Robotics: Science and Systems 2020
Corvalis, Oregon, USA, July 12-16, 2020

A Smooth Representation of Belief over $\mathrm{SO}(3)$ for Deep Rotation Learning with Uncertainty

Valentin Peretroukhin,[1,3] Matthew Giamou,[1] David M. Rosen,[2] W. Nicholas Greene,[3]
Nicholas Roy,[3] and Jonathan Kelly[1]
[1]Institute for Aerospace Studies, University of Toronto;
[2]Laboratory for Information and Decision Systems,
[3]Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology

***Abstract*—Accurate rotation estimation is at the heart of robot perception tasks such as visual odometry and object pose estimation. Deep neural networks have provided a new way to perform these tasks, and the choice of rotation representation is an important part of network design. In this work, we present a novel symmetric matrix representation of the 3D rotation group, $\mathrm{SO}(3)$, with two important properties that make it particularly suitable for learned models: (1) it satisfies a smoothness property that improves convergence and generalization when regressing large rotation targets, and (2) it encodes a symmetric Bingham belief over the space of unit quaternions, permitting the training of uncertainty-aware models. We empirically validate the benefits of our formulation by training deep neural rotation regressors on two data modalities. First, we use synthetic point-cloud data to show that our representation leads to superior predictive accuracy over existing representations for arbitrary rotation targets. Second, we use image data collected onboard ground and aerial vehicles to demonstrate that our representation is amenable to an effective out-of-distribution (OOD) rejection technique that significantly improves the robustness of rotation estimates to unseen environmental effects and corrupted input images, without requiring the use of an explicit likelihood loss, stochastic sampling, or an auxiliary classifier. This capability is key for safety-critical applications where detecting novel inputs can prevent catastrophic failure of learned models.**

I. Introduction

Rotation estimation constitutes one of the core challenges in robotic state estimation. Given the broad interest in applying deep learning to state estimation tasks involving rotations [4, 7, 25–27, 30, 34, 36, 39], we consider the suitability of different rotation representations in this domain. The question of which rotation parameterization to use for estimation and control problems has a long history in aerospace engineering and robotics [9]. In learning, unit quaternions (also known as Euler parameters) are a popular choice for their numerical efficiency, lack of singularities, and simple algebraic and geometric structure. Nevertheless, a standard unit quaternion parameterization does not satisfy an important continuity property that is essential for learning arbitrary rotation targets, as recently detailed in [41]. To address this deficiency, the authors of [41] derived two alternative rotation representations that satisfy this property and lead to better network performance. Both of these representations, however, are *point* representations, and do not quantify network uncertainty—an important capability in safety-critical applications.

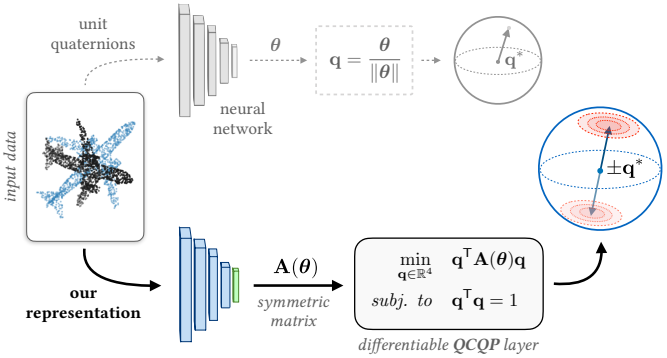


Fig. 1: We represent rotations through a symmetric matrix, $\mathbf{A}$, that defines a Bingham distribution over unit quaternions. To apply this representation to deep rotation regression, we present a differentiable layer parameterized by $\mathbf{A}$ and show how we can extract a notion of uncertainty from the spectrum of $\mathbf{A}$.

In this work, we introduce a novel representation of $\mathrm{SO}(3)$ based on a symmetric matrix that combines these two important properties. Namely, it

1) admits a smooth global section from $\mathrm{SO}(3)$ to the representation space (satisfying the continuity property identified by the authors of [41]);
2) defines a Bingham distribution over unit quaternions; and
3) is amenable to a novel out-of-distribution (OOD) detection method without any additional stochastic sampling, or auxiliary classifiers.

Figure 1 visually summarizes our approach. Our experiments use synthetic and real datasets to highlight the key advantages of our approach. We provide open source Python code[1] of our method and experiments. Finally, we note that our representation can be implemented in only a few lines of code in modern deep learning libraries such as PyTorch, and has marginal computational overhead for typical learning pipelines.

II. Related Work

Estimating rotations has a long and rich history in computer vision and robotics [17, 31, 35]. An in-depth survey of rotation

[1]Code available at https://github.com/utiasSTARS/bingham-rotation-learning.

averaging problem formulations and solution methods that deal directly with multiple rotation measurements is presented in [16]. In this section, we briefly survey techniques that estimate rotations from raw sensor data, with a particular focus on prior work that incorporates machine learning into the rotation estimation pipeline. We also review recent work on differentiable optimization problems and convex relaxation-based solutions to rotation estimation problems that inspired our work.

A. Rotation Parameterization

In robotics, it is common to parameterize rotation states as elements of the matrix Lie group $\mathrm{SO}(3)$ [2, 32]. This approach facilitates the application of Gauss-Newton-based optimization and a local uncertainty quantification through small perturbations defined in a tangent space about an operating point in the group. In other state estimation contexts, applications may eschew full 3×3 orthogonal matrices with positive determinant (i.e., elements of $\mathrm{SO}(3)$) in favour of lower-dimensional representations with desirable properties [9]. For example, Euler angles [33] are particularly well-suited to analyzing small perturbations in the steady-state flight of conventional aircraft because reaching a singularity is practically impossible [10]. In contrast, spacecraft control often requires large-angle maneuvers for which singularity-free unit quaternions are a popular choice [37].

B. Learning-based Rotation Estimation

Much recent robotics literature has focused on improving classical pose estimation with learned models. Learning can help improve outlier classification [39], guide random sample consensus [4], and fine-tune robust losses [27]. Further, fusing learned models of rotation with classical pipelines has been shown to improve accuracy and robustness of egomotion estimates [25, 26].

In many vision contexts, differentiable solvers have been proposed to incorporate learning into bundle adjustment [34], monocular stereo [7], point cloud registration [36], and fundamental matrix estimation [27]. All of these methods rely on either differentiating a singular value decomposition [27, 36], or 'unrolling' local iterative solvers for a fixed number of iterations [7, 34]. Furthermore, adding interpretable outputs to a learned pipeline has been shown to improve generalization [40] and the paradigm of differentiable pipelines has been suggested to tackle an array of different robotics tasks [18].

The effectiveness of learning with various $\mathrm{SO}(3)$ representations is explicitly addressed in [41]. Given a *representation* of $\mathrm{SO}(3)$, by which we mean a surjective mapping $f : \mathcal{X} \to \mathrm{SO}(3)$, the authors of [41] identified the existence of a continuous right-inverse of f, $g : \mathrm{SO}(3) \to \mathcal{X}$, as important for learning. Intuitively, the existence of such a g ensures that the training signal remains continuous for regression tasks, reducing errors on unseen inputs. Similarly, an empirical comparison of $\mathrm{SE}(3)$ representations for learning complex forward kinematics is conducted in [15]. Although full $\mathrm{SE}(3)$ pose estimation is important in most robotics applications, we limit our analysis to $\mathrm{SO}(3)$ representations as most pose estimation tasks can be decoupled into rotation and translation components, and the rotation component of $\mathrm{SE}(3)$ constitutes the main challenge.

