Algebraically rigorous quaternion framework for the neural network pose estimation problem

Chen Lin
CCB & CCM, Flatiron Institute, New York, NY, USA
clin@flatironinstitute.org

Andrew J. Hanson
Dept. of Computer Science, Indiana University, Bloomington, IN, USA
hansona@indiana.edu

Sonya M. Hanson
CCB & CCM, Flatiron Institute, New York, NY, USA
shanson@flatironinstitute.org

Abstract

The 3D pose estimation problem – aligning pairs of noisy 3D point clouds – is a problem with a wide variety of real-world applications. Here we focus on the use of quaternion-based neural network approaches to this problem and apparent anomalies that have arisen in previous efforts to resolve them. In addressing these anomalies, we draw heavily from the extensive literature on closed-form methods to solve this problem. We suggest that the major concerns that have been put forward could be resolved using a simple multi-valued training target derived from rigorous theoretical properties of the rotation-to-quaternion map of Bar-Itzhack. This multi-valued training target is then demonstrated to have good performance for both simulated and ModelNet targets. We provide a comprehensive theoretical context, using the quaternion adjugate, to confirm and establish the necessity of replacing single-valued quaternion functions by quaternions treated in the extended domain of multiple-charted manifolds.

1. Introduction

The basic problem of pose estimation, or specifically 3D point cloud alignment, is important in a wide variety of disciplines, including machine vision, astronautics, robotics, and proteomics, with further applications for the general rotation learning problem in fields ranging from scene reconstruction and cryo-electron microscopy to self-driving cars and robotics. Efforts seeking how to effectively represent rotations in the context of neural networks have raised concerns about possible deficiencies in the use of quaternions [24, 43]. Additionally, Xiang & Li [37, 38] have observed that multiple heuristic charts for a quaternion atlas can resolve some of these objections. Here we provide a rigorous theoretical basis for the quaternion atlas by exploiting the quaternion adjugate matrix to understand and resolve the apparent anomalies, concluding that quaternions have no fundamental defects for this task.

Algebraic Solutions to Pose Estimation We begin by noting the existence of extensive literature on linear algebra techniques for solving the point-cloud alignment problem in closed form, referred to alternatively as the "Generalized Procrustes Problem" [8], the Kabsch problem [19], or RMSD (root mean squared deviation) [6, 10, 15, 16, 28], depending on the field. This is important because the algebraic insights in this literature can inform neural network approaches to similar problems. This algebraic approach completely solves the 3D pose estimation problem with matched input data, with or without noise, and serves as an initial benchmark to evaluate machine learning approaches to this class of problems. As shown in Fig. (1) the algebraic solution provides an order of magnitude improvement on the 3D pose estimation problem in comparison to the current best quaternion-based machine learning approach [24]. Regarding algebraic solutions to the 3D pose estimation problem, there are three threads of relevant work: the SVD method [28], a direct approach using matrix square roots [15], both of which apply to any dimension, and an extensive quaternion-eigensystem based literature [6, 10, 15, 16] that applies specifically to dimensions 2, 3, and 4. We remark that, technically, the existence of closed-form quaternion and rotation solutions to the RMSD task renders neural network approaches to solving this problem irrelevant. However, there are many other problems, for example, the alignment of noisy projection data, that can be phrased as quaternion extraction tasks, but for which no closed form solution exists. We will focus here on the 3D quaternion-based problem and its corresponding machine-learning methods as a prototype for all related rotation-data extraction applications that can be phrased as a quaternion discovery task.

Machine Learning, Cloud Matching, and Quaternions Setting aside the availability of closed-form algebraic solutions, it is well-known (see, e.g., [18]) that neural networks can in principle provide arbitrarily accurate approximations to any function. In fact, there are numerous effective machine-learning approaches to point-cloud registration and LiDAR odometry problems that have achieved excellent results without fully incorporating quaternions. Typical examples include 3DRegNet by Pais et al [23], PointNet [26], and its extensions [41], along with more complicated architectures such as Deep Closest Point [35]. Quaternions have proven to be efficient and convenient rotation representations in SLAM problems, including both Graph-based [3] and Visual-Inertial based [33] pipelines. Recently, quaternions have also found utility in contexts distinct from our presentation here, such as some learning-based LiDAR odometry frameworks [17]. Since quaternions are well-known to avoid problems such as gimbal lock that plague 3D rotation matrices expressed in terms of Euler angles, and to possess desirable properties such as existing on a simply connected manifold with smooth and uniform distance computation [16], it is thus logical to use quaternions to represent rotations in the context of neural networks. However, several recent implementations of this approach [24, 27, 43] have reported that quaternions may be a deficient representation for this task. Zhou et al [43], for example, describe singularities at 180° and a discontinuous behavior that appears incompatible with employing quaternions themselves as the training objective for cloud-matching. To resolve this issue, Xiang & Li [37, 38] have introduced heuristically defined four-fold charts to locally overcome the absence, as noted by Zhou et al [43], of a global quaternion function.

