Eliminating Topological Errors in Neural Network Rotation Estimation Using Self-selecting Ensembles

SITAO XIANG, University of Southern California, USA

Many problems in computer graphics and computer vision applications involves inferring a rotation from a variety of different forms of inputs. With the increasing use of deep learning, neural networks have been employed to solve such problems. However, the traditional representations for 3D rotations, the quaternions and Euler angles, are found to be problematic for neural networks in practice, producing seemingly unavoidable large estimation errors. Previous researches has identified the discontinuity of the mapping from $\mathrm{SO}(3)$ to the quaternions or Euler angles as the source of such errors, and to solve it, embeddings of $\mathrm{SO}(3)$ have been proposed as the output representation of rotation estimation networks instead. In this paper, we argue that the argument against quaternions and Euler angles from local discontinuities of the mappings from $\mathrm{SO}(3)$ is flawed, and instead provide a different argument from the global topological properties of $\mathrm{SO}(3)$ that also establishes the lower bound of maximum error when using quaternions and Euler angles for rotation estimation networks. Extending from this view, we discover that rotation symmetries in the input object causes additional topological problems that even using embeddings of $\mathrm{SO}(3)$ as the output representation would not correctly handle. We propose the self-selecting ensemble, a topologically motivated approach, where the network makes multiple predictions and assigns weights to them. We show theoretically and with experiments that our methods can be combined with a wide range of different rotation representations and can handle all kinds of finite symmetries in 3D rotation estimation problems.

CCS Concepts: • **Mathematics of computing** → **Algebraic topology**; • **Computing methodologies** → *Neural networks*; *Computer graphics.*

Additional Key Words and Phrases: rotation, neural networks, covering space

ACM Reference Format:
Sitao Xiang. 2021. Eliminating Topological Errors in Neural Network Rotation Estimation Using Self-selecting Ensembles. *ACM Trans. Graph.* 40, 4, Article 167 (August 2021), 21 pages. https://doi.org/10.1145/3450626.3459882

1 INTRODUCTION

The modeling of objects is one of the central problems in computer graphics and computer vision, and in many applications, part of this involves inferring a rotation. And in recent years, neural networks have increasingly been employed for solving such problems. The range of applications include object rotation estimation [Mahendran et al. 2017; Su et al. 2015; Xiang et al. 2018], camera parameter estimation [Poursaeed et al. 2018], registration of 3D scans [Pais et al. 2020] and estimation of human pose [Zhou et al. 2016], to name a few. However, difficulties has been encountered, in that the network seems unable to avoid large rotation estimation errors in excess of 100° in certain cases, as observed in [Xiang et al. 2018], for example.

There have been attempts to explain this error and find solutions. [Zhou et al. 2019] is among the first works to attempt a theoretical analysis of this problem. The authors noticed that the functions computed by neural networks are continuous, while the correct functions mapping a rotation to its quaternion or Euler angle representation are not continuous. From this, they defined a notion of continuity of rotation representations, and proposed a 6D and a 5D embedding of $\mathrm{SO}(3)$ which is shown to be continuous, thus making it suitable for use in neural networks.

This has inspired interest in the study of rotation representations, and further proposals for rotation representation with nice mathematical properties have been proposed. For example, in [Levinson et al. 2020] the long known optimal matrix orthogonalization method using Singular Value Decomposition (SVD) has been shown to be a favourable rotation representation in neural networks, and in [Peretroukhin et al. 2020] the authors proposed using the output of the network to parametrize a Quadratically-Constrained Quadratic Program (QCQP), the solution of which is a rotation, thus providing a rotation representation in neural networks.

While we appreciate the past research, we feel that the theory was not built on solid ground. In particular, the proof of the non-viability of quaternion and Euler angle representations is by first identifying a specific mapping from a rotation to one of the correct representations, then locating a discontinuity of this mapping, thus showing that a continuous neural network cannot learn such a mapping. We feel that requiring the neural network to fit a pre-designated mapping from a rotation to one of its representations does not represent the best neural network training strategy. Yet this strategy is widely used. For example, in [Peretroukhin et al. 2020] the authors stated that 180° input rotations are especially problematic. In our view, there is nothing inherently problematic about 180° rotations. The large errors appear to be caused by 180° rotations only because the most commonly used pre-designated representation mapping has discontinuities at these points.

In addition, previous theories explained why continuous neural networks cannot compute quaternion and Euler angle representations exactly, but did not explain why the maximum error is always large.

Unsatisfied by these explanations, we approached this problem from a different view. By discarding the pre-designated representation mapping and giving the network freedom to choose its own mapping, we realized that local behaviors of the mapping from $\mathrm{SO}(3)$ to the quaternions or the Euler angles, such as discontinuities, do not adequately address the non-existence of correct rotation estimation networks. Instead, the fundamental cause is the topological properties of the rotation group $\mathrm{SO}(3)$ and the group of unit

Author's address: Sitao Xiang, sitaoxia@usc.edu, University of Southern California, 3737 Watt Way, Los Angeles, California, USA, 90089.

0730-0301/2021/8-ART167
https://doi.org/10.1145/3450626.3459882

quaternions Sp(1), in particular the fact that Sp(1) is a non-trivial universal cover of SO(3) and the existence of non-contractible loops in SO(3).

Continuing along this path, we discovered that if the input to the rotation estimation problem has rotational symmetry, then the effective topology of the input space could change, which causes the previously proposed embeddings of SO(3) to become ineffective.

This prompted us to seek alternative solutions. In an attempt to eliminate the root of the problem which is non-contractible loops in the input space, we discovered that we can divide the input space into finitely many contractible subsets, so that each subset is free of non-contractible loops and a continuous mapping from this subset to the correct representation exists. Based on this, we propose the self-selecting ensemble, which is a universally effective, representation-independent method for rotation estimation, which is effective for a wide range of symmetries in combination with a wide range of rotation representations.

Specifically, in this paper, we

- Give a topological argument that continuous functions from a 3D rotation to its correct quaternion or Euler angle representation does not exist;
- Establish that a 3D rotation estimation network using quaternions and Euler angles as output representations is guaranteed to produce 180° error on some input;
- Prove that in the presence of symmetry, a 3D rotation estimation network using an embedding of SO(3) as output representation is also guaranteed to produce large error, with error bounds calculated for each different symmetry;
- Propose the self-selecting ensemble, which is provably able to avoid the occurrence of large errors and requires minimal additional computation and no additional supervision in training;
- Identify the necessary number of networks required in a successful ensemble under different combinations of input symmetries and output representations;
- Present experimental results that verify our theoretical analyses.

2 RELATED WORKS

2.1 Rotation Representations for Machine Learning and Their Continuity

Before the wide adoption of deep learning, the problem of rotation representation discontinuity in machine learning has been been noted in [Saxena et al. 2009]. Indeed, [Saxena et al. 2009] has already paid special attention to ambiguities caused by symmetric objects. However, they did not give a rigorous treatment of rotation representation continuity, in particular quaternions and Euler angles which are very commonly used, and their proposed solution for symmetric objects produces a different representation for each kind of symmetry. A more formal definition of the continuity of rotation representation is presented in [Zhou et al. 2019], and they showed that quaternion, Euler angle and axis-angle representations are indeed discontinuous. A 6D and a 5D representation are proposed, which meet their definition of a continuous representation and are shown to outperform the discontinuous representations in rotation estimation problems. Using the same definition of continuity, other continuous representations are proposed in later works: [Levinson et al. 2020] replaces the Gram–Schmidt orthogonalization in the 6D representation with SVD orthogonalization, and [Peretroukhin et al. 2020] represents a rotation as the solution of a Quadratically-Constrained Quadratic Program. However, these later works did not pay attention to complications caused by symmetric objects, and did not give an explanation of the occurrence of large error.

2.2 Rotation Estimation by Classification

Using classification for regression is not a new idea. In the area of rotation estimation, one way is to divide the range of each of the three Euler angles into buckets, and predict each one using classification [Su et al. 2015; Tulsiani and Malik 2015]. A precise estimation of these angles would then require a large number of buckets. To reduce the number of buckets, a hybrid approach combining classification and regression has been proposed in [Mahendran et al. 2018], where each bucket gives a mean value to which an offset is added. In these methods, the number of buckets is more or less arbitrary or determined by trial and error, where in our approach the number of functions in the ensemble is based on solid theoretical foundation and is provably optimal. In addition, the range of each bucket is learned automatically and not hand-designed. We also point out that in a classification-regression hybrid method, the continuity problem persists if the same offset function is shared by all buckets.

3 MOTIVATING EXAMPLE: 2D ROTATION ESTIMATION

For a better understanding of our theory and method, we would like to start our discussion from an extremely simple problem: 2D rotation estimation using neural networks. We have a two-dimensional shape that can rotate freely in the plane around its center. The shape does not have any symmetry. One particular position is considered "canonical". Given an image of some rotated version of this shape, we want to compute its rotation angle relative to the canonical position.

If the range of possible rotations is unconstrained and the network is assumed to be continuous, then it seems impossible to ensure that the network gives the correct output for all possible inputs. People have proposed explanations for this failure. In particular, one argument is that the "correct" function that maps a rotated shape to its rotation angle is not continuous, and thus a continuous network cannot compute such a function correctly for all inputs. A specific example of such an argument might be as follows:

Proof. Since the input is uniquely determined by a 2D rotation, let us identify an input with a rotation matrix $\left[\begin{smallmatrix} a & -b \\ b & a \end{smallmatrix}\right] \in \mathrm{SO}(2)$ with $a^2 + b^2 = 1$. We want the output angle to be in the range $(-\pi, \pi]$. Then, the correct function would be

$$\left(\begin{bmatrix} a & -b \\ b & a \end{bmatrix}\right) = \operatorname{atan2}(b, a) \tag{1}$$

In other words, if we let $R_2(\theta) = \left[\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right]$, then $f(R_2(\theta)) = \theta - 2\pi \cdot \lceil \frac{\theta-\pi}{2\pi} \rceil$ (Here the subscript 2 means two dimensional). f is not continuous at $R_2(\pi) = \left[\begin{smallmatrix} -1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$, since $\lim_{\theta\to\pi^-} f(R_2(\theta)) = \pi \neq -\pi = \lim_{\theta\to\pi^+} f(R_2(\theta))$.

Since the network is continuous, it cannot compute f correctly everywhere. □

We argue, however, that such a proof is flawed. It only proves the case where the range of output is required to be $(-\pi, \pi]$. Such a requirement is not justified: any two rotation angles that differ by an integer multiple of 2π refer to the same rotation. Instead of choosing one of the equivalent angles as “the” correct output, the output of the network should be considered correct as long as it refers to the same rotation as the input.

From a neural network training perspective, the flawed proof reflects the treatment of this problem as an ordinary regression problem where to each input is associated a “ground truth” output and the network is trained to learn the mapping from an input to the ground truth output. Specifically, the network, let us call N, is trained to minimize $\mathbb{E}_M[|N(M) - f(M)|]$, where the expectation is taken over some distribution over $\mathrm{SO}(2)$. Such a treatment removes the flexibility of the network to choose its own output range, thus potentially preventing it from learning a correct function should such functions exist.

In contrast, the most general training procedure should let the network minimize the difference between the output and the input “modulo 2π”, that is, the distance between two 2D rotations should be defined as:

$$d_2(R_2(\alpha), R_2(\beta)) = \min_{k\in\mathbb{Z}} |\alpha - \beta + 2k\pi| \tag{2}$$

This would no longer correspond to minimizing the difference of the network to any predetermined “ground truth” function. Since under this view the unique ground truth does not exist anymore, we cannot prove that a neural network that computes the rotation angle correctly for all inputs does not exist by simply picking one possible correct function and showing that it is not continuous. Instead we must prove that all functions that give correct rotation angles must have discontinuities, or equivalently, that all continuous functions must produce an error on some input.

In fact, we are able to achieve more than proving that a universally correct network does not exist: we can show that for any network, there exists an input such that the output of the network differs from the correct rotation angle by π, which is the largest possible difference between two 2D rotations. We have the following theorem:

Theorem 1. *For any continuous function $f : \mathrm{SO}(2) \to \mathbb{R}$, there exists $M \in \mathrm{SO}(2)$ such that $d_2(M, R_2(f(M))) = \pi$.*

Before introducing a rigorous proof, we examine an intuitive demonstration that will later help us understand the 3D case. In figure 1, at the top left is a 2D shape in its canonical position. Any angle of the form $2k\pi$ with $k \in \mathbb{Z}$ would be a correct rotation angle, and these are marked on the number line to the right with blue and green triangles. The red diamonds mark the angles $(2k + 1)\pi$ for $k \in \mathbb{Z}$: these are the angles which differ from the correct angles by π. They separate the number line into disjoint intervals, each containing one of the correct angles and are colored accordingly.

We start rotating the shape counterclockwise. The marks on the number line moves to the right. After completing one full circle, the shape returns to the canonical position but everything on the number lines has moved to the right by 2π. In particular, the blue regions and the green regions have exchanged their positions.

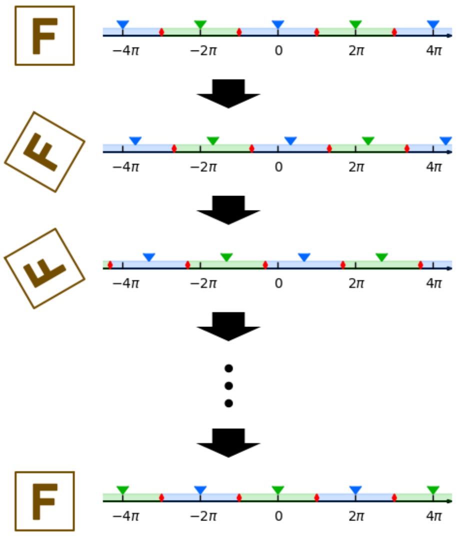


Fig. 1. Demonstration of the idea for the proof of theorem 1

Now consider any continuous function f that maps the input shape to an angle. If at any time the function value coincides with a red diamond, then on that input it produces an error of π. If the initial function value happens to be one of the red diamonds, then we have found such an input. Otherwise, since at the end the shape returns to the same position as the beginning, the function must also give the same value at the end as at the beginning, which means if the initial function value is in the blue region, then at the end it must be in a green region, and vice versa. In any case, since the function value changes continuously, to move between a blue region and a green region it must cross the border between two regions somewhere in the middle, at which point it hits a red diamond.

We can now proceed to capture this intuition precisely in the following proof of theorem 1:

Proof. For any f, let $h(\theta) = \theta - f(R_2(\theta)) - \pi$ and $\widehat{h}(\theta) = h(\theta) - 2\pi \cdot \lceil \frac{h(0)}{2\pi} \rceil$ for all $\theta \in \mathbb{R}$.

$-a < x - a \cdot \lceil \frac{x}{a} \rceil \leq 0$ for all $x \in \mathbb{R}$ and $a > 0$, so $-2\pi < \widehat{h}(0) \leq 0$. Since $R_2(0) = R_2(2\pi) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\widehat{h}(2\pi) - \widehat{h}(0) = h(2\pi) - h(0) = (2\pi - 0) - (f(R_2(2\pi)) - f(R_2(0))) = 2\pi$, so $0 < \widehat{h}(2\pi) \leq 2\pi$.

f and R_2 are both continuous, so $\widehat{h}$ is also continuous. $\widehat{h}(0) \leq 0$ and $\widehat{h}(2\pi) > 0$, by the intermediate value theorem, there exists $\theta_0 \in [0, 2\pi)$ such that $\widehat{h}(\theta_0) = 0$. Then

$$\begin{aligned} & d_2(R_2(\theta_0), R_2(f(R_2(\theta_0)))) \\ =& \min_{k\in\mathbb{Z}} |\theta_0 - f(R_2(\theta_0)) + 2k\pi| \\ =& \min_{k\in\mathbb{Z}} |\theta_0 - f(R_2(\theta_0)) - \pi - 2\pi \cdot \lceil \frac{h(0)}{2\pi} \rceil + (2k+1)\pi| \\ =& \min_{k\in\mathbb{Z}} |\widehat{h}(\theta_0) + (2k+1)\pi| \\ =& \pi \end{aligned} \tag{3}$$

That is, we have found $M = R_2(\theta_0) \in \mathrm{SO}(2)$ such that $d_2(M, R_2(f(M))) = \pi$. □

Nevertheless, the function f in equation 1 is discontinuous at only one input rotation. Under the new approach where we do not pick a unique ground truth function, the network is free to learn to place this discontinuity elsewhere. For example, let $g(R_2(\theta)) = \theta - 2\pi \cdot \lfloor \frac{\theta - 2\pi}{2\pi} \rfloor$, then the discontinuities of f and g are different. Each of f and g gives a correct rotation angle in their respective continuous ranges, and the two ranges together cover the whole $\mathrm{SO}(2)$. In differential geometry terms, each function computes a *chart* and the two charts make up an *atlas*. We will see how this atlas can help us solve rotation estimation correctly using neural networks.