C. Learning Rotations with Uncertainty

Common ways to extract uncertainty from neural networks include approximate variational inference through Monte Carlo dropout [11] and bootstrap-based uncertainty through an ensemble of models [20] or with multiple network *heads* [24]. In prior work [26], we have proposed a mechanism that extends these methods to $\mathrm{SO}(3)$ targets through differentiable quaternion averaging and a local notion of uncertainty in the tangent space of the mean.

Additionally, learned methods can be equipped with novelty detection mechanisms (often referred to as out-of-distribution or OOD detection) to help account for epistemic uncertainty. An autoencoder-based approach to OOD detection was used on a visual navigation task in [28] to ensure that a safe control policy was used in novel environments. A similar approach was applied in [1], where a single variational autoencoder was used for novelty detection and control policy learning. See [23] for a recent summary of OOD methods commonly applied to classification tasks.

Finally, unit quaternions are also important in the broad body of work related to learning directional statistics [33] that enable global notions of uncertainty through densities like the Bingham distribution, which are especially useful in modelling large-error rotational distributions [13, 14]. Since we demonstrate that our representation parameterizes a Bingham belief, it is perhaps most similar to a recently published approach that uses a Bingham likelihood to learn global uncertainty over rotations [13]. Our work differs from this approach in several important aspects: (1) our formulation is more parsimonious; it requires only a single symmetric matrix with 10 parameters to encode both the mode and uncertainty of the Bingham belief, (2) we present analytic gradients for our approach by considering a generalized QCQP-based optimization over rotations, (3) we prove that our representation admits a smooth right-inverse, and (4) we demonstrate that we can extract useful notions of uncertainty from our parameterization without using an explicit Bingham likelihood during training, avoiding the complex computation of the Bingham distribution's normalization constant.

III. Symmetric Matrices and $\mathrm{SO}(3)$

Our rotation representation is defined using the set of real symmetric 4×4 matrices with a simple (i.e., non-repeated) minimum eigenvalue:

$$\mathbf{A} \in \mathbb{S}^4 : \lambda_1(\mathbf{A}) \neq \lambda_2(\mathbf{A}), \tag{1}$$

where λ_i are the eigenvalues of $\mathbf{A}$ arranged such that $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4$, and $\mathbb{S}^n \triangleq \{\mathbf{A} \in \mathbb{R}^{n \times n} : \mathbf{A} = \mathbf{A}^\mathsf{T}\}$. Each such matrix can be mapped to a unique rotation through a differentiable quadratically-constrained quadratic program (QCQP) defined in Figure 2 and by Problem 3. This representation

has several advantages in the context of learned models. In this section, we will (1) show how the matrix $\mathbf{A}$ arises as the data matrix of a parametric QCQP and present its analytic derivative, (2) show that our representation is *continuous* (in the sense of [41]), (3) relate it to the Bingham distribution over unit quaternions, and (4) discuss an intimate connection to rotation averaging.

A. Rotations as Solutions to QCQPs

Many optimizations that involve rotations can be written as the constrained quadratic loss

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} \\ \textit{subj. to} \quad & \mathbf{x}\in\mathcal{C}, \end{aligned} \tag{2}$$

where $\mathbf{x}$ parameterizes a rotation, $\mathcal{C}$ defines a set of appropriate constraints, and $\mathbf{A}$ is a *data matrix* that encodes error primitives, data association, and uncertainty. For example, consider the Wahba problem (WP) [35]:

Problem 1 (WP with unit quaternions)**.** *Given a set of associated vector measurements* $\{\mathbf{u}_i,\mathbf{v}_i\}_{i=1}^N \subset \mathbb{R}^3$*, find* $\mathbf{q}\in S^3$ *that solves*

$$\min_{\mathbf{q}\in S^3} \sum_{i=1}^N \frac{1}{\sigma_i^2}\left\|\hat{\mathbf{v}}_i - \mathbf{q}\odot\hat{\mathbf{u}}_i\odot\mathbf{q}^{-1}\right\|^2. \tag{3}$$

where $\hat{\mathbf{p}}\triangleq\begin{bmatrix}\mathbf{p}^\mathsf{T} & 0\end{bmatrix}^\mathsf{T}$ *is the homogenization of vector* $\mathbf{p}$ *and* $\odot$ *refers to the quaternion product.*

We can convert Problem 1 to the following Quadratically-Constrained Quadratic Program (QCQP):

Problem 2 (WP as a QCQP)**.** *Find* $\mathbf{q}\in\mathbb{R}^4$ *that solves*

$$\begin{aligned} \min_{\mathbf{q}\in\mathbb{R}^4} \quad & \mathbf{q}^\mathsf{T}\mathbf{A}\mathbf{q} \\ \textit{subj. to} \quad & \mathbf{q}^\mathsf{T}\mathbf{q}=1, \end{aligned} \tag{4}$$

where the data matrix, $\mathbf{A}=\sum_{i=1}^N\mathbf{A}_i\in\mathbb{S}^4$*, is the sum of* N *terms each given by*

$$\mathbf{A}_i = \frac{1}{\sigma_i^2}\left((\|\mathbf{u}_i\|^2+\|\mathbf{v}_i\|^2)\mathbf{I}+2\mathbf{\Omega}_\ell(\hat{\mathbf{v}}_i)\mathbf{\Omega}_r(\hat{\mathbf{u}}_i)\right), \tag{5}$$

where $\mathbf{\Omega}_\ell$ *and* $\mathbf{\Omega}_r$ *are left and right quaternion product matrices, respectively (cf. [38]).*

Constructing such an $\mathbf{A}$ for input data that does not contain vector correspondences constitutes the primary challenge of rotation estimation. Consequently, we consider applying the tools of high capacity data-driven learning to the task of predicting $\mathbf{A}$ for a given input. To this end, we generalize Problem 2 and consider a parametric symmetric matrix $\mathbf{A}(\boldsymbol{\theta})$:

Problem 3 (Parametric Quaternion QCQP)**.**

$$\begin{aligned} \min_{\mathbf{q}\in\mathbb{R}^4} \quad & \mathbf{q}^\mathsf{T}\mathbf{A}(\boldsymbol{\theta})\mathbf{q} \\ \textit{subj. to} \quad & \mathbf{q}^\mathsf{T}\mathbf{q}=1, \end{aligned} \tag{6}$$

where $\mathbf{A}(\boldsymbol{\theta})\in\mathbb{S}^4$ *defines a quadratic cost function parameterized by* $\boldsymbol{\theta}\in\mathbb{R}^{10}$.

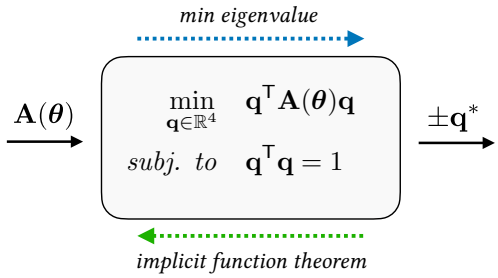


Fig. 2: A differentiable QCQP layer representing Problem 3. The layer takes as input a symmetric matrix $\mathbf{A}$ defined by the parameters $\boldsymbol{\theta}$. The solution is given by the minimum-eigenspace of $\mathbf{A}$ and the implicit function theorem can be used to derive an analytic gradient.