Our contributions. Our main result is to explain precisely the sources of the quaternion singularities observed in the machine learning context, and how the correct atlas of charts is defined by the quaternion adjugate. By extending and exploiting an elegant method of Bar-Itzhack [2] for obtaining the corresponding quaternion from a numerical 3D rotation matrix, we are able to establish a mathematically justifiable, and distinct, form for *multivalued* quaternion-based rotation training closely related to the *ad hoc* suggestion of Xiang & Li [37, 38], which lacks the rigorous theoretical background that we provide. In particular, we connect the quaternion adjugate, first described in [11], to the four distinct forms of the output quaternion required to successfully implement the learning task.

We conclude that the noted apparent anomalies in the use of quaternions in machine learning for the pose estimation problem are not due to quaternions themselves, but to the fact that conventional neural networks learn single-valued functions, while quaternions need to be treated in the extended domain of multi-valued manifolds. Readers with sufficient background in the theory of manifolds, such as the quaternion's three-sphere $\mathbf{S}^3$, will recognize this as an instance of the mandatory requirement to cover such a manifold with an "atlas" of partial charts and transition functions related to their local coordinates [14]. The take-home lesson is that single quaternions may not be computable across their whole domain, but a family of local charts *is* computable. At least one chart will be regular in a neighborhood of each of the possible singularities in the rotation-matrix-to-quaternion map. Clearly, one should therefore take care not to apply neural networks to functions in situations such as those involving nontrivial topological manifolds without carefully studying the computability features.

2. The Quaternion Context

In this section, we briefly discuss the context of quaternions themselves, review the role of quaternion representations in the solution of the point-cloud matching problem, and summarize the observations in machine learning applications that led us to this investigation.

Why Quaternions? Quaternions were discovered by William Rowan Hamilton in 1843, the culmination of his long effort to extend imaginary numbers to additional dimensions. His result succeeded in representing rotations in 3D in a manner exactly parallel to the way complex numbers represent rotations in 2D, using one real component and the three imaginary numbers ijk instead of just i. In addition, quaternions showed that the full structure of 3D rotation frames (triple axes) corresponded to simple 4D points $q = [q_0, q_1, q_2, q_3]$ on the hypersphere $q \cdot q = 1$. This leads to the fact that each reflected pair of quaternions, q and $-q$, corresponds to the same 3D orientation, even though they are different points in quaternion space. Finally, the structure of hyperspherical quaternion points can be exploited to define mathematical curves producing smooth 3D rotation sequences [30], a typical computer animation task, and many other sophisticated computations such as quantifying the statistical properties of orientation-frame clouds [12] and smoothing complex choices of tube and surface orientation in mathematical shape modeling [9].

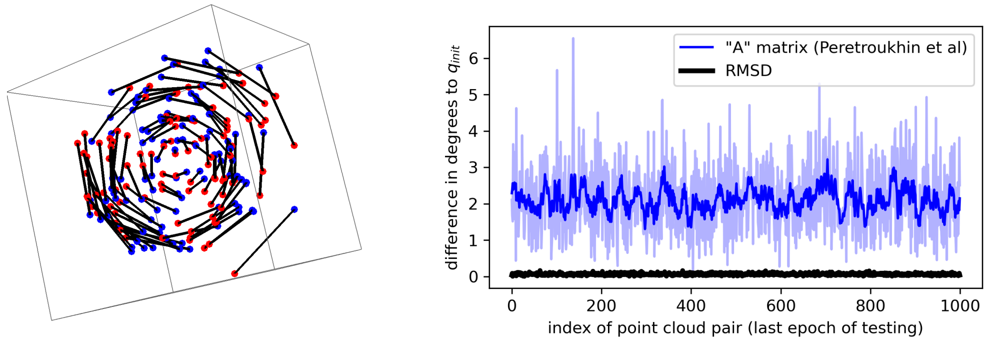

Figure 1. **RMSD shows an order of magnitude improvement over NN methods for the point-cloud alignment problem.** (left) A typical point-cloud data set, with the reference cloud Y in blue, and the rotated sample cloud X in red, with matched points connected. (right) Comparison of current best quaternion-based neural network approach to finding optimal rotations between two point clouds [24] (blue, with sliding average in dark blue), to the closed-form RMSD method in black. Similar experiments described in Sec. 4.