4 THEORETICAL RESULTS

The central idea of our theory is that the guaranteed large error in rotation estimation with neural networks is caused by two factors: that the output space is a non-trivial covering space of the input space, and that there exist non-contractible loops in the possible input range. We show that this is the case for 3D rotation estimation using quaternions as output, just like using angles for 2D rotations. We further show that attempting to make the output space identical to the input space by using an embedding of $\mathrm{SO}(3)$ does not solve this problem completely, since when symmetry is present in the input, the effective topology of the input space is a homogeneous space of $\mathrm{SO}(3)$, thus reintroducing the covering space problem. We instead propose using an ensemble of multiple networks, each responsible for a contractible subset of the input space that collectively cover the whose space.

4.1 3D Rotation Estimation

We have proved that there does not exist any continuous function that maps each 2D rotation to one of its equivalent rotation angles. Now we try to find an analogous argument for 3D rotations.

There are various representations for 3D rotations, with unit quaternions and Euler angles being the most popular. We will first show that there does not exist a continuous function that maps each 3D rotation to one of its quaternion representations, then derive the same result for Euler angles as a corollary.

The quaternion representation of 3D rotations and the definition of the distance between two 3D rotations are reviewed in appendix A. Simply put, the rotation of angle θ around unit vector (x, y, z) is represented by the quaternions $\pm(\cos\frac{\theta}{2} + (x\mathbf{i} + y\mathbf{j} + z\mathbf{k})\sin\frac{\theta}{2})$.

A useful property is that the distance between two 3D rotations represented by quaternions $\mathbf{q}$ and $\mathbf{r}$ is given by

$$d_3(R_\mathrm{q}(\mathbf{q}), R_\mathrm{q}(\mathbf{r})) = 2\arccos|\mathbf{r} \cdot \mathbf{q}| \tag{4}$$

Where $R_\mathrm{q}(\mathbf{q})$ is the rotation represented by $\mathbf{q}$ and $\mathbf{r} \cdot \mathbf{q} = \mathrm{Re}(\mathbf{r}\overline{\mathbf{q}})$ is the dot product of $\mathbf{q}$ and $\mathbf{r}$, seen as 4-vectors.

Since $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation for any $\mathbf{q}$, to convert from a rotation matrix to its quaternion representation requires breaking ties. A common way is to choose between $\mathbf{q}$ and $-\mathbf{q}$ the one with positive scalar part, and positive coefficient of $\mathbf{i}$ if the scalar part is zero, and so on:

$$f(M) = \begin{cases} \frac{a}{2} + \frac{m_{32} - m_{23}}{2a}\mathbf{i} + \frac{m_{13} - m_{31}}{2a}\mathbf{j} + \frac{m_{21} - m_{12}}{2a}\mathbf{k} & (a \neq 0) \\ \frac{b}{2}\mathbf{i} + \frac{m_{21}}{b}\mathbf{j} + \frac{m_{31}}{b}\mathbf{k} & (a = 0, b \neq 0) \\ \frac{c}{2}\mathbf{j} + \frac{m_{32}}{c}\mathbf{k} & (a = b = 0, c \neq 0) \\ \mathbf{k} & (a = b = c = 0) \end{cases} \tag{5}$$

$$a = \sqrt{1 + \mathrm{tr}(M)}, \quad b = \sqrt{2(1 + m_{11})}, \quad c = \sqrt{2(1 + m_{22})}$$

This function is not continuous. Consider $r : \mathbb{R} \to \mathrm{SO}(3), r(\theta) = R_\mathrm{q}(\cos\frac{\theta}{2} + \mathbf{k}\sin\frac{\theta}{2})$, that is, $r(\theta)$ is the rotation around the z-axis by angle θ. Verify that

$$\lim_{\theta \to \pi^-} f(r(\theta)) = \mathbf{k} \neq -\mathbf{k} = \lim_{\theta \to \pi^+} f(r(\theta)) \tag{6}$$

Which also proves that no neural network can compute f correctly for all rotations in $\mathrm{SO}(3)$. But just like in the 2D case, this argument pertains to this particular inverse map only, and on its own does not prove that a continuous inverse of R_q cannot exist.

However, it is true that in general, there is no continuous "inverse" to a non-trivial covering map. We will use covering spaces extensively, so we give a brief introduction here. Extensive discussions on this topic can be found in standard algebraic topology texts, e.g. [Hatcher 2002].

Let X and Y be topological spaces and $\varphi : Y \to X$ be a continuous surjective map. Y is called a covering space of X with covering map φ, if for every $x \in X$, there exists an open neighbourhood U of x such that $\varphi^{-1}[U]$ is a disjoint union of open sets in Y, each being homeomorphic to U under φ. We state the following result:

Theorem 2. *Let Y be a non-trivial covering space of X with covering map φ. That is, φ is not a homeomorphism. Then there does not exist continuous map $f : X \to Y$ such that $\varphi \circ f$ is the identity map.*

Since $\mathrm{Sp}(1)$ is a double cover of $\mathrm{SO}(3)$, simply applying theorem 2 shows that a continuous inverse to R_q does not exist. The 2D case of the nonexistence of a continuous inverse to $R_2 : \mathbb{R} \to \mathrm{SO}(2)$ is also just a special case of theorem 2, since R_2 is a covering map. However, this general result does not establish error bounds as in theorem 1, which is the goal of our next theorem:

Theorem 3. *For any continuous function $f : \mathrm{SO}(3) \to \mathrm{Sp}(1)$, there exists $M \in \mathrm{SO}(3)$ such that $d_3(M, R_\mathrm{q}(f(M))) = \pi$.*

Proof. For any f, let $q : \mathbb{R} \to \mathrm{Sp}(1)$, defined by $q(\theta) = \cos\frac{\theta}{2} + \mathbf{k}\sin\frac{\theta}{2}$. Let $v(\theta) = q(\theta) \cdot f(R_\mathrm{q}(q(\theta)))$. $q(2\pi) = -\mathbf{1} = -q(0)$ and $R_\mathrm{q}(-\mathbf{1}) = R_\mathrm{q}(\mathbf{1})$, so, $v(2\pi) = -v(0)$.

q, R_q and f are all continuous, so v is continuous. By the intermediate value theorem, there exists $\theta_0 \in [0, 2\pi]$ such that $v(\theta_0) = 0$. Then, $d_3(R_\mathrm{q}(q(\theta_0)), R_\mathrm{q}(f(R_\mathrm{q}(q(\theta_0))))) = 2\arccos|v(\theta_0)| = \pi$. That is, we have found $M = R_\mathrm{q}(q(\theta_0)) \in \mathrm{SO}(3)$ such that $d_3(M, R_\mathrm{q}(f(M))) = \pi$. □

Akin to theorem 1, we offer an intuitive demonstration. In figure 2, at the top left is a 3D shape in its canonical position. The ball besides it is a visualization of $\mathrm{Sp}(1)$. Unit quaternions of the form $w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ make up the unit 3-sphere S^3. Due to the difficulty of showing the whole S^3, here we take the 2D section at $x = 0$. The two quaternions representing the identity rotation, $q(0) = (1, 0, 0, 0)$ and $-q(0) = (-1, 0, 0, 0)$, are marked with a blue and a green circle

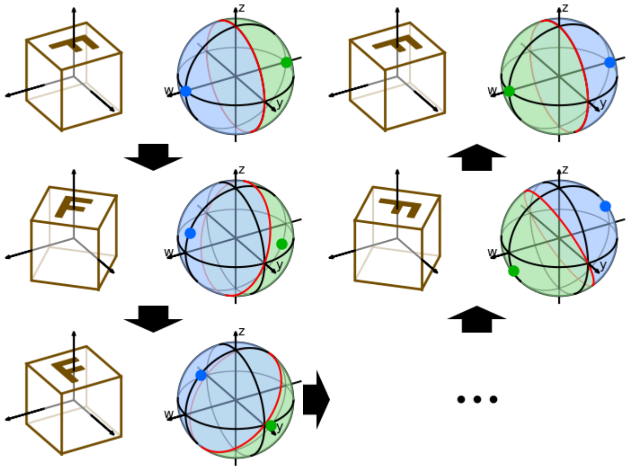


Fig. 2. Demonstration of the idea for the proof to theorem 3

respectively. The red ring are the set of points at quaternion distance $\frac{\pi}{2}$ to $q(0)$ and $-q(0)$, which would correspond to rotation distance π. They actually form a sphere, but here only a 1D section is shown, which is a great circle. They separate S^3 into two hemi-3-spheres, each centered at one of $q(0)$ and $-q(0)$ and are colored accordingly.

We start rotating the shape counterclockwise around the z-axis. The marks on the ball moves accordingly. After completing one full circle, the shape returns to the canonical position but the blue region and the green region have exchanged their places. Similar to the case in theorem 1, in the beginning and in the end, the values of f are located in different colored regions, and somewhere in the middle it must cross the red surface separating the two regions, at which point it produces an error of π.

This can be extended to Euler angles. We discuss it in appendix D.2.

4.2 The Problem of Symmetry

In this section we show that when the input object has symmetry, using embeddings as the output of the network does not eliminate large errors.

Looking back at the demonstrations of theorems 1 and 3, we can see that the central issue is that the output space is a non-trivial covering space of the input space which, among other things, means that for each input there are multiple equivalent outputs that are all correct. The key technique is then to find a loop in the input space such that, if the input is continuously altered along this loop, when the input returns to its initial state some of the equivalent outputs have continuously exchanged their positions.

But the solution to this seems simple: even without the preceding analyses, it is obvious that if we use a rotation representation that is an embedding, none of the issues with quaternions would happen, since with an embedding, by definition there exists a continuous function that maps each input rotation to its correct representation, which is unique. The only drawback might be that the result is a bit less economical in terms of dimensionality of the representation:

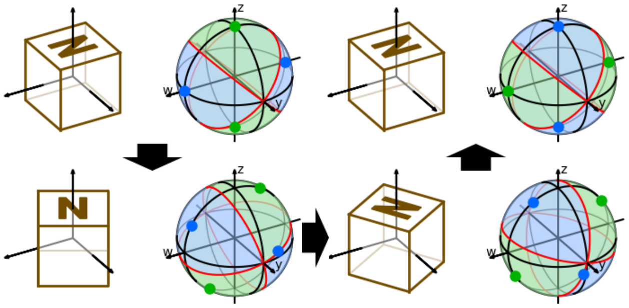


Fig. 3. Estimating the rotation of a shape with symmetry

the minimum n for which an embedding $\mathrm{SO}(2) \to \mathbb{R}^n$ exists is 2 and for $\mathrm{SO}(3) \to \mathbb{R}^n$ it is 5. One concrete example of an embedding $\mathrm{SO}(3) \to \mathbb{R}^5$ is given by [Zhou et al. 2019], which we review in appendix A.

If for whatever rotation estimation problem we are faced with it can be ensured that the input space is indeed SO(3), then it is true that using an embedding as the output rotation representation enables an arbitrarily small maximum error, with the capacity of the neural network being the only limiting factor. However, there is a pitfall: if the input shape has rotation symmetry, then the effective input space might not be SO(3).

While the most common type of symmetry seen in life is reflection symmetry, objects possessing rotation symmetry are nevertheless not uncommon. For example, tables commonly have two-fold or four-fold rotation symmetry, many beverage containers have a five-fold rotation symmetry, cardboard boxes have two-fold rotation symmetry around three different axes. If the input of the rotation estimation problem is such an object, there could be troubles.

For example, consider the shape in figure 3. The letter F on the shape in figure 2 has been replaced with N, which has a rotational symmetry around its center. As a result, when in the canonical position, the shape is invariant under a rotation of π around the z-axis, $Z = R_{\mathrm{q}}(\mathbf{k})$. In general, for any $M \in \mathrm{SO}(3)$, applying M or MZ to the canonical position yields indistinguishable results.

We now try to estimate the rotation of this shape with an embedding of SO(3) as the output representation. SO(3), being non-orientable, is even more difficult to visualize than Sp(1). So, we have kept the sphere section visualization of Sp(1) in figure 3 but now each single point in SO(3) would be represented by a pair of antipodal points on the sphere. To signify this, opposite points are marked with the same color.

Start from the canonical position. Since the rotation Z is indistinguishable from the identity rotation I, both should be considered correct. I is marked with a pair of blue circles and Z is marked with a pair of green circles. Rotate the shape counterclockwise around the z-axis. After a rotation of π, the position of the shape is indistinguishable from the initial position. But the two correct output rotations have exchanged their positions. We can then conclude, as before, that at some point the output of the network must be on the red lines. Note however that this time the red points do not all give the same error. But it is still possible to conclude that at some point the error of the network is no smaller than the smallest error among all red points.

By introducing symmetry in the shape, we cause some input rotations to become indistinguishable, so, the effective input space is not $\mathrm{SO}(3)$ anymore. In general, given a shape, let $G \subset \mathrm{SO}(3)$ be the set of rotations whose actions leave the shape invariant, then it is easy to see that G is a subgroup of $\mathrm{SO}(3)$. In this paper we consider finite G only. If $R_1, R_2 \in G$ and $M \in \mathrm{SO}(3)$, then the actions of MR_1 and MR_2 are indistinguishable, so in the input space they should be identified. That is, two elements of $\mathrm{SO}(3)$ should be identified whenever they are in the same left coset of G. The set of left cosets of G in $\mathrm{SO}(3)$ is a homogeneous space of $\mathrm{SO}(3)$, denoted $\mathrm{SO}(3)/G$.

$\mathrm{SO}(3)$ is a covering space of $\mathrm{SO}(3)/G$, and non-trivial if G is non-trivial. The covering map φ_G maps $M \in \mathrm{SO}(3)$ to the left coset MG. Applying theorem 2, we have the following conclusion:

Corollary 4. *Let G be a finite, non-trivial subgroup* $\mathrm{SO}(3)$. *Then there does not exist continuous map $f : \mathrm{SO}(3)/G \to \mathrm{SO}(3)$ such that $\varphi_G \circ f$ is the identity map.*

This means that, in a rotation estimation problem, if the input object has rotation symmetry, then even if the output representation is an embedding of $\mathrm{SO}(3)$, it is guaranteed that any neural network will produce errors on some input. Note that this does not apply to embeddings only: any representation for which a continuous map to $\mathrm{SO}(3)$ exists would be problematic, since this map can be composed with the neural network to produce a continuous map from the input to $\mathrm{SO}(3)$.

In corollary 4 we did not establish error lower bounds like in theorems 1 and 3. As mentioned in the demonstration, it is relatively easy to prove that the maximum error is no smaller than half the distance between the two closest elements of G, which is $\frac{\pi}{2}$ for the two-fold rotation symmetry example. However, this lower bound is not tight. The proof of a tight lower bound requires advanced techniques, and is presented in appendix D.

4.3 The Self-selecting Ensemble

Since embeddings of $\mathrm{SO}(3)$ fail to provide a universal solution to rotation estimation problems due to input symmetry, we need to seek alternative solutions. We show that regardless of symmetry or output representation, we can always find a set of four functions so that for each input at least one function gives the correct output, which enables us to use an ensemble of networks to estimate 3D rotations correctly.

Neural networks are typically trained with gradient-based methods, and the gradients are usually computed by back-propagation, which requires the network to be differentiable and thus continuous. Nevertheless, we often employ discontinuous operations when making predictions using the trained network, e.g. quantizing a probability distribution into a class label in classification networks. Unfortunately this does not work for a regression problem with a continuous output space. However, by employing an ensemble trick, it is possible to transform this discontinuity of regression into a discontinuity of classification.

Recall that in the discussion of 2D rotation estimation, we have found two versions of atan2 that have different discontinuities. They can be slightly modify so that they are continuous, while keeping the property that their correct range cover the whole $\mathrm{SO}(2)$. The learned functions in Figure 5b give an example. Thus we make the following observation:

Theorem 5. *There exist continuous functions $f_1, f_2 : \mathrm{SO}(2) \to \mathbb{R}$ such that for any $M \in \mathrm{SO}(2)$, $R_2(f_i(M)) = M$ for some $i \in \{1, 2\}$.*

Or, we can say that there exist an atlas of $\mathrm{SO}(2)$ with two charts. But note that such a set of functions indeed have slightly more structure than an atlas: each chart only needs to be defined on a subset of $\mathrm{SO}(2)$, while each function must be defined and continuous on the whole $\mathrm{SO}(2)$ although each only have to be correct on a subset. Nevertheless it is not hard to construct such a set of functions given an atlas.