1) Solving Problem 3: Problem 3 is minimized by a pair of antipodal unit quaternions, $\pm\mathbf{q}^*$, lying in the one-dimensional minimum-eigenspace of $\mathbf{A}(\boldsymbol{\theta})$ [17]. Let $\mathbb{S}^4_\lambda$ be the subset of $\mathbb{S}^4$ with simple minimal eigenvalue λ_1:

$$\mathbb{S}^4_\lambda \triangleq \{\mathbf{A}\in\mathbb{S}^4 : \lambda_1(\mathbf{A})\neq\lambda_2(\mathbf{A})\} \tag{7}$$

For any $\mathbf{A}(\boldsymbol{\theta})\in\mathbb{S}^4_\lambda$, Problem 3 admits the solution $\pm\mathbf{q}^*$. Since $\mathrm{SO}(3)$ is the quotient space of the unit quaternions obtained by identifying antipodal points, this represents a single rotation solution $\mathbf{R}^*\in\mathrm{SO}(3)$. Eigendecomposition of a real symmetric matrix can be implemented efficiently in most deep learning frameworks (e.g., we use the `symeig` function in PyTorch). In practice, we encode $\mathbf{A}$ as

$$\mathbf{A}(\boldsymbol{\theta}) = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4\\ \theta_2 & \theta_5 & \theta_6 & \theta_7\\ \theta_3 & \theta_6 & \theta_8 & \theta_9\\ \theta_4 & \theta_7 & \theta_9 & \theta_{10}\end{bmatrix}, \tag{8}$$

with no restrictions on $\boldsymbol{\theta}$ to ensure that $\lambda_1(\mathbf{A})\neq\lambda_2(\mathbf{A})$. We find that this does not impede training, and note that the complement of $\mathbb{S}^4_\lambda$ is a set of measure zero in $\mathbb{S}^4$.

2) Differentiating Problem 3 wrt $\mathbf{A}$*:* The derivative of $\mathbf{q}^*$ with respect to $\boldsymbol{\theta}$ is guaranteed to exist if λ_1 is simple [21]. Indeed, one can use the implicit function theorem to show that

$$\frac{\partial\mathbf{q}^*}{\partial\mathrm{vec}(\mathbf{A})} = \mathbf{q}^*\otimes(\lambda_1\mathbf{I}-\mathbf{A})^\dagger, \tag{9}$$

where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudo-inverse, $\otimes$ is the Kronecker product, and $\mathbf{I}$ refers to the identity matrix. This gradient is implemented within the `symeig` function in PyTorch and can be efficiently implemented in any framework that allows for batch linear system solves.

B. A Smooth Global Section of $\mathrm{SO}(3)$

Consider the surjective map[2] $f:\mathbb{S}^4_\lambda\to\mathrm{SO}(3)$:

$$f:\mathbf{A}\mapsto\mathbf{R}^*, \tag{10}$$

where $\mathbf{R}^*$ is the rotation matrix corresponding to

$$\pm\mathbf{q}^* = \operatorname*{argmin}_{\mathbf{q}\in S^3}\mathbf{q}^\mathsf{T}\mathbf{A}\mathbf{q}. \tag{11}$$

[2]Surjectivity follows from the fact that f admits a global section Theorem 1.

As noted in Section II-B, the authors of [41] identified the existence of a continuous right-inverse, or section, of f as important for learning. The authors further used topological arguments to demonstrate that a continuous representation is only possible if the dimension of the embedding space is greater than four. In the case of four-dimensional unit quaternions, this discontinuity manifests itself at 180° rotations. For our ten-dimensional representation, we present a proof that one such (non-unique) continuous mapping, $g : \mathrm{SO}(3) \to \mathbb{S}^4_\lambda$, exists and is indeed smooth.

Theorem 1 (Smooth Global Section, $\mathrm{SO}(3) \to \mathbb{S}^4_\lambda$)**.** *Consider the surjective map $f : \mathbb{S}^4_\lambda \to \mathrm{SO}(3)$ such that $f(\mathbf{A})$ returns the rotation matrix defined by the two antipodal unit quaternions $\pm\mathbf{q}^*$ that minimize Problem 3. There exists a smooth and global mapping, or section, $g : \mathrm{SO}(3) \to \mathbb{S}^4_\lambda$ such that $f(g(\mathbf{R})) = \mathbf{R}$.*

Proof: Recall that the mapping $\mathbf{R}(\cdot) : S^3 \to \mathrm{SO}(3)$ from unit quaternions to rotations is continuous, surjective, and identifies antipodal unit-quaternions (i.e., sends them to the same rotation); this shows that $\mathrm{SO}(3) \cong \mathbb{RP}^3$ ($\mathrm{SO}(3)$ is *diffeomorphic* to $\mathbb{RP}^3$) as smooth manifolds. Therefore, it suffices to show that the global section $g : \mathbb{RP}^3 \to \mathbb{S}^4_\lambda$ exists. Let $[\mathbf{q}]$ be an arbitrary element of $\mathbb{RP}^3$ and define

$$g([\mathbf{q}]) \triangleq \mathbf{I} - \mathbf{q}\mathbf{q}^\mathsf{T}, \tag{12}$$

where $\mathbf{q}$ is one of the two representatives ($\pm\mathbf{q}$) of $[\mathbf{q}]$ in S^3. Note that $g(\cdot)$ is well-defined over arbitrary elements of $\mathbb{RP}^3$, since selecting either representative leads to the same output (i.e., $g(-\mathbf{q}) = g(\mathbf{q})$). By construction, $g([\mathbf{q}])$ is the orthogonal projection operator onto the 3-dimensional orthogonal complement of $\mathrm{span}(\mathbf{q}) = \mathrm{span}(-\mathbf{q})$ in $\mathbb{R}^4$. Therefore, $\lambda\{g([\mathbf{q}])\} = \lambda\{\mathbf{I} - \mathbf{q}\mathbf{q}^\mathsf{T}\} = \{0, 1, 1, 1\}$. It follows that $g([\mathbf{q}])$ defines a symmetric matrix with a simple minimum-eigenvalue (i.e., $g([\mathbf{q}]) \in \mathbb{S}^4_\lambda$) and the eigenspace associated with the minimum eigenvalue of 0 is precisely $\mathrm{span}(\mathbf{q}) = \mathrm{span}(-\mathbf{q})$. This in turn implies that:

$$\pm\mathbf{q} = \underset{\mathbf{q}\in S^3}{\operatorname{argmin}}\, \mathbf{q}^\mathsf{T} g\left([\mathbf{q}]\right)\mathbf{q}, \tag{13}$$

and therefore $f\left(g\left([\mathbf{q}]\right)\right) = [\mathbf{q}]$ so that $g(\cdot)$ is a global section of the surjective map $f(\cdot)$. Furthermore, we can see by inspection that this global section is smooth (i.e., continuous and differentiable) since we can always represent $g(\cdot)$ locally using one of the diffeomorphic preimages of $\mathbb{RP}^3$ in S^3 as the smooth function $g_0(\mathbf{q}) = \mathbf{I} - \mathbf{q}\mathbf{q}^\mathsf{T}$. ∎

