiçek, Ö., Ali, S.M., Mahdisoltani, F., Zhang, C., Brox, T.: Learning representations for predicting future activities. *arXiv preprint arXiv:1905.03578* (2019) [3](#)