Now that f_1 and f_2 are continuous, they can be learned by neural networks. Importantly, on top of these, we can add a classifier that predicts which function would give the correct output for each input, which can also be learned by a neural network. During training time, these functions and the classifier can be trained jointly: the error of the whole ensemble is the sum of the error of each individual functions, weighted by the probability assigned by the classifier. Now the discontinuity only happens at inference time, when we select the output of the function with the highest assigned probability. We call this method the *self-selecting ensemble*.

We try to apply this idea to 3D rotation estimation using quaternion representation. We have made the observation that a loop in the input space that causes equivalent representations in the output space to exchange places causes the network to fail. Such loops are always *non-contractible*. So, our goal is that we need to partition $\mathrm{SO}(3)$ into some number of subsets that do not contain non-contractible loops. We can then train an ensemble with as many networks, each only required to be correct on one of the parts. The simplest way is to let each part themselves be contractible. Indeed, in the constructions in our proofs, each part is a (topological) ball or a disjoint union of balls.

For a topological space X, the smallest number k such that there exists an open cover $\{U_i | 1 \leq i \leq k\}$ of X with each U_i contractible is called the *Lusternik–Schnirelmann category* (LS-category) of X, denoted $\mathrm{cat}(X)$. It has been shown in [Takens 1968] that for a smooth compact manifold X, $\mathrm{cat}(X) \leq \dim(X) + 1$. $\mathrm{SO}(3)$ is a smooth compact manifold and $\dim(\mathrm{SO}(3)) = 3$, so $\mathrm{cat}(X) \leq 4$, that is, we can find four contractible subsets of $\mathrm{SO}(3)$ that cover it.

To find such subsets, the easiest way is to find such subsets for its double cover S^3 instead while requiring the membership of p and $-p$ to always be the same for all p, so that by identifying antipodal points we then get suitable subsets for $\mathrm{SO}(3)$. Observe the construction for lower dimensions: as shown in figure 4, S^0 can be covered by one pair of points; S^1 can be covered by two pairs of arcs; S^2 can be covered by three pairs of caps. Following the same pattern, S^3 can be covered by four pairs of hyper-caps. In (w, x, y, z) coordinates, the four subsets consists of points with $|w| > \frac{1}{2} - \epsilon$, $|x| > \frac{1}{2} - \epsilon$, $|y| > \frac{1}{2} - \epsilon$ and $|z| > \frac{1}{2} - \epsilon$, respectively, for some small $\epsilon > 0$.

We can then construct a set of four continuous functions such that for every input rotation at least one of them gives a correct quaternion representation.

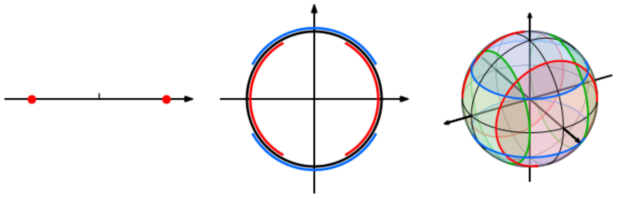


Fig. 4. Covering S^n with $(n+1)$ center-symmetric pairs of contractible subsets, $n = 0, 1, 2$

Theorem 6. *There exist continuous functions $f_1, f_2, f_3, f_4 : \mathrm{SO}(3) \to \mathrm{Sp}(1)$ such that for any $M \in \mathrm{SO}(3)$, $R_q(f_i(M)) = M$ for some $i \in \{1, 2, 3, 4\}$.*

We can also show that a set of 4 functions is optimal. Indeed, a set of 3 functions can do no better than a single function in terms of maximum error, which is a stronger result than what can be derived from the general theory of LS-category:

Theorem 7. *For any three continuous functions $f_1, f_2, f_3 : \mathrm{SO}(3) \to \mathrm{Sp}(1)$, there exists $M \in \mathrm{SO}(3)$ such that $d_3(M, R_q(f_i(M))) = \pi$ for all $i \in \{1, 2, 3\}$.*

Proofs of theorems 6 and 7 can be found in appendix D, along with similar results for Euler angles, for which we additionally need to solve the problem of gimbal lock.

Thus far we have shown that 3D rotation estimation of non-symmetric object with quaternion or Euler angle output can be solved with an ensemble of four networks. But this is no better than using an embedding of SO(3), which does not require an ensemble. The true strength of the ensemble method is that it works equally well on objects with rotation symmetry, where an embedding fails:

Theorem 8. *Let G be a finite subgroup of $\mathrm{SO}(3)$, then there exist continuous functions $f_1, f_2, f_3, f_4 : \mathrm{SO}(3)/G \to \mathrm{SO}(3)$ such that for every $M \in \mathrm{SO}(3)/G$, $\varphi_G(f_i(M))) = M$ for some $i \in \{1, 2, 3, 4\}$.*

This would also follow from the property of the LS-category, since for a finite subgroup G of SO(3), SO(3)/G is again a smooth compact manifold of dimension 3. Note that although the theorem is stated for using an embedding, it is equally valid for quaternion representations: as we have discussed extensively, the key to the construction of a successful ensemble is to find a collection of contractible subsets that cover the input space, and the choice of output space matters little.

Unlike the non-symmetric case, the derivation for a tight lower bound of maximum rotation estimation error on SO(3)/G using only one function is much more complicated, as is the construction of four open subsets that cover SO(3)/G. These will be discussed in appendix D.

5 EXPERIMENTS

In this section we verify our theoretical results with four 3D rotation estimation experiments. The first experiment is a simple 2D toy problem: the input is a simple rotated square, with either a letter F or N resulting in different symmetries. The network is trained to estimate its rotation, relative to a fixed position regarded as canonical. The second experiment is a 3D toy problem, with input being a simple cube drawn with orthographic projection and flat coloring. In the Third experiment, the inputs are changed to photorealistic renderings of 3D models of various chairs and tables. In the fourth experiments, the inputs are changed to point clouds of those chairs and tables.

5.1 Network and Training

In all of our experiments, whenever an ensemble of networks is desired, all networks in the ensemble as well as the classifier are implemented as a single network that branches only at the last layer, so that comparing to a non-ensemble with just a single network, an ensemble requires only a little bit of additional computation.

For the first three experiments, we use convolutional networks, with input image size of 256 × 256. For experiment 4, we use a network that is similar to PointNet [Qi et al. 2017] on a high level but much simpler. The specific architecture is unimportant as far as qualitative results are concerned, and our networks only serve as examples. As a reference, the sequence of layers is given in Table 7 in appendix B.

Since we will be considering symmetric objects, rotation distance in such cases should be properly defined. Define the rotation distance between $M_1, M_2 \in \mathrm{SO}(3)$ under symmetry G as:

$$d_G(M_1, M_2) = \min_{S \in G} d_3(M_1, M_2 S) \tag{7}$$

That is, the distance from M_1 to M_2 under symmetry G is the shortest distance from M_1 to any rotation that is indistinguishable from M_2 under G.

The loss function is computed as follows: Whether the input is a rendered image or a 3D point cloud, it can always be identified by a rotation relative to the canonical position. So we just use $M \in \mathrm{SO}(3)$ to refer to the input. On input M, let $f_1(M), \ldots, f_n(M)$ be the output of an ensemble of size n. Let $w_1(M), \ldots, w_n(M)$ be the weights assigned to each branch in the ensemble by the classifier. These "weights" are not exactly probabilities as usually is the case for classification problems, and they are not required to be positive and sum to 1. Then, the loss $\mathcal{L}$ of the ensemble on input M of an object with symmetry G is computed as:

$$\overline{w}(M) = \frac{1}{n} \sum_{i=1}^{n} w_i(M)$$

$$\mathcal{L}(M) = \sum_{i=1}^{n} \max\{w_i(M) - \overline{w}(M) + \frac{1}{n}, 0\} \cdot d_G(f_i(M), M) \tag{8}$$

In particular, this training procedure is weakly supervised in that the exact division of the input space into contractible subsets and the exact form of the mapping from the input to the output on each subset is not used as training labels. Instead, only the rotation distance function is required, which is no more than what is required to train a single non-ensemble rotation estimation network.

Note that objects with different symmetries can be mixed in the same dataset, and the form of the loss function for each individual object can thus be different.

The most important part of our theoretical results is the establishment of lower bounds of maximum error for certain representations and ensemble sizes, so we are interested in trying to lower the maximum error of the network. However, uniform random sampling of the possible input space minimizes the expected error rather than the maximum. So we have adjusted the training procedure: rather than sampling a new batch for each iteration, after each iteration we sort the samples in the current training batch by their error, keep the larger half of them, then draw half a batch of new samples. These are combined to form the training batch of the next iteration.

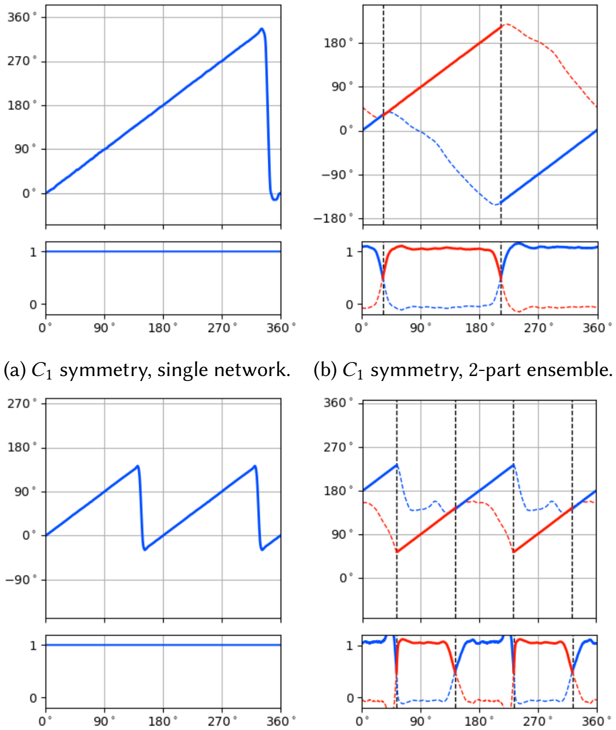
All networks are trained with the Adam optimization algorithm, with batch size 16 and learning rate 10^{-4}. For experiment 1, the training time is $40,000$ iterations. For experiments 2 and 3 it is $200,000$ iterations and for experiment 4 it is $120,000$ iterations. To ensure that the whole $\mathrm{SO}(2)$ or $\mathrm{SO}(3)$ is covered rather than just training on a small finite subset, instead of preparing training samples beforehand, all training data are generated on the fly during training by randomly sampling a rotation, transforming the object with that rotation, then either rendering it for image input or just taking the transformed point coordinates for point cloud input.

5.2 2D Toy Problem

This experiment is included to show the actual functions learned by the network. The inputs are simple squares having letter F (with C_1 symmetry, i.e. no symmetry) or N (with C_2 symmetry) on them. For each of these two letters, we train two rotation angle estimation networks, one using a single network, the other using a two-part ensemble. The functions learned by these networks are shown in Figure 5. In each figure, the upper penal is the output of each part in the ensemble, and the lower penal is the weight assigned by the classifier. Each color represents one part in the ensemble. For each input, the output with the highest weight is shown with thick lines. We can clearly see that the single networks give correct outputs in most of the input range but produce high errors near the place where it have to jump between two equivalent angles. In contrast, in the ensembles, the thick lines always give the correct output and is able to produce a discontinuity despite all functions in the ensemble being continuous.

5.3 3D Toy Problem

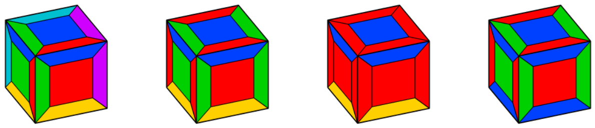
For the second experiment, the inputs are simple orthographic drawings of a colored cube. There is no difference between the training and testing data. By adjusting the color of each face, it is possible to produce several different symmetries. For example, if all faces are colored differently, the cube will not have any rotation symmetry (C_1); if the front and back faces are of the same color and the left and right faces are of the same color, the cube will have two-fold rotation symmetry (C_2), and so on. Examples of cubes with different symmetries are shown in Figure 6. One complication is that if the view is facing one face directly, then only that face is visible and without knowing the color of other faces it is not possible to uniquely determine the rotation of the cube. To solve this problem, on each face, beside each edge, we paint a color strip with the color of the adjacent face.



(a) C_1 symmetry, single network. (b) C_1 symmetry, 2-part ensemble.

(c) C_2 symmetry, single network. (d) C_2 symmetry, 2-part ensemble.

Fig. 5. Functions learned by the 2D rotation estimation networks.



(a) C_1 symmetry (b) C_2 symmetry (c) C_4 symmetry (d) D_2 symmetry

Fig. 6. Drawings of cubes with different symmetries.

The experiment is repeated under 5 different configurations. In four of them, the dataset consists solely of images of one type of symmetry, with the symmetry being C_1, C_2, C_4 and D_2 for each. In the fifth configuration images of C_1 symmetry and C_2 symmetry are mixed in equal proportions. For each kind of symmetry, the color assignment is fixed. For the experiments where cubes of different symmetries are mixed, each symmetry is assigned a different background color to help the network more easily tell them apart.

For the C_1 and C_2 symmetries, we compare the result of using the quaternion, the Euler angle, the 6D embedding in [Zhou et al. 2019], the SVD method in [Levinson et al. 2020] and the QCQP method in [Peretroukhin et al. 2020]. We then observe that the qualitative distinction is between embedding-type (6D, SVD, QCQP) and non-embedding-type (Euler angle, quaternion) of representations, and the difference within each group is quantitative. For the other symmetries in this experiment, as well as for the subsequent two experiments, we only compare one kind of representation from each type as a representative: quaternion for the non-embedding-type and the 6D embedding for the embedding-type.

We tested each network with $500,000$ randomly sampled rotations. The results are shown in Tables 1 through 3. In addition, loss distribution curves have been included in Figure 9 in appendix C. Each table lists the mean and maximum error of each representation. "Mixed Euler" is the four-part ensemble with two x-y-z Euler angles and two x-z-y Euler angles, while the other Euler angle ensembles are homogeneous. Rows marked with a red up-pointing triangle means that theoretically this representation should produce maximal maximum error (120° for D_2 symmetry and 180° for the rest). Rows marked with a blue down-pointing triangle means that theoretically this representation should produce small maximum error.

The results for C_1 symmetry is shown in Table 1. We can see that embedding-type representations solves the rotation estimation problem in the non-symmetric case satisfactorily, while for the non-embedding-type representations a four-part ensemble is need, and with fewer than 4 networks, near-180° errors are guaranteed to occur. We can also see that the mixed-type Euler angle ensemble gives lower maximum error than the homogeneous ensemble, albeit just barely.

Note that while inputs producing 180° errors are guaranteed to exist for some representations, in general they make up only a subset of measure zero in the whole input space, which means it is virtually impossible for a random input to produce 180° error. However for any $\epsilon > 0$, errors at least $(180 - \epsilon)^\circ$ occur with nonzero probability, and by increasing the number of test samples we can produce arbitrarily close to 180° maximum error.

The results for C_2 symmetry is shown in Table 2. The mixed Euler ensemble is not used since the construction we gave is specifically for the non-symmetrical setting. In this case, a single 6D embedding network gave a maximum error of nearly 180°, despite being an embedding. The SVD and QCQP representations, however, give maximum errors of nearly 90°. The mapping from SVD or QCQP representation to $\mathrm{SO}(3)$ is not continuous everywhere, and thus the lower bound of maximum error of 180° derived in appendix D do not exactly apply. But we have nevertheless observed that a single network using either representation fails to prevent the occurrence of large errors in the symmetric case. Incidentally, the 90° error bound is the same that can be observed from the demonstration in Figure 3. Further theoretical investigation is need to establish the lower bound for these representations.

On the other hand, the four-part ensemble of each of these representations all give small error.

The smaller-sized ensembles give varying errors. Indeed, unlike in the non-symmetric case, the lower bound of maximum error for smaller-sized ensembles is smaller than 180°.

For the C_4, D_2 and mixed C_1/C_2 symmetries, we tested four-part quaternion ensemble, single 6D embedding and four-part 6D embedding ensemble only. The results are given in Table 3. In each configuration, the ensembles give small error but the single 6D embedding network gives maximum error close to the theoretical bound. In particular, the error of nearly 120° for D_2 symmetry serves to verify our calculation of lower bound in appendix D: we predicted that with D_n symmetry the lower bound for maximum error should be $\arccos(-\sin^2 \frac{\pi}{2n})$. Plugging in $n = 2$ gives $\arccos(-\sin^2 \frac{\pi}{4}) = \arccos(-\frac{1}{2}) = 120^\circ$.