C. $\mathbf{A}$ and the Bingham Distribution

We can further show that our representation space, $\mathbf{A}(\boldsymbol{\theta}) \in \mathbb{S}^4_\lambda$, defines a Bingham distribution over unit quaternions. Consequently, we may regard $\mathbf{A}$ as encoding a *belief* over rotations which facilitates the training of rotation models with uncertainty. The Bingham distribution is an antipodally symmetric distribution that is derived from a zero-mean Gaussian in $\mathbb{R}^{d+1}$ conditioned to lie on the unit hypersphere, S^d [3]. For unit quaternions ($d = 3$), the probability density function of $\mathbf{x} \sim \textsc{Bingham}\left(\mathbf{D}, \boldsymbol{\Lambda}\right)$ is

$$p(\mathbf{x}; \mathbf{D}, \boldsymbol{\Lambda}) = \frac{1}{N(\boldsymbol{\Lambda})} \exp\left(\sum_{i=1}^{3} \lambda_i^{\mathrm{dc}} (\mathbf{d}_i^\mathsf{T}\mathbf{x})^2\right) \tag{14}$$

$$= \frac{1}{N(\boldsymbol{\Lambda})} \exp\left(\mathbf{x}^\mathsf{T}\mathbf{D}\boldsymbol{\Lambda}\mathbf{D}^\mathsf{T}\mathbf{x}\right), \tag{15}$$

where $\mathbf{x} \in S^3$, $N(\boldsymbol{\Lambda})$ is a normalization constant, and $\mathbf{D} \in O(4)$ is an orthogonal matrix formed from the three orthogonal unit column vectors $\mathbf{d}_i$ and a fourth mutually orthogonal unit vector, $\mathbf{d}_4$. The matrix of dispersion coefficients, $\boldsymbol{\Lambda}$, is given by $\mathrm{diag}\left(\lambda_1^{\mathrm{dc}}, \lambda_2^{\mathrm{dc}}, \lambda_3^{\mathrm{dc}}, 0\right)$ with $\lambda_1^{\mathrm{dc}} \leq \lambda_2^{\mathrm{dc}} \leq \lambda_3^{\mathrm{dc}} \leq 0$ (note that these dispersion coefficients are eigenvalues of the matrix $\mathbf{D}\boldsymbol{\Lambda}\mathbf{D}^\mathsf{T}$). Each λ_i^{dc} controls the spread of the probability mass along the direction given by $\mathbf{d}_i$ (a small magnitude λ^{dc} implying a large spread and vice-versa). The mode of the distribution is given by $\mathbf{d}_4$.

Crucially, $\textsc{Bingham}\left(\mathbf{D}, \boldsymbol{\Lambda}\right) = \textsc{Bingham}\left(\mathbf{D}, \boldsymbol{\Lambda} + c\mathbf{I}\right)$ for all $c \in \mathbb{R}$ [8]. Thus, a careful diagonalization of our representation, $\mathbf{A} \in \mathbb{S}^4_\lambda$, fully describes $\textsc{Bingham}\left(\mathbf{D}, \boldsymbol{\Lambda}\right)$. Namely, to ensure that the mode of the density is given by the solution of Problem 3, we set $\mathbf{D}\boldsymbol{\Lambda}\mathbf{D}^\mathsf{T} = -\mathbf{A}$ since the smallest eigenvalue of $\mathbf{A}$ is the largest eigenvalue of $-\mathbf{A}$.

To recover the dispersion coefficients, λ_i^{dc}, we evaluate the non-zero eigenvalues of $-\mathbf{A} + \lambda_1\mathbf{I}$ (defining the equivalent density $\textsc{Bingham}\left(\mathbf{D}, \boldsymbol{\Lambda} + \lambda_1\mathbf{I}\right)$) where λ_i are the eigenvalues of $\mathbf{A}$ in ascending order as in Equation (1). Then, $\{\lambda_1^{\mathrm{dc}}, \lambda_2^{\mathrm{dc}}, \lambda_3^{\mathrm{dc}}\} = \{-\lambda_4 + \lambda_1, -\lambda_3 + \lambda_1, -\lambda_2 + \lambda_1\}$. This establishes a relation between the eigenvalue *gaps* of $\mathbf{A}$ and the dispersion of the Bingham density defined by $\mathbf{A}$.

IV. Using $\mathbf{A}$ with Deep Learning

We consider applying our formulation to the learning task of fitting a set of parameters, $\boldsymbol{\pi}$, such that the deep rotation regressor $\mathbf{R} = \mathtt{NN}(\mathbf{x}; \boldsymbol{\pi})$ minimizes a training loss $\mathcal{L}$ while generalizing to unseen data (as depicted in Figure 1).

A. Self-Supervised Learning

In many self-supervised learning applications, one requires a rotation matrix to transform one set of data onto another. Our representation admits a differentiable transformation into a rotation matrix though $\mathbf{R}^* = \mathbf{R}(\pm\mathbf{q}^*)$, where $\mathbf{R}(\mathbf{q})$ is the unit quaternion to rotation matrix projection (e.g., Eq. 4 in [41]). Since our layer solves for a pair of antipodal unit quaternions ($\pm\mathbf{q}^* \in \mathbb{RP}^3 \cong \mathrm{SO}(3)$), and therefore admits a smooth global section, we can avoid the discontinuity identified in [41] during back-propagation. In this work, however, we limit our attention to the supervised rotation regression problem to directly compare with the continuous representation of [41].

B. Supervised Learning: Rotation Loss Functions

For supervised learning over rotations, there are a number of possible choices for loss functions that are defined over $\mathrm{SO}(3)$. A survey of different bi-invariant metrics which are suitable for this task is presented in [16]. For example, four

possible loss functions include,

$$\mathcal{L}_{\text{quat}}\left(\mathbf{q}, \mathbf{q}_{\text{gt}}\right) = d_{\text{quat}}\left(\mathbf{q}, \mathbf{q}_{\text{gt}}\right)^2, \tag{16}$$

$$\mathcal{L}_{\text{chord}}\left(\mathbf{R}, \mathbf{R}_{\text{gt}}\right) = d_{\text{chord}}\left(\mathbf{R}, \mathbf{R}_{\text{gt}}\right)^2, \tag{17}$$

$$\mathcal{L}_{\text{ang}}\left(\mathbf{R}, \mathbf{R}_{\text{gt}}\right) = d_{\text{ang}}\left(\mathbf{R}, \mathbf{R}_{\text{gt}}\right)^2, \tag{18}$$

$$\mathcal{L}_{\text{BINGHAM}}\left(\mathbf{D}, \mathbf{\Lambda}, \mathbf{q}_{\text{gt}}\right) = \mathbf{q}_{\text{gt}}^{\mathsf{T}}\mathbf{D}\mathbf{\Lambda}\mathbf{D}^{\mathsf{T}}\mathbf{q}_{\text{gt}} + N(\mathbf{\Lambda}), \tag{19}$$

where[3]

$$d_{\text{quat}}\left(\mathbf{q}, \mathbf{q}_{\text{gt}}\right) = \min\left(\left\|\mathbf{q}_{\text{gt}} - \mathbf{q}\right\|_2, \left\|\mathbf{q}_{\text{gt}} + \mathbf{q}\right\|_2\right), \tag{20}$$

$$d_{\text{chord}}\left(\mathbf{R}, \mathbf{R}_{\text{gt}}\right) = \left\|\mathbf{R}_{\text{gt}} - \mathbf{R}\right\|_{\text{F}}, \tag{21}$$