Table 1. Mean and maximum error of various representations on the toy problem with C_1 symmetry.

Representation	Mean err. (°)	Max err. (°)	
Euler ×1	126.5082	180.0000	△
Euler ×2	3.2577	179.5049	△
Euler ×3	1.4588	178.0552	△
Euler ×4	1.4751	8.1049	
Mixed Euler	1.5434	7.6233	▽
Quat ×1	4.8754	179.8199	△
Quat ×2	1.7230	179.4418	△
Quat ×3	1.0384	178.8919	△
Quat ×4	1.0038	4.6940	▽
6D	1.0242	3.6379	▽
SVD	0.9225	3.6405	▽
QCQP	1.0017	3.6695	▽

Broadly, we can draw the conclusion that for quaternion and Euler angle representations, both the symmetric and non-symmetric 3D rotation estimation problem can be solved with a four-part ensemble but not a smaller ensemble, while with embedding-type representations, non-symmetric case can be solved with a single network but for the symmetric case an ensemble is still required.

Stated differently, the self-selecting ensemble provides a unified method for solving 3D rotation estimation problems, regardless of rotation representation or object symmetry.

In particular, the results for the mixed C_1/C_2 symmetry shows that although $\mathrm{SO}(3)$ and $\mathrm{SO}(3)/C_2$ are different spaces and thus the collection of four contractible subsets that cover them are also different, a single four-part ensemble can handle them simultaneously.

5.4 Object Rotation Estimation From Images

In this experiment we solve the same problem as in experiment 2, estimating the rotation of an object from an image, but under a more practical setting: we use rendered 3D models of tables and chairs. We gathered 102 chairs and 124 tables from 3D Warehouse[1]. We chose these models specifically for their symmetry: we inspected each model to ensure that none of the chair models has nontrivial symmetry and the table models all have two-fold rotation symmetry (C_2).

Real-life objects are rarely perfectly symmetric, and this is reflected in our setup: the symmetry needs not be exact. The texture can be slightly asymmetric, and tiny parts may be present on one side but not on the other. The difference only has to be small enough so that it is hard for the network to distinguish between two views that differ by a rotation of 180° around the z-axis.

Examples of these models are shown in Figure 7. The models are so aligned that in their canonical position, a person sitting on a chair would face the positive x direction and a table would have its longer side parallel to the x-axis.

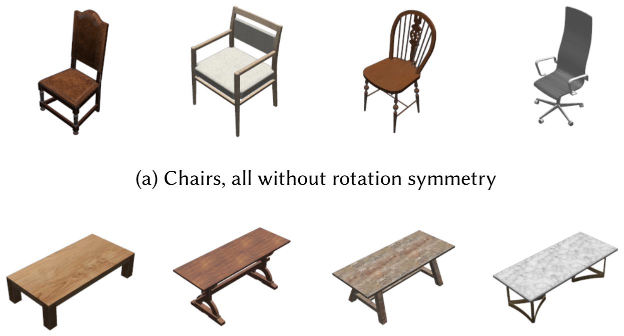
[1] https://3dwarehouse.sketchup.com/

Table 2. Mean and maximum error of various representations on the toy problem with C_2 symmetry.

Representation	Mean err. (°)	Max err. (°)	
Euler ×1	102.3217	179.8983	△
Euler ×2	2.6732	178.5300	
Euler ×3	1.7388	109.7955	
Euler ×4	1.2381	18.8490	
Quat ×1	5.9578	178.9537	△
Quat ×2	1.6456	111.6298	
Quat ×3	1.0721	80.4804	
Quat ×4	1.0855	5.2841	▽
6D ×1	13.4844	179.8992	△
6D ×2	1.4515	179.1538	
6D ×3	1.0352	106.9555	
6D ×4	1.0292	9.3179	▽
SVD ×1	1.7343	89.9928	
SVD ×4	0.8699	4.3780	▽
QCQP ×1	2.4835	89.9885	
QCQP ×4	0.9074	3.6334	▽

Table 3. Mean and maximum error of various representations on the toy problem with C_1/C_2, C_4 or D_2 symmetries.

Symmetry	Representation	Mean err. (°)	Max err. (°)	
C_1/C_2	Quat ×4	1.2139	6.4589	▽
	6D ×1	12.5738	179.9038	△
	6D ×4	1.0952	6.3162	▽
C_4	Quat ×4	1.1063	13.8207	▽
	6D ×1	6.6305	178.5681	△
	6D ×4	0.9179	13.4447	▽
D_2	Quat ×4	0.9058	5.6622	▽
	6D ×1	20.7738	118.8476	△
	6D ×4	0.8450	8.3506	▽



(a) Chairs, all without rotation symmetry

(b) Tables, all with C_2 rotation symmetry

Fig. 7. Examples of models used in experiment 3.

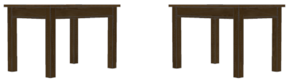


Fig. 8. Two non-equivalent rotations that are easily confused.

Table 4. Mean and maximum error on rotation estimation from images.

Object	Representation	Mean err. (°)	Max err. (°)	
Chairs	Quat ×4	5.3492	64.8730	▽
	6D	6.8317	65.3484	▽
Tables	Quat ×4	2.4796	15.5027	▽
	6D ×1	7.8170	178.7823	△
	6D ×4	2.2435	15.3726	▽

We keep 10 each of chairs and tables for testing and use the others for training. The rendering process needs not be differentiable, so off-the-shelf renderers can be used.

We found that there are certain rotations that are not equivalent but are nevertheless difficult to tell apart from a certain view, for example, when the camera is close to the surface plane of the table as shown in Figure 8. We consider this type of error as generalization error, and not the kind of topological error we want to investigate, so we must eliminate it to isolate topological errors for our analysis. To make sure that the network receives enough information to distinguish non-equivalent rotations, for each rotation of the model we render the front view, left view and top view and concatenate the three images along the channel dimension to serve as the input to the network.

We trained our networks on chairs and tables separately. We tested each network with 100, 000 images. The results are shown in Table 4 and Figure 10 in appendix C.

We can see that the maximum error on the chairs dataset using four-part quaternion ensemble or 6D embedding are predicted to be “small” but they are not as small as in the toy problem. However, the fundamental difference is the between existence and non-existence of 180° errors, and the different magnitude of error below that is only a quantitative difference. The maximum errors are far away from 180°, so in this sense they are indeed small. We think that this error is mostly due to the inherent difficulty of this problem, rather than the topological properties of the representation.

5.5 Object Rotation Estimation From Point Clouds

In this experiment, we solve the same problem of rotation estimation of real-world objects. But instead of rendered images, we estimate the rotation from point clouds. This setting would be of more relevance to computer graphics applications, since point cloud registration is an important problem.

Table 5. Mean and maximum error on rotation estimation from point clouds.

Object	Representation	Mean err. (°)	Max err. (°)	
Chairs	Quat ×4	4.4065	15.0382	▽
	6D	3.7559	15.0945	▽
Tables	Quat ×4	3.2760	34.0510	▽
	6D ×1	19.5237	178.4878	△
	6D ×4	2.2307	19.8830	▽
Mixed	Quat ×4	5.1593	39.0340	▽
	6D ×1	8.5855	178.2173	△
	6D ×4	5.3365	28.6523	▽

Table 6. Mean and maximum error of each type of object when training on the combined dataset.

Object	Representation	Mean err. (°)	Max err. (°)
Chairs	Quat ×4	5.5323	21.1359
	6D ×4	5.6780	24.6610
Tables	Quat ×4	4.7863	39.0340
	6D ×4	4.9950	28.6523

The point clouds are sampled from those 3D models in experiment 3. Specifically, we sample 2, 000 points from each model using the Poisson Disk sampling [Corsini et al. 2012] filter in MeshLab [Cignoni et al. 2008]. Again, to reflect the situation that actual scanned point clouds are rarely perfectly symmetric, we did not enforce this perfect symmetry: there is no guarantee that for one point in the point cloud of a C_2-symmetric table, its reflection through the z-axis is also present in the point cloud.

These point clouds are generated and fixed before training. During training, a rotation is randomly selected and the input is generated by rotating the point cloud.

It should be noted that if points in a point cloud are stored in a fixed order, then if a rotation is applied to the point could, this rotation can be easily recovered by comparing the position of corresponding points, regardless of any possible symmetry of the geometric configuration of the points. However, since in general the order of points in a point cloud are assumed to have no significance, neural networks that process point clouds are typically designed to be order-agnostic, as is the case for PointNet [Qi et al. 2017] on which our network is based. In such a case, a symmetric point cloud in two equivalent rotations appear indistinguishable to the network.

We trained our networks on chairs and tables separately, as well as on the combined set of chairs and tables. We tested each network with 100, 000 point clouds. The results are shown in Table 5 and Figure 11 in appendix C.

In general, the results of experiment 3 and 4 reconfirms the findings in experiment 2, but in a more practical setting.

We are also interested in the error of each type of object when training on the combined dataset, given in Table 6, compared to when training on each type of object separately. Despite the training dataset being larger, the error of each type of object when training on the combined dataset is larger than when training on that type of object alone. This shows that it is generally more difficult to handle different types of symmetry at once using the same network. In particular, as can be seen in Figure 5, with a higher order of symmetry the weight function must have higher frequency, which could negatively affect the performance on lower order symmetry, which favors a low frequency weight function.

6 DISCUSSIONS

One interesting finding is the contrast between our results on rotation estimation from point clouds and those of a similar experiment in [Zhou et al. 2019]. In their experiments, rotation estimation were tested on a collection of (non-symmetric) airplane models. Despite the fact that the 6D and 5D embedding that they have proposed should in theory give small error, all representations compared in their experiments gave maximum errors close to 180°, thus failing to show an important advantage of embeddings over non-embeddings. Indeed, the authors of [Levinson et al. 2020] repeated the same experiment and their SVD method also gave a maximum error of 180°. In [Peretroukhin et al. 2020] the same experiment is repeated again, adding their QCQP method, but the authors did not discuss maximum error. The difference might be that we carefully checked the data to rule out near-symmetry in our chairs, which some of their airplanes might possess, or that our training strategy emphasized lowering maximum error. Whichever is the case, here in addition to showing the power of ensembles, our experiments confirmed the qualitative difference between embeddings and non-embeddings in the non-symmetric case.

The theoretical thinking behind the idea of the self-selecting ensemble is to divide a topologically non-trivial input space into topologically simpler (contractible) parts and use one network to handle each part, which is not specific to rotation estimation problems. For example, we might want to produce a 3D hair model from a portrait photo. Depending on the specific method, this might involve inferring the 3D direction of a short hair segment from local image information. In such a case, (x, y, z) and $(-x, -y, -z)$ are two indistinguishable directions, which means effectively the input space is the real projective plane $\mathbf{RP}^2$ while the output space is the 2-sphere S^2. S^2 is a covering space of $\mathbf{RP}^2$, so if neural networks are employed for this task, topological errors could occur. It is worthwhile to investigate the application of the ensemble method to more general problems with nontrivial topology.

7 CONCLUSION

In this paper, from a topological point of view, we gave a rigorous proof that continuous functions that map rotations to the correct quaternion or Euler angle representations do not exist, and established lower bounds for their maximum error, thus presenting a case against their use in rotation estimation neural networks. Furthermore, similar negative results are established for rotation estimation of objects with rotation symmetry using embeddings of SO(3), thus showing that proposing new rotation representations alone is unlikely to solve rotation estimation problems satisfactorily under different symmetries. We proposed the self-selecting ensemble, a general method independent of the choice of rotation representation, and with elaborate theoretical analysis, we proved that a sufficiently large ensemble eliminates the guaranteed existence of large errors. We showed with experiments that in practice our method is effective under many different symmetries and with many different rotation representations.

ACKNOWLEDGMENTS

Research was sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-20-2-0053, and sponsored by the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005, the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA and in part by the ONR YIP grant N00014-17-S-FO14. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation.

The author thanks Yi Zhou and Connelly Barnes for early discussions regarding the continuity of rotation representations. He also wishes to thank the reviewers and committee members for their comments and guidance. This is the author's first peer-reviewed publication, and he learned a lot from working on this project and from the review process.

REFERENCES

Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. 2008. MeshLab: an Open-Source Mesh Processing Tool. In *Eurographics Italian Chapter Conference*, Vittorio Scarano, Rosario De Chiara, and Ugo Erra (Eds.). The Eurographics Association. https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136

Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. 2012. Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes. *IEEE Transaction on Visualization and Computer Graphics* 18, 6 (2012), 914–924. http://vcg.isti.cnr.it/Publications/2012/CCS12 http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.34.

Erik W Grafarend and Wolfgang Kühnel. 2011. A minimal atlas for the rotation group SO (3). *GEM-International Journal on Geomathematics* 2, 1 (2011), 113–122.

Allen Hatcher. 2002. *Algebraic Topology*. Cambridge University Press. Electronic version available at https://pi.math.cornell.edu/~hatcher/AT/ATpage.html.

Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. 2020. An analysis of svd for deep rotation estimation. *arXiv preprint arXiv:2006.14616* (2020).

Siddharth Mahendran, Haider Ali, and René Vidal. 2017. 3d pose regression using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2174–2182.

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

G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 2020. 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7193–7203.

Valentin Peretroukhin, Matthew Giamou, David M Rosen, W Nicholas Greene, Nicholas Roy, and Jonathan Kelly. 2020. A smooth representation of belief over so (3) for deep rotation learning with uncertainty. *arXiv preprint arXiv:2006.01031* (2020).

Omid Poursaeed, Guandao Yang, Aditya Prakash, Qiuren Fang, Hanqing Jiang, Bharath Hariharan, and Serge Belongie. 2018. Deep fundamental matrix estimation without correspondences. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. 2009. Learning 3-d object orientation from images. In *2009 IEEE International Conference on Robotics and Automation*. IEEE, 794–800.

Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. 2015. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*. 2686–2694.

Floris Takens. 1968. The minimal number of critical points of a function on a compact manifold and the Lusternik-Schnirelman category. *Inventiones mathematicae* 6, 3 (1968), 197–244.

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

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. 2018. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*.

Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. 2016. Deep kinematic pose regression. In *European Conference on Computer Vision*. Springer, 186–201.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5745–5753.

A MATHEMATICS REVIEW

A.1 Quaternion representation of 3D rotations

The unit quaternion representation for 3D rotations is a classic example of exceptional isomorphisms. It arises from the fact that the spin group Spin(3), the (two-fold) universal cover of the special orthogonal group SO(3), is, as a Lie group, isomorphic to the multiplicative group of unit quaternions, or, the compact symplectic group Sp(1). As a result, the quaternion representation has several nice properties: composition of rotations correspond to multiplication of quaternions, and the quaternion representation, in a sense, preserves the metric in SO(3).

The spin group Spin(n) is constructed as a subgroup of the invertible elements of the Clifford algebra $\mathrm{Cl}^{+}_{0,n}(\mathbb{R})$, and in the case of $n = 3$, we have $\mathrm{Cl}^{+}_{0,3}(\mathbb{R}) \cong \mathbb{H}$, which is what gives rise to $\mathrm{Spin}(3) \cong \mathrm{Sp}(1)$. $\mathbb{R}^3$ can be embedded in $\mathbb{H}$ as the subset of pure quaternions, by the map

$$(x, y, z) \mapsto x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \tag{9}$$

or, if we regard $p = [x\ y\ z]^T$ as a column vector and use the dot product,

$$p \mapsto [\mathbf{i}\ \mathbf{j}\ \mathbf{k}]p \tag{10}$$

By the definition of Clifford algebras, a unit quaternion $\mathbf{q} \in \mathrm{Sp}(1)$ should act on embedded 3-vectors $\mathbf{p} = [\mathbf{i}\ \mathbf{j}\ \mathbf{k}]p$ by the map

$$\mathbf{p} \mapsto \mathbf{q}\mathbf{p}\overline{\mathbf{q}} \tag{11}$$

Where $\overline{\mathbf{q}}$ is the conjugate of $\mathbf{q}$. So, if $M = R_{\mathrm{q}}(\mathbf{q}) \in \mathrm{SO}(3)$ is the rotation matrix represented by the quaternion $\mathbf{q}$, we should have

$$[\mathbf{i}\ \mathbf{j}\ \mathbf{k}](Mp) = \mathbf{q}([\mathbf{i}\ \mathbf{j}\ \mathbf{k}]p)\overline{\mathbf{q}} \tag{12}$$

Let $\mathbf{q} = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$. Expanding both sides and comparing term by term, we find that

$$M = \begin{bmatrix} 1 - 2c^2 - 2d^2 & 2(bc - da) & 2(bd + ca) \\ 2(bc + da) & 1 - 2b^2 - 2d^2 & 2(cd - ba) \\ 2(bd - ca) & 2(cd + ba) & 1 - 2b^2 - 2c^2 \end{bmatrix} \tag{13}$$

A.2 3D Rotation Distance

The “difference” between two 3D rotations is again a 3D rotation. That is, for all $M_1, M_2 \in \mathrm{SO}(3)$, $M_2 M_1^{-1} \in \mathrm{SO}(3)$. An important property of orthogonal matrices is that all their eigenvalues have length 1. Complex eigenvalues of real matrices always appear in conjugate pairs. For an orthogonal matrix, this means their eigenvalues must be 1, −1 or pairs of complex numbers of the form $\cos\theta \pm i \sin\theta$.