$$d_{\text{ang}}\left(\mathbf{R}, \mathbf{R}_{\text{gt}}\right) = \left\|\text{Log}\left(\mathbf{R}\mathbf{R}_{\text{gt}}^{\mathsf{T}}\right)\right\|, \tag{22}$$

and $\text{Log}\left(\cdot\right)$ is defined as in [32]. Since our formulation fully describes a Bingham density, it is possible to use the likelihood loss, $\mathcal{L}_{\text{BINGHAM}}$, to train a Bingham belief (in a similar manner to [13] who use an alternate 19-parameter representation). However, the normalization constant $N(\mathbf{\Lambda})$ is a hypergeometric function that is non-trivial to compute. The authors of [13] evaluate this constant using a fixed-basis non-linear approximation aided by a precomputed look-up table. In this work, we opt instead to compare to other point representations and leave a comparison of different belief representations to future work. Throughout our experiments, we use the chordal loss $\mathcal{L}_{\text{chord}}$ which can be applied to both rotation matrix and unit quaternion outputs.[4] However, despite eschewing a likelihood loss, we can still extract a useful notion of uncertainty from deep neural network regression using the eigenvalues of our matrix $\mathbf{A}$. To see why, we present a final interpretation of our representation specifically catered to the structure of deep models.

V. $\mathbf{A}$ and Rotation Averaging

We take inspiration from [30] wherein neural-network-based pose regression is related to an interpolation over a set of base poses. We further elucidate this connection by specifically considering rotation regression and relating it to rotation averaging over a (learned) set of base rotations using the chordal distance. Since these base rotations must span the training data, we argue that $\mathbf{A}$ can represent a notion of epistemic uncertainty (i.e., distance to training samples) without an explicit likelihood loss.

Consider that given N rotation samples expressed as unit quaternions $\mathbf{q}_i$, the rotation average according to the chordal metric can be computed as [16]:

$$\overline{\mathbf{q}}_{d_{\text{chord}}} = \underset{\mathbf{q}\in S^3}{\text{argmin}} \sum_i^N d_{\text{chord}}(\mathbf{q}, \mathbf{q}_i) = f\left(-\sum_i^N \mathbf{q}_i\mathbf{q}_i^{\mathsf{T}}\right), \tag{23}$$

where $f(\cdot)$ is defined by[5] Equation (10). A weighted averaging version of this operation is discussed in [22]. Next, consider a feed-forward neural network that uses our representation by regressing ten parameters, $\boldsymbol{\theta} \in \mathbb{R}^{10}$. If such a network has a final fully-connected layer prior to its output, we can separate it into two components: (1) the last layer parameterized by the weight matrix $\mathbf{W} \in \mathbb{R}^{10\times N}$ and the bias vector $\mathbf{b} \in \mathbb{R}^{10}$, and (2) the rest of the network $\boldsymbol{\gamma}(\cdot)$ which transforms the input $\mathbf{x}$ (e.g., an image) into N coefficients given by γ_i. The output of such a network is then given by

$$\mathbf{q}^* = f(\mathbf{A}(\boldsymbol{\theta}(\mathbf{x}))) = f\left(\mathbf{A}\left(\sum_i^N \mathbf{w}_i\gamma_i(\mathbf{x}) + \mathbf{b}\right)\right) \tag{24}$$

$$= f\left(\sum_i^N \mathbf{A}(\mathbf{w}_i)\gamma_i(\mathbf{x}) + \mathbf{A}(\mathbf{b})\right) \tag{25}$$

where $\mathbf{w}_i$ refers to the ith column of $\mathbf{W}$ and the second line follows from the linearity of the mapping defined in Equation (8). In this manner, we can view rotation regression with our representation as analogous to computing a weighted chordal average over a set of learned base orientations (parameterized by the symmetric matrices defined by the column vectors $\mathbf{w}_i$ and $\mathbf{b}$). During training the network tunes both the bases and the weight function $\boldsymbol{\gamma}(\cdot)$.[6]

A. Dispersion Thresholding (DT) as Epistemic Uncertainty

The positive semi-definite (PSD) matrix $\sum_i^N \mathbf{q}_i\mathbf{q}_i^{\mathsf{T}}$ is also called the *inertia* matrix in the context of Bingham maximum likelihood estimation because the operation $f\left(-\sum_i^N \mathbf{q}_i\mathbf{q}_i^{\mathsf{T}}\right)$ can be used to compute the maximum likelihood estimate of the mode of a Bingham belief given N samples [14]. Although our symmetric representation $\mathbf{A}$ is not necessarily PSD, we can perform the same canonicalization as in Section III-C, and interpret $\mathbf{\Lambda} = -\mathbf{A} + \lambda_1\mathbf{I}$ as the (negative) inertia matrix. Making the connection to Bingham beliefs, we then use

$$\text{tr}\left(\mathbf{\Lambda}\right) = \sum_i^3 \lambda_i^{\text{dc}} = 3\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 \tag{26}$$

as a measure of network uncertainty.[7] We find empirically that this works surprisingly well to measure epistemic uncertainty (i.e., model uncertainty) without any explicit penalty on $\mathbf{\Lambda}$ during training. To remove OOD samples, we compute a threshold on $\text{tr}\left(\mathbf{\Lambda}\right)$ based on quantiles over the training data (i.e., retaining the lowest qth quantile). We call this technique *dispersion thresholding* or DT. Despite the intimate connection to both rotation averaging and Bingham densities, further work is required to elucidate why exactly this notion of uncertainty is present without the use of an uncertainty-aware loss like $\mathcal{L}_{\text{BINGHAM}}$. We leave a thorough investigation for future work, but note that this metric can be related to

[3]Note that it is possible to relate all three metrics without converting between representations–e.g., $d^2_{\text{chord}}\left(\mathbf{R}(\mathbf{q}), \mathbf{R}(\mathbf{q}_{\text{gt}})\right) = 2d^2_{\text{quat}}(4 - d^2_{\text{quat}})$.

[4]The chordal distance has also been shown to be effective for initialization and optimization in SLAM [5, 29].

[5]The negative on $\mathbf{q}_i\mathbf{q}_i^{\mathsf{T}}$ is necessary since f computes the eigenvector associated with λ_{min} whereas the average requires λ_{max}.

[6]We can make a similar argument for the unit quaternion representation since the normalization operation $f_n(\mathbf{y}) = \mathbf{y}\left\|\mathbf{y}\right\|^{-1}$ corresponds to the mean $\overline{\mathbf{q}}_{d_{\text{quat}}} = \text{argmin}_{\mathbf{q}\in S^3}\sum_i^N d_{\text{quat}}(\mathbf{q}, \mathbf{q}_i) = f_n\left(\sum_i^N \mathbf{q}_i\right)$.

[7]Note that under this interpretation the matrix $\mathbf{\Lambda}$ does not refer directly to the Bingham dispersion matrix parameterized by the inertia matrix, but the two can be related by inverting an implicit equation–see [14].

the norm $\left\|\sum_i^N \mathbf{w}_i \gamma_i(\mathbf{x}) + \mathbf{b}\right\|$ which we find empirically to also work well as an uncertainty metric for other rotation representations. We conjecture that the 'basis' rotations $\mathbf{w}_i, \mathbf{b}$ must have sufficient spread over the space to cover the training distribution. During test-time, OOD inputs, $\mathbf{x}_{\text{OOD}}$, result in weights, $\gamma_i(\mathbf{x}_{\text{OOD}})$, which are more likely to 'average out' these bases due to the compact nature of $\text{SO}(3)$. Conversely, samples closer to training data are more likely to be nearer to these learned bases and result in larger $|\text{tr}\,(\mathbf{\Lambda})|$.

VI. Experiments

We present the results of extensive synthetic and real experiments to validate the benefits of our representation. In each case, we compare three[8] representations of $\text{SO}(3)$: (1) unit quaternions (i.e., a normalized 4-vector, as outlined in Figure 1), (2) the best-performing continuous six-dimensional representation, `6D`, from [41], and (3) our own symmetric-matrix representation, $\mathbf{A}$. We report all rotational errors in degrees based on $d_{\text{ang}}(\cdot,\cdot)$.

A. Wahba Problem with Synthetic Data

First, we simulated a dataset where we desired a rotation from two sets of unit vectors with known correspondence. We considered the generative model,

$$\mathbf{v}_i = \hat{\mathbf{R}}\mathbf{u}_i + \boldsymbol{\epsilon}_i\ , \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right), \tag{27}$$

where $\mathbf{u}_i$ are sampled from the unit sphere. For each training and test example, we sampled $\hat{\mathbf{R}}$ as $\text{Exp}\left(\hat{\boldsymbol{\phi}}\right)$ (where we define the capitalized exponential map as [32]) with $\hat{\boldsymbol{\phi}} = \phi\frac{\mathbf{a}}{\|\mathbf{a}\|}$ and $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\phi \sim U[0, \phi_{\text{max}})$, and set $\sigma = 0.01$. We compared the training and test errors for different learning rates in selected range using the `Adam` optimizer. Taking inspiration from [41], we employed a dynamic training set and constructed each mini-batch from 100 sampled rotations with 100 noisy matches, $\mathbf{u}_i, \mathbf{v}_i$, each. We defined an epoch as five mini-batches. Our neural network structure mimicked the convolutional structure presented in [41] and we used $\mathcal{L}_{\text{chord}}$ to train all models.

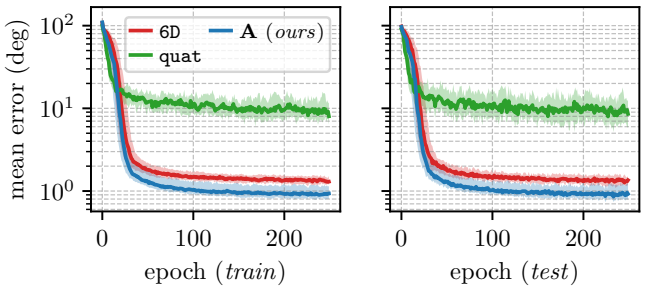


Fig. 3: Angular errors for 25 different trials each with learning rates sampled from the range $\{10^{-4}, 10^{-3}\}$ (log-uniform) and $\phi_{\text{max}} = 180°$. We show $\{10, 50, 90\}^{\text{th}}$ percentiles at each epoch.

[8]We note that regressing 3×3 rotation matrices directly would also satisfy the continuity property of [41] but we chose not to include this as the `6D` representation fared better in the experiments of [41].

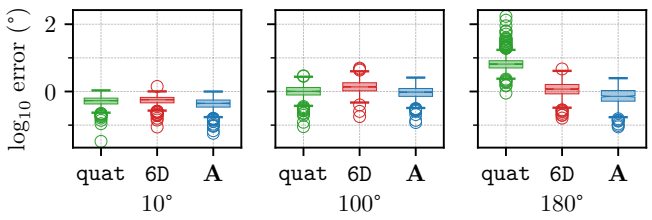
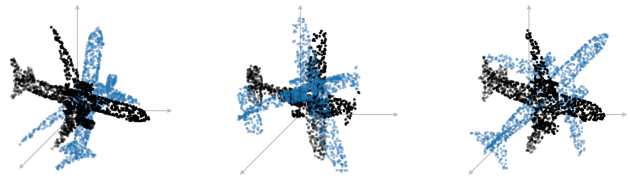
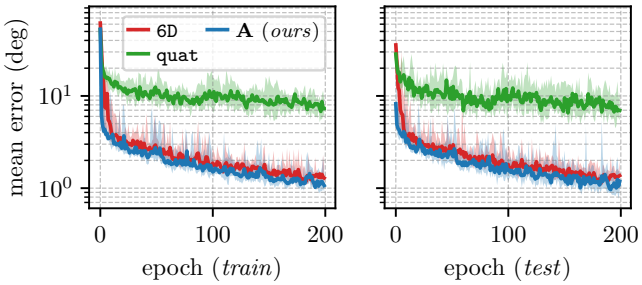


Fig. 4: Box and whiskers plots for three different settings of ϕ_{max} for three rotation representations applied to synthetic data. The unit quaternion representation results in large errors as $\phi_{\text{max}} \to 180°$.



(a) We sample the ShapeNet `airplane` category and randomly rotate point clouds to generate our training and test data.



(b) Mean angular errors for 10 different ShapeNet dataset trials each with learning rates sampled in the range $\{10^{-4}, 10^{-3}\}$ (log-uniform). We plot $\{10, 50, 90\}^{\text{th}}$ percentiles at each epoch.

Fig. 5: A summary of our ShapeNet experiments.

Figure 3 displays the results of 25 experimental trials with different learning rates on synthetic data. For arbitrary rotation targets, both continuous representations outperform the discontinuous unit quaternions, corroborating the results of [41]. Moreover, our symmetric representation achieves the lowest errors across training and testing. Figure 4 depicts the performance of each representation on training data restricted to different maximum angles. As hypothesized in [41], the discontinuity of the unit quaternion manifests itself on regression targets with angles of magnitude near 180 degrees.

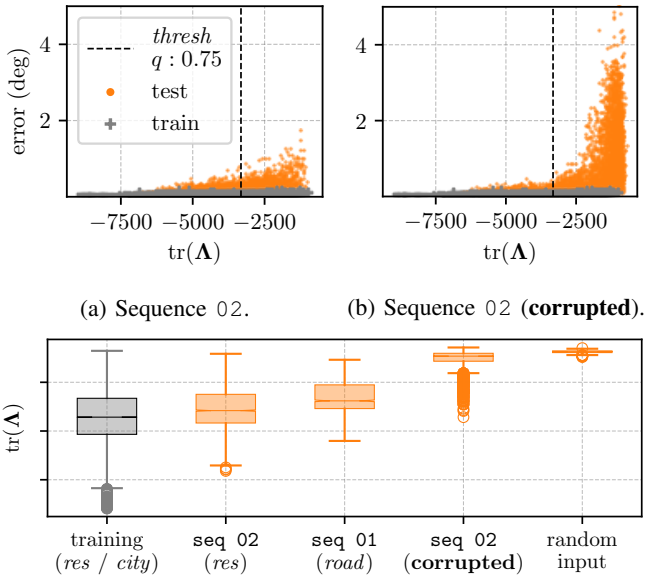
B. Wahba Problem with ShapeNet

Next, we recreated the experiment from [41] on 2,290 airplane point clouds from ShapeNet [6], with 400 held-out point clouds. During each iteration of training we randomly selected a single point cloud and transformed it with 10 sampled rotation matrices (Figure 5a). At test time, we applied 100 random rotations to each of the 400 held-out point clouds. Figure 5b compares the performance of our representation against that of unit quaternions and the `6D` representation, with results that are similar to the synthetic case in Figure 3.