For 3D rotation matrices, their three eigenvalues are exactly 1 and $\cos\theta \pm i\sin\theta$. The eigenspace of eigenvalue 1 is the rotation axis and θ is the rotation angle. The trace of a matrix equals the sum of its eigenvalues. So, for a 3D rotation matrix M with rotation angle θ, $\mathrm{tr}(M) = 1 + (\cos\theta + i\sin\theta) + (\cos\theta - i\sin\theta) = 1 + 2\cos\theta$, so $\theta = \arccos\frac{\mathrm{tr}(M)-1}{2}$. Thus, the distance between two 3D rotations is defined as

$$d_3(M_1, M_2) = \arccos\frac{\mathrm{tr}(M_2 M_1^{-1}) - 1}{2} \tag{14}$$

This can be conveniently computed from the quaternion representation: let $\mathbf{q} = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$. Then,

$$\begin{aligned} &d_3(I, R_{\mathrm{q}}(\mathbf{q})) \\ =&\arccos\frac{\mathrm{tr}(R_{\mathrm{q}}(\mathbf{q})) - 1}{2} \\ =&\arccos\frac{2 - 4(b^2 + c^2 + d^2)}{2} \\ =&\arccos(2a^2 - 1) \\ =&2\arccos|a| \end{aligned} \tag{15}$$

So, for $\mathbf{q}, \mathbf{r} \in \mathrm{Sp}(1)$,

$$\begin{aligned} &d_3(R_{\mathrm{q}}(\mathbf{q}), R_{\mathrm{q}}(\mathbf{r})) \\ =&\arccos\frac{\mathrm{tr}(R_{\mathrm{q}}(\mathbf{r})R_{\mathrm{q}}(\mathbf{q})^{-1}) - 1}{2} \\ =&d_3(I, R_{\mathrm{q}}(\mathbf{r}\overline{\mathbf{q}})) \\ =&2\arccos|\mathrm{Re}(\mathbf{r}\overline{\mathbf{q}})| \\ =&2\arccos|\mathbf{r}\cdot\mathbf{q}| \end{aligned} \tag{16}$$

where $\mathbf{r}\cdot\mathbf{q} = \mathrm{Re}(\mathbf{r}\overline{\mathbf{q}})$ is the dot product of $\mathbf{q}$ and $\mathbf{r}$, seen as 4-vectors. Note that the geodesic distance between $\mathbf{q}$ and $\mathbf{r}$ on the unit 3-sphere is just the angle between vectors $\mathbf{q}$ and $\mathbf{r}$ which is

$$d_{\mathrm{q}}(\mathbf{q}, \mathbf{r}) = \arccos(\mathbf{r}\cdot\mathbf{q}) \tag{17}$$

which gives us a nice relationship between the metrics in Sp(1) and SO(3):

$$\begin{aligned} &d_3(R_{\mathrm{q}}(\mathbf{q}), R_{\mathrm{q}}(\mathbf{r})) \\ =&2\min\{d_{\mathrm{q}}(\mathbf{q}, \mathbf{r}), \pi - d_{\mathrm{q}}(\mathbf{q}, \mathbf{r})\} \\ =&2\min\{d_{\mathrm{q}}(\mathbf{q}, \mathbf{r}), d_{\mathrm{q}}(\mathbf{q}, -\mathbf{r})\} \end{aligned} \tag{18}$$

A.3 Embedding of SO(3) in $\mathbb{R}^6$ and $\mathbb{R}^5$

SO(3) can be embedded in $\mathbb{R}^9$ simply by taking the rotation matrix. Note that for any $M \in \mathrm{SO}(3)$, its third column is uniquely determined by its first two columns, since the third column can be computed from the cross product of the first two. So it is unnecessary to keep the last column, which gives us an embedding $\mathrm{SO}(3) \to \mathbb{R}^6$:

$$M \mapsto (m_{11}, m_{21}, m_{31}, m_{12}, m_{22}, m_{32}) \tag{19}$$

The dimensionality can be further reduced. Notice that $m_{12}^2 + m_{22}^2 + m_{32}^2 = 1$, so if we let $h = \sqrt{m_{31}^2 + m_{12}^2 + m_{22}^2 + m_{32}^2} = \sqrt{1 + m_{31}^2}$ then $h > 0$ and (m_{12}, m_{22}, m_{32}) never equals $(0, 0, 0)$. Then, $\frac{1}{h}(m_{12}, m_{22}, m_{32}, m_{31})$ is on the unit 3-sphere and is never $(0, 0, 0, 1)$. We can then use the stereographic projection to project it onto $\mathbb{R}^3$, thus reducing the dimensionality of the embedding by 1. The end result is

$$\begin{gathered} M \mapsto (m_{11}, m_{21}, t\cdot m_{12}, t\cdot m_{22}, t\cdot m_{32}) \\ t = \sqrt{1 + m_{31}^2} + m_{31} \end{gathered} \tag{20}$$

Table 7. Network structure.

Experiment 1&2	Experiment 3	Experiment 4
Input:256 × 256 × 3	Input:256 × 256 × 9	Input:n × 3
Conv(3 → 32,/2)	Conv(9 → 32,/2)	INBlock(3 → 256)
Conv(32 → 32)	Conv(32 → 32)	INBlock(256 → 256)
Conv(32 → 64,/2)	Conv(32 → 64,/2)	INBlock(256 → 256)
Conv(64 → 64)	Conv(64 → 64)	INBlock(256 → 256)
Conv(64 → 128,/2)	Conv(64 → 128,/2)	INBlock(256 → 256)
Conv(128 → 128)	Conv(128 → 128)	
Conv(128 → 256,/2)	Conv(128 → 256,/2)	
Conv(256 → 256)	Conv(256 → 256)	
Conv(256 → 256,/2)	Conv(256 → 256,/2)	
Conv(256 → 256)	Conv(256 → 256)	
	Conv(256 → 256)	
	Conv(256 → 256)	
Conv(256 → 256,/2)	Conv(256 → 256,/2)	
Conv(256 → 256)	Conv(256 → 256)	
	Conv(256 → 256)	Conv(256 → 1024)
	Conv(256 → 256)	MaxPool
FC(4×4×256 → 512)	FC(4×4×256 → 512)	FC(1024 → 512)
FC(512 → 512)	FC(512 → 512)	FC(512 → 512)
FC(512 → 512)	FC(512 → 512)	FC(512 → 512)
FC(512 → 512)	FC(512 → 512)	FC(512 → 512)
FC(512 → 512)	FC(512 → 512)	FC(512 → 512)
	FC(512 → 512)	
	FC(512 → 512)	
(Output layer depends on representation and ensemble size)		

B NETWORK ARCHITECTURE

Table 7 shows the structure of the networks used in our experiments. "Conv" are convolutional layers. "/2" means stride 2. "FC" are fully connected layers. "INBlock" are modified residual blocks: on the residue branch, the convolution layer is followed by an instance normalization layer. The number of input channels in experiment 2 will be explained in section 5.4. All convolution and fully connected layers are followed by Leaky ReLU, they are omitted from the table for clarity.

C ERROR DISTRIBUTION CURVES

The error distribution curve of the three experiments are shown in figures 9, 10 and 11. Each figure shows the testing error distribution by percentile, and the vertical axis is in logarithm scale. For each error curve, the maximum error is marked with a triangle to the right.

D ADDITIONAL THEORETICAL RESULTS

D.1 Quaternion Ensembles

Theorem 6. *There exist continuous functions $f_1, f_2, f_3, f_4 : \mathrm{SO}(3) \to \mathrm{Sp}(1)$ such that for any $M \in \mathrm{SO}(3)$, $R_\mathrm{q}(f_i(M)) = M$ for some $i \in \{1, 2, 3, 4\}$.*

Proof. We give an example of such a set of functions. For clarity, we define them as functions of quaternions. If we ensure that $f_i(\mathbf{q}) = f_i(-\mathbf{q})$ for all $\mathbf{q} \in \mathrm{Sp}(1)$ and $i \in \{1, 2, 3, 4\}$, they will be well-defined functions of rotations. N is the normalization function $N(\mathbf{q}) = \frac{\mathbf{q}}{||\mathbf{q}||}$ defined for all $\mathbf{q} \neq \mathbf{0}$. Let

$$f_1(a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}) = \tag{21}$$
$$\begin{cases} a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} & (a \geq \frac{1}{2}) \\ N((1 - a) + 2ab\mathbf{i} + 2ac\mathbf{j} + 2ad\mathbf{k}) & (0 \leq a < \frac{1}{2}) \\ N((1 + a) + 2ab\mathbf{i} + 2ac\mathbf{j} + 2ad\mathbf{k}) & (-\frac{1}{2} \leq a < 0) \\ -(a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}) & (a < -\frac{1}{2}) \end{cases}$$

The definitions of f_2, f_3 and f_4 are analogous, with b, c and d respectively playing the role of a.

Check that these functions are indeed continuous at case boundaries and that $f_i(-\mathbf{q}) = f_i(\mathbf{q})$. For a unit quaternion $= a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, $a^2 + b^2 + c^2 + d^2 = 1$, so $\max\{a^2, b^2, c^2, d^2\} \geq \frac{1}{4}$. If a^2, b^2, c^2 or d^2 is at least $\frac{1}{4}$, then f_1, f_2, f_3 or f_4 respectively will give the correct output. □

Theorem 7. *For any three continuous functions $f_1, f_2, f_3 : \mathrm{SO}(3) \to \mathrm{Sp}(1)$, there exists $M \in \mathrm{SO}(3)$ such that $d_3(M, R_\mathrm{q}(f_i(M))) = \pi$ for all $i \in \{1, 2, 3\}$.*

Proof. Consider functions $v_i : \mathrm{Sp}(1) \to \mathbb{R}$ defined by $v_i(\mathbf{q}) = \mathbf{q} \cdot f_i(R_\mathrm{q}(\mathbf{q}))$, for $i = 1, 2, 3$. For any $\mathbf{q} \in \mathrm{Sp}(1)$, Since $R_\mathrm{q}(-\mathbf{q}) = R_\mathrm{q}(\mathbf{q})$, $v_i(-\mathbf{q}) = -\mathbf{q} \cdot f_i(R_\mathrm{q}(-\mathbf{q})) = -\mathbf{q} \cdot f_i(R_\mathrm{q}(\mathbf{q})) = -v_i(\mathbf{q})$.

Let $V : \mathrm{Sp}(1) \to \mathbb{R}^3$ defined by $V(\mathbf{q}) = (v_1(\mathbf{q}), v_2(\mathbf{q}), v_3(\mathbf{q}))$, then $V(-\mathbf{q}) = -V(\mathbf{q})$ for any $\mathbf{q} \in \mathrm{Sp}(1)$. By the Borsuk–Ulam theorem,[2] there exists $\mathbf{q}_0 \in \mathrm{Sp}(1)$ such that $V(-\mathbf{q}_0) = V(\mathbf{q}_0)$, then $-V(\mathbf{q}_0) = V(-\mathbf{q}_0) = V(\mathbf{q}_0)$, so $V(\mathbf{q}_0) = \mathbf{0}$, which means $v_1(\mathbf{q}_0) = v_2(\mathbf{q}_0) = v_3(\mathbf{q}_0) = 0$. So for $M_0 = R_\mathrm{q}(\mathbf{q}_0) \in \mathrm{SO}(3)$, $d_3(M_0, f_i(R_0)) = \pi$ for $i = 1, 2, 3$. □

D.2 Euler Angle Ensembles

We extend our proof of error bounds and construction of 4-part ensembles to Euler angle representations.

Note that a continuous map from Euler angles to one of the equivalent quaternions does exist. As an example, consider the extrinsic x-y-z Euler angle. The rotation defined by Euler angles (α, β, γ) is:

$$R_{xyz}(\alpha, \beta, \gamma) = \begin{bmatrix} c_\beta c_\gamma & -c_\alpha s_\gamma + s_\alpha s_\beta c_\gamma & s_\alpha s_\gamma + c_\alpha s_\beta c_\gamma \\ c_\beta s_\gamma & c_\alpha c_\gamma + s_\alpha s_\beta s_\gamma & -s_\alpha c_\gamma + c_\alpha s_\beta s_\gamma \\ -s_\beta & s_\alpha c_\beta & c_\alpha c_\beta \end{bmatrix} \tag{22}$$

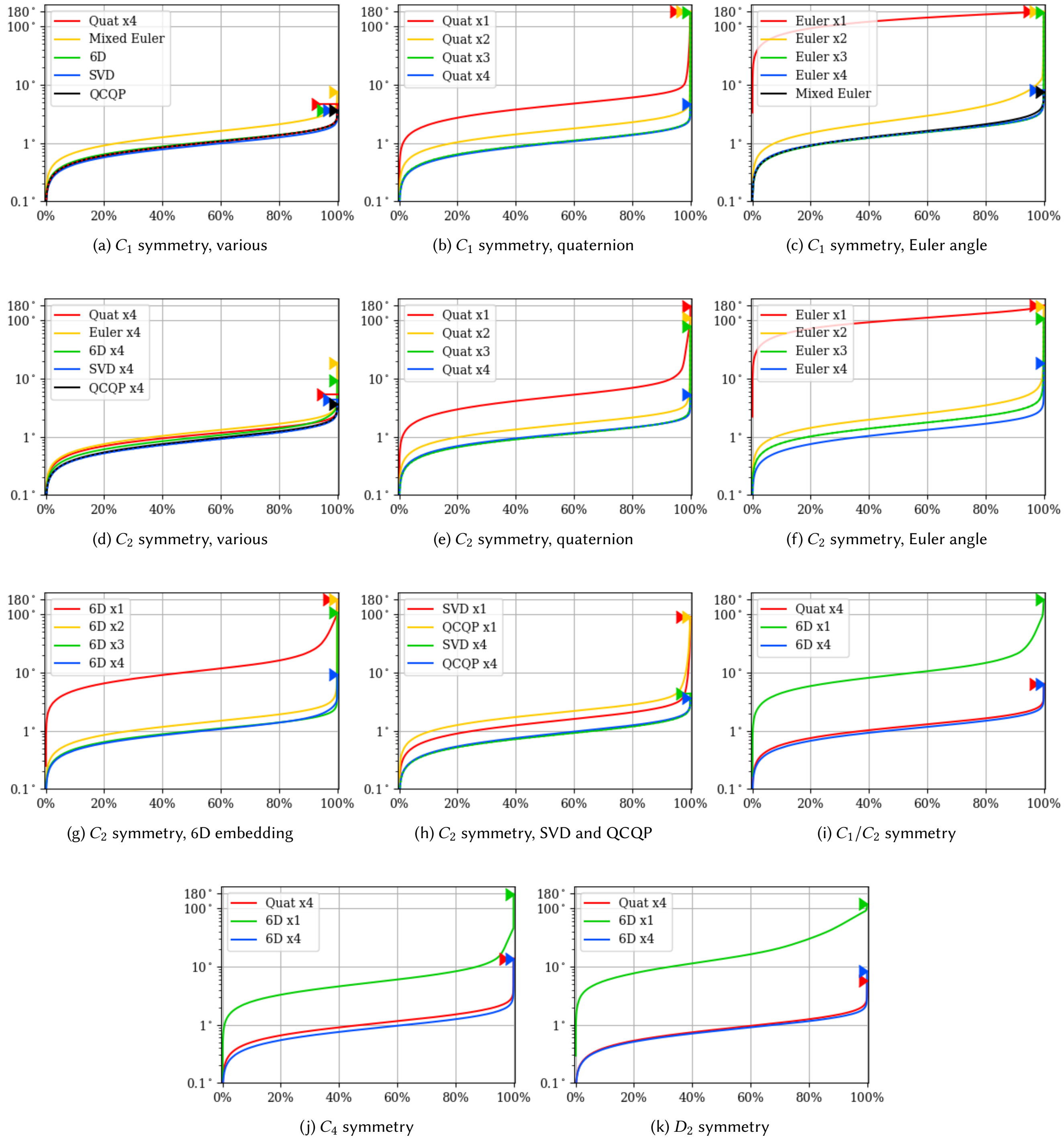
Where $s_\theta = \sin\theta$ and $c_\theta = \cos\theta$. To convert from Euler angles to quaternions, simply convert each of the three elemental rotations and multiply them:

$$\begin{aligned} &Q_{xyz}(\alpha, \beta, \gamma) \\ =&(\cos\frac{\gamma}{2} + \mathbf{k}\sin\frac{\gamma}{2})(\cos\frac{\beta}{2} + \mathbf{j}\sin\frac{\beta}{2})(\cos\frac{\alpha}{2} + \mathbf{i}\sin\frac{\alpha}{2}) \end{aligned} \tag{23}$$

Then we will have $R_\mathrm{q}(Q_{xyz}(\alpha, \beta, \gamma)) = R_{xyz}(\alpha, \beta, \gamma)$.

Corollary 9. *For any continuous function $f : \mathrm{SO}(3) \to \mathbb{R}^3$, there exists $M \in \mathrm{SO}(3)$ such that $d_3(M, R_{xyz}(f(M))) = \pi$.*

[2] See e.g. page 174 of [Hatcher 2002]



(a) C_1 symmetry, various

(b) C_1 symmetry, quaternion

(c) C_1 symmetry, Euler angle

(d) C_2 symmetry, various

(e) C_2 symmetry, quaternion

(f) C_2 symmetry, Euler angle

(g) C_2 symmetry, 6D embedding

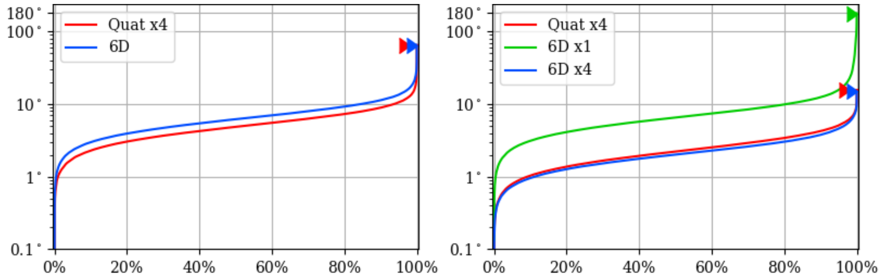
(h) C_2 symmetry, SVD and QCQP

(i) C_1/C_2 symmetry

(j) C_4 symmetry

(k) D_2 symmetry

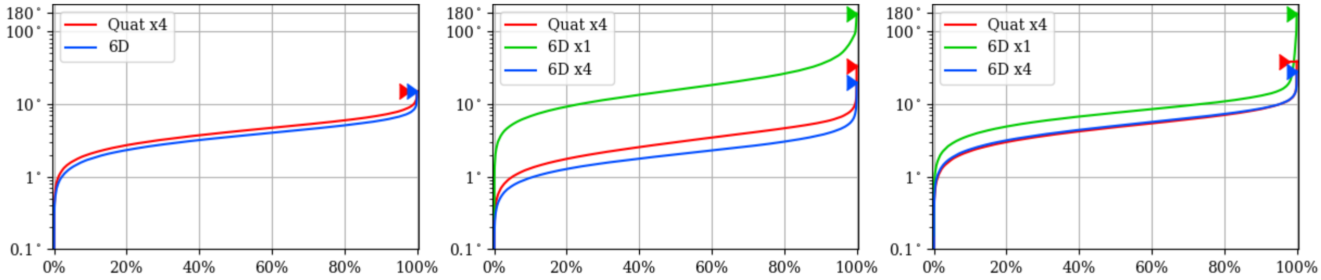
Fig. 9. Loss distribution of different configuration and output representation for the toy problem.



(a) Chairs

(b) Tables

Fig. 10. Loss distribution of different configuration and output representation for rotation estimation from images.



(a) Chairs

(b) Tables

(c) Mixed

Fig. 11. Loss distribution of different configuration and output representation for rotation estimation from point clouds.

Proof. Assume that there exists continuous $f : \mathrm{SO}(3) \longrightarrow \mathbb{R}^3$ such that for any $M \in \mathrm{SO}(3)$, $d_3(M, R_{xyz}(f(M))) \neq \pi$. Then, $g = Q_{xyz} \circ f$ is a continuous function such that for any $M \in \mathrm{SO}(3)$, $d_3(M, R_\mathrm{q}(g(M))) \neq \pi$, which is a violation of theorem 3. □

The argument for other differently ordered Euler angles are completely analogous.

Unlike corollary 9 which follows easily from theorem 3, constructing an ensemble of four functions using Euler angle representation is not as straightforward, due to a peculiarity of Euler angles: gimbal lock.

Euler angle representation decomposes a rotation into a chain of three elemental rotations around three fixed axes, and gimbal lock happens if applying the second rotation to the first axis causes it to coincide with the third axis. In such cases changing the first rotation or the third rotation would have the same effect, and the rotation degree of freedom is reduced by one. As an example, in extrinsic x-y-z Euler angles, gimbal lock happens whenever the second rotation angle is $\pm\frac{\pi}{2}$.

We pick a gimbal lock position and show that any continuous function from $\mathrm{SO}(3)$ to $\mathbb{R}^3$ will always produce error somewhere in its neighbourhood.

Theorem 10. *Let U be any neighborhood of $R_{xyz}(0, \frac{\pi}{2}, 0)$ in* SO(3)*. There exists no continuous function $f : U \longrightarrow \mathbb{R}^3$ such that $R_{xyz}(f(M)) = M$ for every $M \in U$.*

Proof. Assume that such a function f exists. It can be checked that $Q_{xyz}(\theta, \phi, \theta) \cdot Q_{xyz}(0, \frac{\pi}{2}, 0) = \cos(\frac{\pi}{4} - \frac{\phi}{2})$, so for $0 \leq \phi \leq \pi$, $d_3(R_{xyz}(\theta, \phi, \theta), R_{xyz}(0, \frac{\pi}{2}, 0)) = \frac{\pi}{2} - \phi$. U is a neighbourhood, so it contains an open ball centered at $R_{xyz}(0, \frac{\pi}{2}, 0)$, so there exists $\epsilon > 0$ such that $R_{xyz}(\theta, \phi, \theta) \in U$ for all $\theta \in \mathbb{R}$ and $\frac{\pi}{2} - \epsilon < \phi < \frac{\pi}{2}$.

If $R_{xyz}(\alpha, \beta, \gamma) = R_{xyz}(\alpha', \beta', \gamma')$, but $\beta \neq k\pi + \frac{\pi}{2}$ for any $k \in \mathbb{Z}$, we must have $(\alpha', \beta', \gamma') = (\alpha{+}2m\pi, \beta{+}2n\pi, \gamma{+}2k\pi)$ or $(\alpha', \beta', \gamma') = (\alpha + (2m+1)\pi, -\beta + (2n+1)\pi, \gamma + (2k+1)\pi)$ for some $m, n, k \in \mathbb{Z}$. In any case, let

$$g(\theta, \phi) = \frac{f^{[1]}(R_{xyz}(\theta, \phi, \theta)) - \theta}{\pi} \tag{24}$$

where $f^{[1]}$ is the first coordinate of f. Then for $\theta \in \mathbb{R}$ and $\frac{\pi}{2} - \epsilon < \phi < \frac{\pi}{2}$, g is continuous and integer-valued, so it must be constant. But since $R_{xyz}(0, \phi, 0) = R_{xyz}(2\pi, \phi, 2\pi)$, we have $g(0, \phi) - g(2\pi, \phi) = 2$, which is a contradiction. □

If all functions use the same sequence of Euler angle axes, then they all have the same set of gimbal lock positions, and these cannot be handled correctly however many functions are used. So, to construct a correct ensemble, Euler angles with different axis sequences

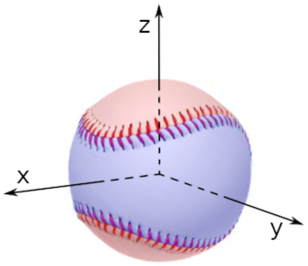


Fig. 12. Deciding which locations should use the x-y-z or x-z-y type of Euler angle

are necessary. Extrinsic x-z-y Euler angles have a distinct set of gimbal lock positions: the rotation defined by x-z-y Euler angle (α, β, γ) is

$$R_{xzy}(\alpha, \beta, \gamma) = \begin{bmatrix} c_\beta c_\gamma & -c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ s_\beta & c_\alpha c_\beta & -s_\alpha c_\beta \\ c_\beta s_\gamma & -c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma \end{bmatrix} \tag{25}$$

Compare with R_{xyz} in equation 22: at the gimbal lock positions $\beta = \pm\frac{\pi}{2}$, the first column of $R_{xyz}(\alpha, \beta, \gamma)$ is $(0, \mp 1, 0)$ while the first column of $R_{xzy}(\alpha, \beta, \gamma)$ is $(0, 0, \pm 1)$. Indeed, a gimbal-lock-free ensemble of four sets of Euler angles can be constructed by mixing these two types of Euler angles. The detailed proof is laborious but the idea is simple.

Put an arrow on the unit sphere at $(1, 0, 0)$, pointing up towards $(0, 0, 1)$. Then a 3D rotation can be uniquely identified by its effect on this arrow. In particular, if we think of $(0, 0, \pm 1)$ as the north/south pole and represent the rotation as x-y-z Euler angle (α, β, γ), then β determines the latitude, γ the longitude, and α the heading. x-z-y Euler angles can be interpreted similarly if we instead take $(0, \pm 1, 0)$ as the poles and the xOz great circle as the equator.

We will be using four functions, two of the x-y-z type and two of the x-z-y type. First, each rotation is assigned either to the x-y-z or the x-z-y type by the location of the arrow. To avoid gimbal lock, the range of latitude assigned to each type must not contain the polar region. Then, to eliminate non-contractible loops, the arrow must not be allowed to "go around the globe" within each subset, so the range of longitude must not be the full circle. The way to do this is perfectly illustrated by a baseball, as shown in figure 12: the blue region is assigned to the x-y-z type, and the red region is assigned to the x-z-y type.

Then, within each type, each rotation is assigned to one of the two functions by the heading of the arrow: the arrow must not be allowed to rotate around itself for a full circle within the subset of rotations assigned to each function, which is another kind of non-contractible loop. For example, arrows pointing eastwards relative to the meridians are assigned to one function, while those pointing westwards are assigned to the other function. To summarize, SO(3) is divided into four overlapping contractible subsets with the following ranges of latitude, longitude and heading in table 8:

Table 8. The subset assigned to each function, specified by Euler angle type and range

#	Type	Latitude(β)	Longitude(γ)	Heading(α)
1	x-y-z	$[-\frac{\pi}{4}, \frac{\pi}{4}]$	$[-\frac{3\pi}{4}, \frac{3\pi}{4}]$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$
2				$[\frac{\pi}{2}, \frac{3\pi}{2}]$
3	x-z-y		$[\frac{\pi}{4}, \frac{7\pi}{4}]$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$
4				$[\frac{\pi}{2}, \frac{3\pi}{2}]$

The remaining task is then to define the four functions so that they produce the correct Euler angle in its assigned range and is continuous on the whole SO(3). We formally state the theorem about mixed Euler angle 4-part ensembles and give a proof:

Theorem 11. *There exists continuous functions $f_1, f_2, f_3, f_4 : \mathrm{SO}(3) \to \mathbb{R}^3$ such that for any $M \in \mathrm{SO}(3)$, at least one of the following is equal to M: $R_{xyz}(f_1(M))$, $R_{xyz}(f_2(M))$, $R_{xzy}(f_3(M))$ and $R_{xzy}(f_4(M))$.*

The proof uses essentially the same charts as given in [Grafarend and Kühnel 2011]. But as we have stated, an ensemble has slightly more structure than an atlas since each function is required to be defined and continuous everywhere. In general, given an atlas, we shrink/expand each chart a bit to create a core region and a margin so that the core regions of all chars still cover the whole manifold. We let each function compute its corresponding chart in the core region and be constant beyond the chart, and blend these two cases in the margin. We give such a set of functions explicitly.

Proof. Let $\operatorname{atan2}_{\triangleleft}$ be the usual atan2 function with range $(-\pi, \pi]$ and $\operatorname{atan2}_{\triangleright}$ be an alternative version with range $[0, 2\pi)$. Let $\mathcal{T}$ be a parametrized family of functions, defined as follows:

$$\mathcal{T}(a_1, a_2, a_3, a_4)(x) = \begin{cases} 0 & (x \leq a_1) \\ \frac{x - a_1}{a_2 - a_1} & (a_1 < x \leq a_2) \\ 1 & (a_2 < x \leq a_3) \\ \frac{a_4 - x}{a_4 - a_3} & (a_3 < x \leq a_4) \\ 0 & (a_4 < x) \end{cases} \tag{26}$$

That is, $\mathcal{T}(a_1, a_2, a_3, a_4)$ is a piecewise linear function defined by connecting the points $(a_1, 0)$, $(a_2, 1)$, $(a_3, 1)$, $(a_4, 0)$ in $[a_1, a_4]$ and constantly 0 beyond that range. Define these instances of $\mathcal{T}$:

$$\begin{aligned} \mathcal{T}_{\alpha\triangleleft} &= \mathcal{T}(-\frac{2\pi}{3}, -\frac{\pi}{2}, \frac{\pi}{2}, \frac{2\pi}{3}) \\ \mathcal{T}_{\alpha\triangleright} &= \mathcal{T}(\frac{\pi}{3}, \frac{\pi}{2}, \frac{3\pi}{2}, \frac{5\pi}{3}) \\ \mathcal{T}_{\beta} &= \mathcal{T}(-\frac{\pi}{3}, -\frac{\pi}{4}, \frac{\pi}{4}, \frac{\pi}{3}) \\ \mathcal{T}_{\gamma\triangleleft} &= \mathcal{T}(-\frac{5\pi}{6}, -\frac{3\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{6}) \\ \mathcal{T}_{\gamma\triangleright} &= \mathcal{T}(\frac{\pi}{6}, \frac{\pi}{4}, \frac{7\pi}{4}, \frac{11\pi}{6}) \end{aligned} \tag{27}$$

Let $M \in \mathrm{SO}(3)$ be a rotation matrix. Define these functions:

$$\alpha_1(M) = \begin{cases} \mathrm{atan2}_{\triangleleft}(m_{32}, m_{33}) & (m_{31} \neq \pm 1) \\ \pi & (m_{31} = \pm 1) \end{cases} \tag{28}$$

$$\alpha_2(M) = \begin{cases} \mathrm{atan2}_{\triangleright}(m_{32}, m_{33}) & (m_{31} \neq \pm 1) \\ 0 & (m_{31} = \pm 1) \end{cases}$$

$$\beta_1(M) = -\sin^{-1} m_{31}$$

$$\gamma_1(M) = \begin{cases} \mathrm{atan2}_{\triangleleft}(m_{21}, m_{11}) & (m_{31} \neq \pm 1) \\ \pi & (m_{31} = \pm 1) \end{cases}$$

$$t_1(M) = \mathcal{T}_{\alpha\triangleleft}(\alpha_1(M)) \cdot \mathcal{T}_{\beta}(\beta_1(M)) \cdot \mathcal{T}_{\gamma\triangleleft}(\gamma_1(M))$$

$$t_2(M) = \mathcal{T}_{\alpha\triangleright}(\alpha_2(M)) \cdot \mathcal{T}_{\beta}(\beta_1(M)) \cdot \mathcal{T}_{\gamma\triangleleft}(\gamma_1(M))$$

$$f_1(M) = (\alpha_1(M), \beta_1(M), \gamma_1(M)) \cdot t_1(M)$$

$$f_2(M) = (\alpha_2(M), \beta_1(M), \gamma_1(M)) \cdot t_2(M)$$

$$\alpha_3(M) = \begin{cases} \mathrm{atan2}_{\triangleleft}(-m_{23}, m_{22}) & (m_{21} \neq \pm 1) \\ \pi & (m_{21} = \pm 1) \end{cases}$$

$$\alpha_4(M) = \begin{cases} \mathrm{atan2}_{\triangleright}(-m_{23}, m_{22}) & (m_{21} \neq \pm 1) \\ 0 & (m_{21} = \pm 1) \end{cases}$$

$$\beta_3(M) = \sin^{-1} m_{21}$$

$$\gamma_3(M) = \begin{cases} \mathrm{atan2}_{\triangleright}(m_{31}, m_{11}) & (m_{21} \neq \pm 1) \\ 0 & (m_{21} = \pm 1) \end{cases}$$

$$t_3(M) = \mathcal{T}_{\alpha\triangleleft}(\alpha_3(M)) \cdot \mathcal{T}_{\beta}(\beta_3(M)) \cdot \mathcal{T}_{\gamma\triangleright}(\gamma_3(M))$$

$$t_4(M) = \mathcal{T}_{\alpha\triangleright}(\alpha_4(M)) \cdot \mathcal{T}_{\beta}(\beta_3(M)) \cdot \mathcal{T}_{\gamma\triangleright}(\gamma_3(M))$$

$$f_3(M) = (\alpha_3(M), \beta_3(M), \gamma_3(M)) \cdot t_3(M)$$

$$f_4(M) = (\alpha_4(M), \beta_3(M), \gamma_3(M)) \cdot t_4(M)$$

We prove that f_1, f_2, f_3 and f_4 defined as such meet the requirements in the theorem. Consider the continuity of f_1. For any $M \in \mathrm{SO}(3)$, find any $\alpha, \beta, \gamma \in \mathbb{R}$ such that $R_{xyz}(\alpha, \beta, \gamma)$.