TABLE I: Relative rotation learning on the KITTI dataset with different representations of $\mathrm{SO}(3)$. All training is done on uncorrupted data. We show that our OOD technique, DT, can dramatically improve rotation accuracy by rejecting inputs that are likely to lead to high error.

		Normal Test		**Corrupted Test (50%)**		
Sequence	Model	Mean Error (°)	Kept (%)	Mean Error (°)	Kept (%)	Precision[3](%)
`00` (4540 pairs)	`quat`	0.16	100	0.74	100	—
	`6D` [41]	0.17	100	0.68	100	—
	`6D` + auto-encoder (*AE*)[1]	0.16	32.0	0.61	19.1	57.4
	A *(ours)*	0.17	100	0.71	100	—
	A + DT (Section V-A)[2]	**0.12**	69.5	**0.12**	37.3	**99.00**
`02` (4660 pairs)	`quat`	0.16	100	0.64	100	—
	`6D`	0.15	100	0.69	100	—
	`6D` + AE[1]	0.19	15.4	0.47	9.9	77.83
	A	0.16	100	0.72	100	—
	A + DT[2]	**0.12**	70.1	**0.11**	34.0	**99.50**
`05` (2760 pairs)	`quat`	0.13	100	0.72	100	—
	`6D`	0.11	100	0.76	100	—
	`6D` + AE[1]	0.10	41.6	0.40	27.7	76.05
	A	0.12	100	0.72	100	—
	A + DT[2]	**0.09**	79.1	**0.10**	39.2	**97.41**

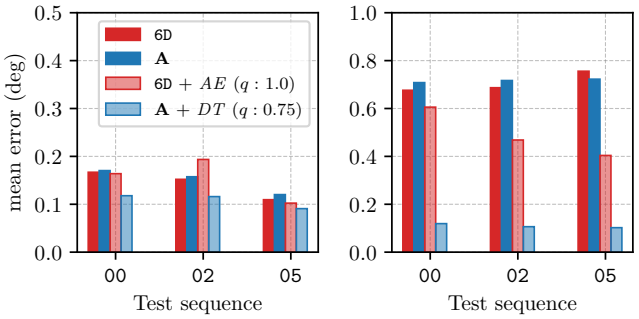
[1] Thresholding based on $q = 1.0$. [2] Thresholding based on $q = 0.75$. [3] % of corrupted images that are rejected.



(a) Sequence `02`. (b) Sequence `02` (**corrupted**).

(c) Different test sets.

Fig. 6: The dispersion thresholding metric, $\mathrm{tr}\,(\mathbf{\Lambda})$, plotted for data corresponding to test sequence `02` (refer to text for train/test split). Corruption is applied to the test data without altering the training data. DT thresholding leads to an effective OOD rejection scheme without the need to retrain the model.



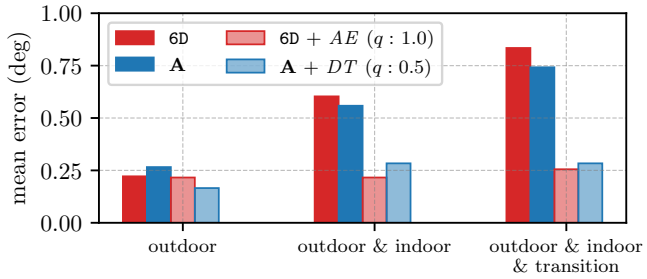
(a) Uncorrupted images. (b) Corrupted images.

Fig. 7: Mean rotation errors for three different test sequences from the KITTI odometry dataset. Since these relative rotation targets are 'small', we see no significant difference in baseline accuracy. However, our dispersion thresholding technique (Section V-A) can significantly improve performance, and outperforms an alternative OOD method based on reconstruction error using an auto-encoder (AE). See Table I for full statistics.

C. Visual Rotation-Only Egomotion Estimation: KITTI

Third, we used our representation to learn relative rotation from sequential images from the KITTI odometry dataset [12]. By correcting or independently estimating rotation, these learned models have the potential to improve classical visual odometry pipelines [25]. We note that even in the limit of no translation, camera rotation can be estimated independently of translation from a pair of images [19]. To this end, we built a convolutional neural network that predicted the relative camera orientation between sequential images recorded on sequences from the *residential* and *city* categories in the dataset. We selected sequences `00`, `02`, and `05` for testing and trained three models with the remaining sequences for each. In accord with the results in Figure 4, we found that there was little change in performance across different $\mathrm{SO}(3)$ representations since rotation magnitudes of regression targets from KITTI are typically on the order of one degree. However, we found that our DT metric acted as a useful measure of epistemic uncertainty. To

(a) MAV with a Point Grey Flea3 global shutter camera (IV) in three environments: outdoor (I), indoor (II) and transition (III).



(b) Mean rotation errors for three different test sequences from the MAV dataset, using a model trained on outdoor data. The auto-encoder rejection performs well, yet our DT technique is able to match its performance without requiring a separate model.

Fig. 8: A summary of our MAV experiments.

further validate this notion, we manually corrupted the KITTI test images by setting random rectangular regions of pixels to uniform black. Figure 6c displays the growth in magnitude that is manifest in the DT metric as data becomes less similar to that seen during training. Figure 6 displays the estimation error for test sequence `02` with and without corruption. We stress that these corruptions are only applied to the test data; as we highlight numerically in Table I, DT is able to reject corrupted images and other images that are likely to lead to high test error. Indeed, in all three test sequences, we observed a nearly constant mean rotation error for our formulation ($\mathbf{A}$ + DT) with and without data corruption.

1) Auto-Encoder (AE) OOD Method: We compared our DT thresholding approach with an auto-encoder-based OOD technique inspired by work in novelty detection in mobile robotics [1, 28]. For each training set, we trained an auto-encoder using an L_1 pixel-based reconstruction loss, and then rejected test-time inputs whose mean pixel reconstruction error is above the qth percentile in training. Figure 7 and Table I detail results that demonstrate that our representation paired with DT performs significantly better than the `6D` representation with an AE-based rejection method. Importantly, we stress that our notion of uncertainty is embedded within the representation and does not require the training of an auxiliary OOD classifier.

D. MAV Indoor-Outdoor Dataset

Finally, we applied our representation to the task of training a relative rotation model on data collected using a Flea3 global-shutter camera mounted on the Micro Aerial Vehicle depicted in Figure 8a. We considered a dataset in which the vehicle undergoes dramatic lighting and scene changes as it transitions from an outdoor to an indoor environment. We trained a model using outdoor images with ground-truth rotation targets supplied by an onboard visual-inertial odometry system (note that since we were primarily interested in characterizing our dispersion thresholding technique, such coarse rotation targets sufficed), and an identical network architecture to that used in the KITTI experiments. Figure 8b details the performance of our representation against the `6D` representation paired with an auto-encoder OOD rejection method. We observe that, compared to the KITTI experiment, the AE-based OOD rejection technique fares much better on this data. We believe this is a result of the way we partitioned our MAV dataset; images were split into a train/test split using a random selection, so test samples were recorded very near (temporally and spatially) to training samples. Nevertheless, our method, $\mathbf{A}$ + DT, performs on par with `6D` + AE on all three test sets, but does not require the training of a separate classifier.