Case 1: $M \in R_{xyz}((-\pi, \pi), (-\frac{\pi}{2}, \frac{\pi}{2}), (-\pi, \pi))$. $-1 < m_{31} = -\sin\beta < 1$, so in this range $\alpha_1(M) = \mathrm{atan2}_{\triangleleft}(m_{32}, m_{33})$ and $\gamma_1 = \mathrm{atan2}_{\triangleleft}(m_{21}, m_{11})$. Furthermore, $m_{33} = \cos\alpha\cos\beta > -1$ and $m_{11} = \cos\beta\cos\gamma > -1$, so α_1 and γ_1 are continuous. $\mathcal{T}$ is always continuous. so t_1 is continuous, and so f_1 is continuous.

Case 2: $M \in R_{xyz}((-\frac{4\pi}{3}, -\frac{2\pi}{3}) \cup (\frac{2\pi}{3}, \frac{4\pi}{3}), \mathbb{R}, \mathbb{R})$. $\mathcal{T}_{\alpha\triangleleft}(\alpha_1(M)) = 0$, so $f_1(M) = (0, 0, 0)$ is constant and thus continuous.

Case 3: $M \in R_{xyz}(\mathbb{R}, (-\frac{2\pi}{3}, -\frac{\pi}{3}) \cup (\frac{\pi}{3}, \frac{2\pi}{3}), \mathbb{R})$. $\mathcal{T}_{\beta}(\beta_1(M)) = 0$, so $f_1(M) = (0, 0, 0)$ is constant and thus continuous.

Case 4: $M \in R_{xyz}(\mathbb{R}, \mathbb{R}, (-\frac{7\pi}{6}, -\frac{5\pi}{6}) \cup (\frac{5\pi}{6}, \frac{7\pi}{6}))$. $\mathcal{T}_{\gamma\triangleleft}(\gamma_1(M)) = 0$, so $f_1(M) = (0, 0, 0)$ is constant and thus continuous.

It is obvious that if the range of α, β and γ all fall outside the ranges specified in cases 2, 3 and 4 respectively, then they must satisfy case 1. So any point in SO(3) is covered by at least one case, so f_1 is continuous on all of SO(3). Additionally, when $\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, $\beta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ and $\gamma \in [-\frac{3\pi}{4}, \frac{3\pi}{4}]$, $\alpha_1(M) = \alpha$, $\beta_1(M) = \beta$, $\gamma_1(M) = \gamma$ and $\mathcal{T}_{\alpha\triangleleft}(\alpha_1(M)) = \mathcal{T}_{\beta}(\beta_1(M)) = \mathcal{T}_{\gamma\triangleleft}(\gamma_1(M)) = 1$, so $f_1(M) = (\alpha, \beta, \gamma)$.

Similarly, we can prove that f_2, f_3 and f_4 are continuous, and each gives the correct Euler angles if the input is in their respective range specified in table 8. Several things in the proof above need to be changed for each function accordingly:

For f_2 and f_4, the range of α in case 1 is $(0, 2\pi)$ and the range of α in case 4 is $(-\frac{\pi}{3}, \frac{\pi}{3}) \cup (\frac{5\pi}{3}, \frac{7\pi}{3})$.

For f_3 and f_4, use R_{xzy} instead of R_{xyz}. The range of γ in case 1 is $(0, 2\pi)$ and the range of γ in case 4 is $(-\frac{\pi}{6}, \frac{\pi}{6}) \cup (\frac{11\pi}{6}, \frac{13\pi}{6})$.

Now we need to prove that any rotation in SO(3) falls within the correct range of at least one of f_1, f_2, f_3 and f_4. For $M \in \mathrm{SO}(3)$, let $m^* = \max\{m_{11}, -m_{11}, m_{21}, -m_{21}, m_{31}, -m_{31}\}$.

Case 1: $m^* \in \{-m_{21}, m_{11}, m_{21}\}$. One of the *x-y-z* type functions (f_1 and f_2) will be correct. $2m_{31}^2 \leq \max\{m_{21}^2, m_{11}^2\} + m_{31}^2 \leq m_{11}^2 + m_{21}^2 + m_{31}^2 = 1$, so $-\frac{\sqrt{2}}{2} \leq m_{31} \leq \frac{\sqrt{2}}{2}$, so $-\frac{\pi}{4} \leq \beta_1(M) \leq \frac{\pi}{4}$. at least one each of the following γ cases and α cases will be satisfied:

Case 1-γ-1: $m^* = -m_{21}$. $m_{21} \leq m_{11} \leq -m_{21}$, $-\frac{3\pi}{4} \leq \gamma_1(M) \leq -\frac{\pi}{4}$.

Case 1-γ-2: $m^* = m_{11}$. $-m_{11} \leq m_{21} \leq m_{11}$, $-\frac{\pi}{4} \leq \gamma_1(M) \leq \frac{\pi}{4}$.

Case 1-γ-3: $m^* = m_{21}$. $-m_{21} \leq m_{11} \leq m_{21}$, $\frac{\pi}{4} \leq \gamma_1(M) \leq \frac{3\pi}{4}$.

Case 1-α-1: $m_{33} \geq 0$. $-\frac{\pi}{2} \leq \alpha_1(M) \leq \frac{\pi}{2}$. $R_{xyz}(f_1(M)) = M$.

Case 1-α-2: $m_{33} \leq 0$. $\frac{\pi}{2} \leq \alpha_2(M) \leq \frac{3\pi}{2}$. $R_{xyz}(f_2(M)) = M$.

Case 2: $m^* \in \{-m_{31}, -m_{11}, m_{31}\}$ One of the *x-z-y* type functions (f_3 and f_4) will be correct. $2m_{21}^2 \leq \max\{m_{31}^2, m_{11}^2\} + m_{21}^2 \leq m_{11}^2 + m_{21}^2 + m_{31}^2 = 1$, so $-\frac{\sqrt{2}}{2} \leq m_{21} \leq \frac{\sqrt{2}}{2}$, so $-\frac{\pi}{4} \leq \beta_3(M) \leq \frac{\pi}{4}$. at least one each of the following γ cases and α cases will be satisfied:

Case 2-γ-1: $m^* = -m_{31}$. $m_{31} \leq m_{11} \leq -m_{31}$, $\frac{\pi}{4} \leq \gamma_3(M) \leq \frac{3\pi}{4}$.

Case 2-γ-2: $m^* = -m_{11}$. $m_{11} \leq m_{31} \leq -m_{11}$, $\frac{3\pi}{4} \leq \gamma_3(M) \leq \frac{5\pi}{4}$.

Case 2-γ-3: $m^* = m_{31}$. $-m_{31} \leq m_{11} \leq m_{31}$, $\frac{5\pi}{4} \leq \gamma_3(M) \leq \frac{7\pi}{4}$.

Case 2-α-1: $m_{22} \geq 0$. $-\frac{\pi}{2} \leq \alpha_3(M) \leq \frac{\pi}{2}$. $R_{xzy}(f_3(M)) = M$.

Case 2-α-2: $m_{22} \leq 0$. $\frac{\pi}{2} \leq \alpha_4(M) \leq \frac{3\pi}{2}$. $R_{xzy}(f_4(M)) = M$.

In conclusion, f_1, f_2, f_3 and f_4 are continuous functions from SO(3) to $\mathbb{R}^3$ and for any rotation $M \in \mathrm{SO}(3)$, at least one of $R_{xyz}(f_1(M))$, $R_{xyz}(f_2(M))$, $R_{xzy}(f_3(M))$ and $R_{xzy}(f_4(M))$ is equal to M. □

D.3 Symmetric Cases

Corollary 4. *Let G be a finite, non-trivial subgroup* SO(3). *Then there does not exist continuous map $f : \mathrm{SO}(3)/G \to \mathrm{SO}(3)$ such that $\varphi_G \circ f$ is the identity map.*

Here we discuss what corollary 4 did not address: to prove a tight lower bound for the maximum error that must occur for a continuous function that computes the rotation of a symmetrical 3D object.

We start from introducing the possible rotation symmetrics of a 3D shape. The finite subgroups of SO(3) are, in Schoenflies notations:

- Cyclic symmetry C_n, the rotation symmetry of regular n-pyramids.
- Dihedral symmetry D_n, the rotation symmetry of regular n-prisms.
- Polyhedral symmetry, T, O and I, the rotation symmetry of regular tetrahedra, octahedra and icosahedra.

We will work in quaternion representations and make extensive use of the preimage of each group under R_{q}, i.e. the set of all quaternions that represent some rotation in that group. They are finite subgroups of Sp(1), with twice the size:

Table 9. The finite rotation symmetry groups

G	$\lvert G\rvert$	Elements of $\widehat{G}$
C_n	n	$(\cos\frac{k\pi}{n}, 0, 0, \sin\frac{k\pi}{n}), 0 \le k < 2n$
D_n	$2n$	$(\cos\frac{k\pi}{n}, 0, 0, \sin\frac{k\pi}{n}), 0 \le k < 2n$ $(0, \cos\frac{k\pi}{n}, \sin\frac{k\pi}{n}, 0), 0 \le k < 2n$
T	12	$\mathrm{Perm}(\pm 1, 0, 0, 0)$ $(\pm\frac{1}{2}, \pm\frac{1}{2}, \pm\frac{1}{2}, \pm\frac{1}{2})$
O	24	$\mathrm{Perm}(\pm 1, 0, 0, 0)$ $(\pm\frac{1}{2}, \pm\frac{1}{2}, \pm\frac{1}{2}, \pm\frac{1}{2})$ $\mathrm{Perm}(\pm\frac{\sqrt{2}}{2}, \pm\frac{\sqrt{2}}{2}, 0, 0)$
I	60	$\mathrm{Perm}(\pm 1, 0, 0, 0)$ $(\pm\frac{1}{2}, \pm\frac{1}{2}, \pm\frac{1}{2}, \pm\frac{1}{2})$ $\mathrm{EvenPerm}(0, \pm\frac{\sqrt{5}+1}{4}, \pm\frac{1}{2}, \pm\frac{\sqrt{5}-1}{4})$

- The preimage of C_n is the cyclic group of twice the order, C_{2n}.
- The preimage of D_n is the dicyclic group Dic_n.
- The preimages of T, O and I are the binary polyhedral groups $2T$, $2O$ and $2I$.

Generically we refer to the preimage of G, $R_q^{-1}[G]$, as $\widehat{G}$. We have $\mathrm{SO}(3)/G \cong \mathrm{Sp}(1)/\widehat{G}$ (Informally, we "multiplied" both the "numerator" and the "denominator" by 2). The size of G and all the quaternion elements of $\widehat{G}$ are listed in table 9. Perm() refers to all permutations of the sequence and EvenPerm() refers to all even permutations.

Since we will use quaternions extensively, we first write the definition of rotation distance under symmetry in terms of quaternions:

$$\begin{aligned} & d_G(R_q(\mathbf{p}), R_q(\mathbf{q})) \\ =& \min_{S\in G} d_3(R_q(\mathbf{p}), R_q(\mathbf{q})S) \\ =& \min_{\mathbf{s}\in\widehat{G}} 2d_q(\mathbf{p}, \mathbf{qs}) \end{aligned} \tag{29}$$

Our proof for the lower bound of maximum error is valid for all rotation groups containing any rotation of angle π, which includes every rotation group except C_n with an odd n. We believe however that for C_n with an odd n the conclusion is the same. Assume that G is one of the applicable groups. Let f be any continuous function from $\mathrm{SO}(3)/G$ to $\mathrm{SO}(3)$. Now pick one of the two preimages of $f(\varphi_G(I_{3\times 3}))$ under R_q as $\mathbf{s}$. Define $\widehat{f} : \mathrm{Sp}(1) \to \mathrm{Sp}(1)$ by the following method:

For any $\mathbf{r} \in \mathrm{Sp}(1)$, let h be a path from $\mathbf{1}$ to $\mathbf{r}$, that is, h is a continuous function from $[0, 1]$ to $\mathrm{Sp}(1)$ such that $h(0) = \mathbf{1}$ and $h(1) = \mathbf{r}$. Then $f \circ \varphi_G \circ R_q \circ h$ is a path in $\mathrm{SO}(3)$ starting from $f(\varphi_G(I_{3\times 3})) = R_q(\mathbf{s})$. By the lifting property of covering spaces, $f \circ \varphi_G \circ R_q \circ h$ lifts to a unique path in $\mathrm{Sp}(1)$ starting from $\mathbf{s}$. Let g be that path. We prove that the value of $g(1)$ is independent of the choice of h:

for any two paths h_1 and h_2 from $\mathbf{1}$ to $\mathbf{r}$, suppose that $f\circ\varphi_G\circ R_q\circ h_1$ lifts to g_1 from $\mathbf{s}$ and $f \circ \varphi_G \circ R_q \circ h_2$ lifts to g_2 from $\mathbf{s}$. Since $\mathrm{Sp}(1)$ is simply connected, there exists a homotopy of paths from h_1 to h_2. That is, there exists a continuous function $H : [0, 1]\times[0, 1] \to \mathrm{Sp}(1)$ such that $H(t, 0) = h_1(t)$ and $H(t, 1) = h_2(t)$ for all $0 \le t \le 1$ and $H(0, u) = \mathbf{1}$ and $H(1, u) = \mathbf{r}$ for all $0 \le u \le 1$. Now let $N = f \circ \varphi_G \circ R_q \circ H$, then $N : [0, 1] \times [0, 1] \to \mathrm{SO}(3)$ is a homotopy of paths in $\mathrm{SO}(3)$, $N(\cdot, 0) = f\circ\varphi_G\circ R_q\circ h_1$ and $N(\cdot, 1) = f\circ\varphi_G\circ R_q\circ h_2$. $R_q \circ g_1 = N(\cdot, 0)$. By the lifting property of covering spaces, N lifts to a unique homotopy $\widehat{N}$ in $\mathrm{Sp}(1)$ such that $\widehat{N}(\cdot, 0) = g_1$. Since $R_q(\widehat{N}(0, u)) = R_q(\mathbf{s})$, for any u we must have $\widehat{N}(0, u) = \mathbf{s}$ or $\widehat{N}(0, u) = -\mathbf{s}$. $\widehat{N}(0, u)$ is also continuous in u, so it must be constant. So we must have $\widehat{N}(0, 1) = \widehat{N}(0, 0) = \mathbf{s}$. Likewise we must have $\widehat{N}(1, 1) = \widehat{N}(1, 0)$.

$\widehat{N}(\cdot, 1)$ is a lift of $N(\cdot, 1)$ starting from $\widehat{N}(0, 1) = \mathbf{s}$. Since g_2 is the unique such lift, we have $\widehat{N}(\cdot, 1) = g_2$. So, $g_2(1) = \widehat{N}(1, 1) = \widehat{N}(1, 0) = g_1(1)$.

Now we have proven that the value of $g(1)$ is independent of the choice of h. So we can define $\widehat{f}(\mathbf{r}) = g(1)$. $\widehat{f}(\mathbf{r})$ is a preimage of $f(\varphi_G(R_q(\mathbf{r})))$. By the construction of $\widehat{f}$, we can see that if h is a path from $\mathbf{p}$ to $\mathbf{q}$, then the lift of $f \circ \varphi_G \circ R_q \circ h$ starting from $\widehat{f}(\mathbf{p})$ will end at $\widehat{f}(\mathbf{q})$.

Let $\widehat{G} = R_q^{-1}[G]$ be the binary polyhedral group corresponding to G. Since G contains a rotation of angle π, $\widehat{G}$ must contain a pure quaternion. Let $\mathbf{q}_0 = x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \in \widehat{G}$ be such a quaternion. Define $\mathbf{q}(t) = \cos\frac{\pi}{2}t + \sin\frac{\pi}{2}t \cdot \mathbf{q}_0$. Now for any $\mathbf{p} \in S^3$, let $r_{\triangleleft}(t) = \mathbf{p}\mathbf{q}(t)$ and $r_{\triangleright}(t) = \mathbf{p}\mathbf{q}_0\mathbf{q}(t)$, then $r_{\triangleleft}(t)$ is a path from $\mathbf{p}$ to $\mathbf{p}\mathbf{q}_0$, $r_{\triangleright}(t)$ is a path from $\mathbf{p}\mathbf{q}_0$ to $\mathbf{p}\mathbf{q}_0^2 = -\mathbf{p}$ and $r_{\triangleright}(t) = r_{\triangleleft}(t)\mathbf{q}_0$

Note that $\varphi_G(R_q(\mathbf{p})) = \varphi_G(R_q(\mathbf{pq}))$ if $\mathbf{q} \in \widehat{G}$. So, $\varphi_G \circ R_q \circ r_{\triangleleft}$ and $\varphi_G \circ R_q \circ r_{\triangleright}$ are the same path in $\mathrm{SO}(3)/G$ and that path is a loop. Then, $v = f \circ \varphi_G \circ R_q \circ r_{\triangleleft} = f \circ \varphi_G \circ R_q \circ r_{\triangleright}$ is a loop in $\mathrm{SO}(3)$. Consider the lift of v in $\mathrm{Sp}(1)$. Let $\mathbf{u} = \widehat{f}(\mathbf{p})$, then the lift of v in $\mathrm{Sp}(1)$ must start and end at either $\mathbf{u}$ or $-\mathbf{u}$. If the lifted path starting from $\mathbf{u}$ also ends at $\mathbf{u}$, then the lifted path starting from $-\mathbf{u}$ must end at $-\mathbf{u}$, for otherwise the reverse of v lifts to two paths path from $\mathbf{u}$ to two different destinations $\mathbf{u}$ and $-\mathbf{u}$ which is impossible. Likewise, if the lifted path starting from $\mathbf{u}$ ends at $-\mathbf{u}$, then the lifted path starting from $-\mathbf{u}$ must end at $\mathbf{u}$.

Now join $r_{\triangleleft}$ and $r_{\triangleright}$ as a long path r: $r(t) = \mathbf{q}(2t)$. r is a path from $\mathbf{p}$ to $-\mathbf{p}$. The first half of r coincides with $r_{\triangleleft}$ and the second half of r coincides with $r_{\triangleright}$. Consider the lift of $f \circ \varphi_G \circ R_q \circ r$. It can be obtained by joining two lifts of v, lets call $\widehat{v}_{\triangleleft}$ and $\widehat{v}_{\triangleright}$. If $\widehat{v}_{\triangleleft}$ goes from $\mathbf{u}$ to $-\mathbf{u}$, then $\widehat{v}_{\triangleright}$ goes from $-\mathbf{u}$ to $\mathbf{u}$. If $\widehat{v}_{\triangleleft}$ loops from $\mathbf{u}$ back to $\mathbf{u}$, then $\widehat{v}_{\triangleleft}$ does the same. In any case, the whole lifted path is a loop from $\mathbf{u}$ back to $\mathbf{u}$.

The start of this path is $\mathbf{u} = \widehat{f}(\mathbf{p})$. By the construction of $\widehat{f}$, the end point equals $\widehat{f}(-\mathbf{p})$. So for any $\mathbf{p} \in \mathrm{Sp}(1)$, $\widehat{f}(\mathbf{p}) = \widehat{f}(-\mathbf{p})$.

Now consider the error on input generated by $R_q(\mathbf{p})$. Recall that $\widehat{f}(\mathbf{p})$ is a preimage of $f(\varphi_G(R_q(\mathbf{p})))$. We have

$$\begin{aligned} & d_G(R_q(\mathbf{p}), f(\varphi_G(R_q(\mathbf{p})))) \\ =& \min_{\mathbf{q}\in\widehat{G}} 2d_q(\mathbf{p}, \widehat{f}(\mathbf{p})\mathbf{q}) \end{aligned} \tag{30}$$

Let us now select 3 quaternions $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$ from $\widehat{G}$. The method of selection will be considered later. Let $l_i(\mathbf{p}) = \mathbf{p}\cdot\widehat{f}(\mathbf{p}) - \mathbf{p}\cdot(\widehat{f}(\mathbf{p})\mathbf{q}_i)$, then

$$\begin{aligned} & l_i(-\mathbf{p}) \\ =&(-\mathbf{p})\cdot\widehat{f}(-\mathbf{p}) - (-\mathbf{p})(\cdot\widehat{f}(-\mathbf{p})\mathbf{q}_i) \\ =&-\mathbf{p}\cdot\widehat{f}(\mathbf{p}) + \mathbf{p}\cdot(\widehat{f}(\mathbf{p})\mathbf{q}_i) \\ =&-l_i(\mathbf{p}) \end{aligned} \tag{31}$$

Let $L(\mathbf{p}) = (l_1(\mathbf{p}), l_2(\mathbf{p}), l_3(\mathbf{p}))$. By the Borsuk–Ulam theorem, there exists $\mathbf{p}_0$ such that $L(\mathbf{p}_0) = \mathbf{0}$. Then we have

$$\mathbf{p}_0\cdot\widehat{f}(\mathbf{p}_0) - \mathbf{p}_0\cdot(\widehat{f}(\mathbf{p}_0)\mathbf{q}_i) = 0, \quad i = 1, 2, 3 \tag{32}$$

$\mathbf{a}\cdot(\mathbf{bc}) = \mathrm{Re}(\overline{\mathbf{a}}(\mathbf{bc})) = \mathrm{Re}((\overline{\mathbf{a}}\mathbf{b})\mathbf{c}) = (\overline{\mathbf{b}\mathbf{a}})\cdot\mathbf{c}$ for any quaternions $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$, so equivalently

$$(\overline{\widehat{f}(\mathbf{p}_0)\mathbf{p}_0})\cdot(\mathbf{q}_i - \mathbf{1}) = 0, \quad i = 1, 2, 3 \tag{33}$$

In non-degenerate cases, this will give us three independent linear equations in the four components of $\overline{\widehat{f}(\mathbf{p}_0)\mathbf{p}_0}$. Together with $||\overline{\widehat{f}(\mathbf{p}_0)\mathbf{p}_0}|| = 1$, $\overline{\widehat{f}(\mathbf{p}_0)\mathbf{p}_0}$ can be uniquely determined up to negation. From this, we can compute

$$\begin{aligned} & d_G(R_\mathrm{q}(\mathbf{p}_0), f(\varphi_G(R_\mathrm{q}(\mathbf{p}_0)))) \\ =& \min_{\mathbf{q}\in\widehat{G}} 2d_\mathrm{q}(\mathbf{p}_0, \widehat{f}(\mathbf{p}_0)\mathbf{q}) \\ =& \min_{\mathbf{q}\in\widehat{G}} 2\cos^{-1}(\mathbf{p}_0\cdot(\widehat{f}(\mathbf{p}_0)\mathbf{q})) \\ =& \min_{\mathbf{q}\in\widehat{G}} 2\cos^{-1}((\overline{\widehat{f}(\mathbf{p}_0)\mathbf{p}_0})\cdot\mathbf{q}) \end{aligned} \tag{34}$$

which gives us a lower bound for $\max_\mathbf{p} d_G(R_\mathrm{q}(\mathbf{p}), f(\varphi_G(R_\mathrm{q}(\mathbf{p}))))$, i.e. the lower bound for the maximum error that must occur. It turns out that the highest lower bound so obtained is exactly the largest possible value of d_G and so must be tight.

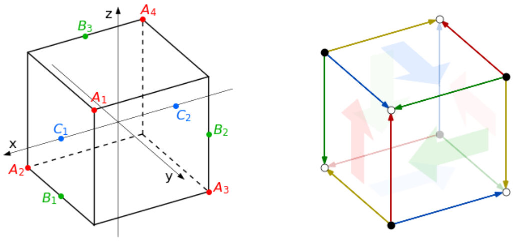
We list one possible optimal choice of all choices of $\mathbf{q}_i$ and the resulting error bound in table 10. C_n is a degenerate case where one of the $\mathbf{q}_i$ is redundant, but all solutions of $\overline{\widehat{f}(\mathbf{p}_0)\mathbf{p}_0}$ give the same bound.

For the construction of four open contractible subsets that cover $\mathrm{SO}(3)/G$, we demonstrate a general method with a concrete example. Let's take $G = D_2$. D_2 consists of four elements: the identity, and the rotation of π around each of the x, y and z axes. As an abstract group it is the Klein four-group.

We visualize the structure of $\mathrm{Sp}(1)/\mathrm{Dic}_2 \cong \mathrm{SO}(3)/D_2$ first. $\mathrm{Dic}_2 = \{\pm 1, \pm\mathbf{i}, \pm\mathbf{j}, \pm\mathbf{k}\}$. It is also called the "quaternion group" Q_8, not to be confused with the "multiplicative group of unit quaternions". Remember that $\mathrm{Sp}(1)/\mathrm{Dic}_2$ is the set of left cosets of Dic_2 in $\mathrm{Sp}(1)$. We can select one point from each coset of Dic_2 in $\mathrm{Sp}(1)$ as a representative. For example, in each coset, we select the element that is the closest to 1. If there is a tie, take all of the closest points. Points in this representative set correspond one-to-one with points in $\mathrm{Sp}(1)/\mathrm{Dic}_2$, except on the boundary: the boundary are exactly the points causing a tie, so multiple points on the boundary actually represent the same coset. To go from the representative set to $\mathrm{Sp}(1)/\mathrm{Dic}_2$, we just need to identify these equivalent points.

Table 10. Error bound for each symmetry.

G	Choice of $\mathbf{q}_i$	Error bound
C_n	$(-1, 0, 0, 0)$ $(\cos\frac{\pi}{n}, 0, 0, \sin\frac{\pi}{n})$	π
D_n	$(\cos\frac{\pi}{n}, 0, 0, \sin\frac{\pi}{n})$ $(0, 1, 0, 0)$ $(0, \cos\frac{\pi}{n}, \sin\frac{\pi}{n}, 0)$	$\arccos(-\sin^2\frac{\pi}{2n})$
T	$(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, -\frac{1}{2})$ $(\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{2})$	$\frac{\pi}{2}$
O	$(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0, 0)$ $(\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 0)$ $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$	$\arccos\frac{2\sqrt{2}-1}{4}$
I	$(\frac{\sqrt{5}+1}{4}, \frac{1}{2}, 0, \frac{\sqrt{5}-1}{4})$ $(\frac{\sqrt{5}+1}{4}, \frac{\sqrt{5}-1}{4}, \frac{1}{2}, 0)$ $(\frac{\sqrt{5}+1}{4}, 0, \frac{\sqrt{5}-1}{4}, \frac{1}{2})$	$\arccos\frac{3\sqrt{5}-1}{8}$



(a)

(b)

Fig. 13. Structure of $\mathrm{Sp}(1)/\mathrm{Dic}_2$. (a): each set of points marked with the same color should be identified. (b): vertices or edges or faces should be glued together, with matching directions, if they are marked with the same color.

Indeed, cutting a space apart and specifying how the boundary points should be glued back is a common way for visualizing a space. For example, constructing a torus by gluing together opposite edges of a square is an introductory example in topology.

The representative set is the set of points in $\mathrm{Sp}(1)$ that are closer to $\mathbf{1}$ than any of $\pm\mathbf{i}$, $\pm\mathbf{j}$ and $\pm\mathbf{k}$. Its shape is a bulging cube with vertices at $\frac{1}{2} \pm \frac{1}{2}\mathbf{i} \pm \frac{1}{2}\mathbf{j} \pm \frac{1}{2}\mathbf{k}$. Although geometrically a spherical polyhedron in 4D space, topologically this makes no difference so for simplicity we "project" it from (w, x, y, z) to (x, y, z) coordinates and draw it as a cube with flat faces, as shown in figure 13.

In figure 13 (a), we marked three set of points. A's are on four of the vertices:

$$\begin{aligned} A_1 &= \frac{1}{2}(1+\mathbf{i}+\mathbf{j}+\mathbf{k}) \qquad (35)\\ A_2 &= \frac{1}{2}(1+\mathbf{i}-\mathbf{j}-\mathbf{k})\\ A_3 &= \frac{1}{2}(1-\mathbf{i}+\mathbf{j}-\mathbf{k})\\ A_4 &= \frac{1}{2}(1-\mathbf{i}-\mathbf{j}+\mathbf{k}) \end{aligned}$$

They represent the same element of $\mathrm{Sp}(1)/\mathrm{Dic}_2$. For example, $-\mathbf{i} \in \mathrm{Dic}_2$ and $A_1(-\mathbf{i}) = A_4$, so A_1 and A_4 are in the same coset of Dic_2 in $\mathrm{Sp}(1)$ and should be identified.

Similarly, the B's on the center of three of the edges are equivalent:

$$\begin{aligned} B_1 &= \frac{\sqrt{3}}{3}(1+\mathbf{i}-\mathbf{k}) \qquad (36)\\ B_2 &= \frac{\sqrt{3}}{3}(1-\mathbf{i}+\mathbf{j})\\ B_3 &= \frac{\sqrt{3}}{3}(1-\mathbf{j}+\mathbf{k}) \end{aligned}$$

As do the C's on the center of two of the faces:

$$\begin{aligned} C_1 &= \frac{\sqrt{2}}{2}(1+\mathbf{i}) \qquad (37)\\ C_2 &= \frac{\sqrt{2}}{2}(1-\mathbf{i}) \end{aligned}$$

These equivalence relations are summarized in figure 13 (b). To get $\mathrm{Sp}(1)/\mathrm{Dic}_2$, we glue together two sets of four vertices each, four sets of three edges each and three sets of two faces each. The opposite faces are glued together with a 90 degree twist.

Keeping boundary equivalences in mind, we can then find four contractible subsets that cover this cube. The four subsets are shown in figure 14. Each row is one subset. The first subset is formed by starting from the center of the cube and growing a region until it is close to touching the surface. The second subset is formed by starting from the center of each face and growing several regions simultaneously until they are close to touching each other. The third and fourth subsets are formed similarly to the second one but starting from centers of edges and from vertices instead.

Now we glue these regions together according to the boundary equivalences. The first subset is just a cube. For the second subset, each region is a pyramid, and pairs of them need to be glued together along their base, so we get three 4-bipyramids. Similarly, the third subset consists of four 3-bipyramids, and the fourth subset consists of two tetrahedra. Each of these polyhedra is homeomorphic to a ball and thus contractible, so any disjoint union of them is also contractible.

To see how these subsets cover the whole $\mathrm{Sp}(1)/\mathrm{Dic}_2$, consider any point. If it is not close to a face, then it is covered by the first subset. Otherwise, if it is not close to an edge, then it is covered by the second subset. Otherwise, if it is not close to a vertex, then it is covered by the third subset. Otherwise it is covered by the fourth subset.

In general, any $\mathrm{SO}(3)/G$ can be similarly constructed by gluing together surface points of a suitable polyhedron, and the strategy of forming four subsets by growing from the center, face-centers, edge-centers and vertices respectively can be applied.

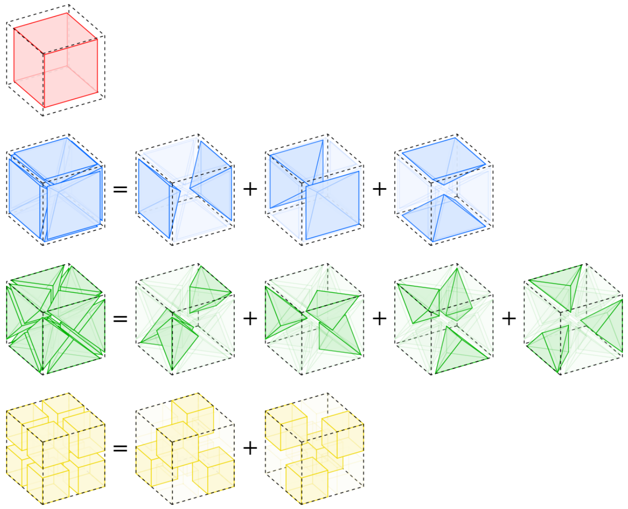


Fig. 14. Four contractible subsets that cover $\mathrm{Sp}(1)/\mathrm{Dic}_2$. Each row is one subset, shown as the disjoint union of (topological) balls.

Note that while elaborate constructions are given to demonstrate that an ensemble is able to solve rotation estimation problems, with or without symmetry, the knowledge of these constructions are not required to train such an ensemble: the classifier needs not be trained to classify the points in $\mathrm{SO}(3)$ or $\mathrm{SO}(3)/G$ according to some known subsets. Our training method lets the network discover these subsets automatically without explicit supervision, and that is why it is called a "self-selecting" ensemble.