VII. Discussion and Limitations

Rotation representation is an important design criterion for state estimation and no single representation is optimal in all contexts; ours is no exception. Importantly, the differentiable layer which solves Problem 3 incurs some computational cost. This cost is negligible at test-time but can slow learning during training when compared to other representations that require only basic operations like normalization. In practice, we find that for common convolutional networks, training is bottlenecked by other parts of the learning pipeline and our representation adds marginal processing time. For more compact models, however, training time can be increased. Further, our representation does not include symmetric matrices where the minimal eigenvalue is non-simple. In this work, we do not enforce this explicitly; instead, we assume that this will not happen to within machine precision. In practice, we find this occurs exceedingly rarely, though explicitly enforcing this constraint is a direction of future research.

VIII. Conclusions and Future Work

In this work, we presented a novel representation of $\mathrm{SO}(3)$ based on a symmetric matrix $\mathbf{A}$. Our representation space can be interpreted as a data matrix of a QCQP, as defining a Bingham belief over unit quaternions, or as parameterizing a weighted rotation average over a set of base rotations. Further, we proved that this representation admits a smooth global section of $\mathrm{SO}(3)$ and developed an OOD rejection method based solely on the eigenvalues of $\mathbf{A}$. Avenues for future work include combining our representation with a Bingham likelihood loss, and further investigating the connection between $\mathbf{A}$, epistemic uncertainty, and rotation averaging. Finally, we are especially interested in leveraging our representation to improve the reliability and robustness (and ultimately, safety) of learned perception algorithms in real-world settings.

REFERENCES

[1] Alexander Amini, Wilko Schwarting, Guy Rosman, Brandon Araki, Sertac Karaman, and Daniela Rus. Variational Autoencoder for End-to-End Control of Autonomous Driving with Novelty Detection and Training De-biasing. In *IEEE/RSJ Intl. Conf. Intelligent Robots and Systems (IROS)*, pages 568–575, 2018.

[2] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[3] Christopher Bingham. An Antipodally Symmetric Distribution on the Sphere. *The Annals of Statistics*, 2(6):1201–1225, 1974.

[4] Eric Brachmann and Carsten Rother. Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses. In *Intl. Conf. Computer Vision (ICCV)*, 2019.

[5] L Carlone, R Tron, K Daniilidis, and F Dellaert. Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In *IEEE Intl. Conf. Robotics and Automation (ICRA)*, pages 4597–4604, 2015.

[6] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], 2015.

[7] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J. Davison. Learning to solve nonlinear least squares for monocular stereo. In *European Conf. Computer Vision (ECCV)*, 2018.

[8] J. E. Darling and K. J. DeMars. Uncertainty Propagation of correlated quaternion and Euclidean states using partially-conditioned Gaussian mixtures. In *Intl. Conf. Information Fusion (FUSION)*, pages 1805–1812, 2016.

[9] James Diebel. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. Technical report, Stanford University, 2006.

[10] Bernard Etkin. *Dynamics of atmospheric flight*. Wiley, 1972.

[11] Yarin Gal. *Uncertainty in Deep Learning*. PhD Thesis, University of Cambridge, 2016.

[12] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. *Intl. J. Robotics Research*, 32(11):1231–1237, 2013.

[13] Igor Glitschenski, Wilko Schwarting, Roshni Sahoo, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep Orientation Uncertainty Learning based on a Bingham Loss. In *Intl. Conf. Learning Representations (ICLR)*, 2020.

[14] Jared Glover and Leslie Pack Kaelbling. Tracking 3-D Rotations with the Quaternion Bingham Filter. Technical Report MIT-CSAIL-TR-2013-005, MIT, 2013.

[15] Reinhard M. Grassmann and Jessica Burgner-Kahrs. On the Merits of Joint Space and Orientation Representations in Learning the Forward Kinematics in SE(3). In *Robotics: Science and Systems (RSS)*, 2019.

[16] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation Averaging. *Intl. J. Computer Vision*, 103(3):267–305, 2013.

[17] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Optical Society of America*, 4(4):629, 1987.

[18] Peter Karkus, Xiao Ma, David Hsu, Leslie Kaelbling, Wee Sun Lee, and Tomas Lozano-Perez. Differentiable Algorithm Networks for Composable Robot Learning. In *Robotics: Science and Systems (RSS)*, 2019.

[19] Laurent Kneip, Roland Siegwart, and Marc Pollefeys. Finding the Exact Rotation between Two Images Independently of the Translation. In *European Conf. Computer Vision (ECCV)*, pages 696–709, 2012.

[20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6405–6416, 2017.

[21] Jan R. Magnus. On Differentiating Eigenvalues and Eigenvectors. *Econometric Theory*, 1(2):179–191, 1985.

[22] F. Landis Markley, Yang Cheng, John L. Crassidis, and Yaakov Oshman. Averaging Quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007.

[23] Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. In *Intl. Conf. Learning Representations (ICLR)*, 2020.

[24] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4026–4034, 2016.

[25] Valentin Peretroukhin and Jonathan Kelly. DPC-Net: Deep Pose Correction for Visual Localization. *IEEE Robotics and Automation Letters*, 3(3):2424–2431, 2018.

[26] Valentin Peretroukhin, Brandon Wagstaff, and Jonathan Kelly. Deep Probabilistic Regression of Elements of SO(3) using Quaternion Averaging and Uncertainty Injection. In *CVPR'19 Workshop on Uncertainty and Robustness in Deep Visual Learning*, pages 83–86, 2019.

[27] René Ranftl and Vladlen Koltun. Deep Fundamental Matrix Estimation. In *European Conf. Computer Vision (ECCV)*, volume 11205, pages 292–309, 2018.

[28] Charles Richter and Nicholas Roy. Safe Visual Navigation via Deep Learning and Novelty Detection. In *Robotics: Science and Systems (RSS)*, 2017.

[29] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *Intl. J. Robotics Research*, 38(2-3):95–125, 2019.

[30] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the Limitations of CNN-Based Absolute Camera Pose Regression. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3297–3307, 2019.

[31] Davide Scaramuzza and Friedrich Fraundorfer. Visual Odometry [Tutorial]. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.

[32] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics. *arXiv:1812.01537 [cs]*, 2018.

[33] Suvrit Sra. Directional Statistics in Machine Learning: A Brief Review. *arXiv:1605.00316 [stat]*, 2016.

[34] Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment networks. In *Intl. Conf. Learning Representations (ICLR)*, 2019.

[35] Grace Wahba. Problem 65-1: A Least Squares Estimate of Satellite Attitude. *SIAM Review*, 7(3):409–409, 1965.

[36] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *Intl. Conf. Computer Vision (ICCV)*, 2019.

[37] Bong Wie and Peter M. Barba. Quaternion feedback for spacecraft large angle maneuvers. *Journal of Guidance, Control, and Dynamics*, 8(3): 360–365, 1985.

[38] Heng Yang and Luca Carlone. A Quaternion-based Certifiably Optimal Solution to the Wahba Problem with Outliers. In *Intl. Conf. Computer Vision (ICCV)*, pages 1665–1674, 2019.

[39] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to Find Good Correspondences. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2666–2674, 2018.

[40] Brady Zhou, Philipp Krähenbühl, and Vladlen Koltun. Does computer vision matter for action? *Science Robotics*, 4(30), 2019.

[41] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the Continuity of Rotation Representations in Neural Networks. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 5745–5753, 2019.