Quaternions and RMSD Point-Cloud Alignment. The analytic approach to the 3D pose estimation problem can be expressed as a quaternion eigenvalue problem by starting with a $3 \times K$ matrix $\mathbf{Y}$ forming a list of K 3D reference points, a matching list of rotated sample data points $\mathbf{X}$, and the task of transforming $\mathbf{X}$ back to its optimal alignment with $\mathbf{Y}$ (see, e.g., [10] for a recent review). The basic procedure starts with the least-squares loss function [34],

$$\text{loss}(R, \mathbf{X}, \mathbf{Y}) = \|R \cdot \mathbf{X} - \mathbf{Y}\|^2 = \sum_{k=1}^{K} \|R \cdot x_k - y_k\|^2 , \quad (1)$$

whose minimum with respect to variations in the rotation matrix R is our desired answer R_{opt}. Following Horn's approach [16] to the RMSD problem, we make two transformations: first, we express the 3D rotation matrix R in its classic form $R(q)$ using quadratic quaternion polynomials,

$$R(q) = \begin{bmatrix} {q_0}^2 + {q_1}^2 - {q_2}^2 - {q_3}^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & {q_0}^2 - {q_1}^2 + {q_2}^2 - {q_3}^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2 \end{bmatrix} , \quad (2)$$

where we will always assume unit-length quaternions, $q \cdot q = {q_0}^2 + {q_1}^2 + {q_2}^2 + {q_3}^2 = 1$, enforcing determinant $\det R(q) = (q \cdot q)^3 = +1$. We will see further versions of this fundamental equation later on. Second, if we multiply out Eq. (1) and apply $R \cdot R^{\text{t}} = I_3$, the only remaining non-constant term is a negative cross-term that allows us to rewrite the loss minimization task as a maximization task for the cross-term

$$\Delta(E) = \text{tr}\, R(q) \cdot E \;=\; q \cdot M(E) \cdot q \; , \quad (3)$$

where $E = \left[\mathbf{X} \cdot \mathbf{Y}^{\text{t}}\right]$ is the 3×3 *cross-covariance matrix* of the data. By examining the coefficients of the $q_i q_j$ terms in Eq. (3) from Eq. (2) we obtain the traceless, symmetric 4×4 *profile matrix*

$$M(E) = \begin{bmatrix} D_0 & E_{yz} - E_{zy} & E_{zx} - E_{xz} & E_{xy} - E_{yx} \\ E_{yz} - E_{zy} & D_x & E_{xy} + E_{yx} & E_{zx} + E_{xz} \\ E_{zx} - E_{xz} & E_{xy} + E_{yx} & D_y & E_{yz} + E_{zy} \\ E_{xy} - E_{yx} & E_{zx} + E_{xz} & E_{yz} + E_{zy} & D_z \end{bmatrix} , \quad (4)$$

where $D_0 = E_{xx} + E_{yy} + E_{zz}$, $D_x = E_{xx} - E_{yy} - E_{zz}$, $D_y = -E_{xx} + E_{yy} - E_{zz}$, and $D_z = -E_{xx} - E_{yy} + E_{zz}$. The maximal value of $\Delta(E)$ occurs when q is the (normalized) eigenvector of the maximal eigenvalue λ_{opt} of the 4×4 matrix $M(E)$, that is, $\Delta(E) = \lambda_{\text{opt}}$ (see [4, 13, 6, 7, 16, 5, 20, 21]).

These eigenvectors can be obtained from the adjugate of the characteristic equation of the profile matrix that is solved by λ_{opt},

$$\chi(E) = [(M(E)) - \lambda_{\text{opt}} I_4] \;\Rightarrow\; det\chi = 0. \quad (5)$$

The adjugate matrix, $A(\chi)$, is defined by $\chi \cdot A(\chi) = \det(\chi) I_4 \equiv 0$. Although this fact has not been widely exploited, the adjugate columns are by construction proportional, with independent scale factors, to the single optimizing eigenvector q. But since the profile matrix is symmetric, so is its adjugate, thus all the *rows* are also proportional to the same eigenvectors: therefore the structure of the adjugate matrix of the RMSD optimization problem must be proportional to the rank one matrix of "adjugate quaternion variables," defined as $q_{ij} = q_i q_j$, giving

$$\text{adjugate: } A(q_{ij}) \;=\; \begin{bmatrix} q_{00} & q_{01} & q_{02} & q_{03} \\ q_{01} & q_{11} & q_{12} & q_{13} \\ q_{02} & q_{12} & q_{22} & q_{23} \\ q_{03} & q_{13} & q_{23} & q_{33} \end{bmatrix} . \quad (6)$$

We will find other independent occurrences of this fundamental matrix later on.

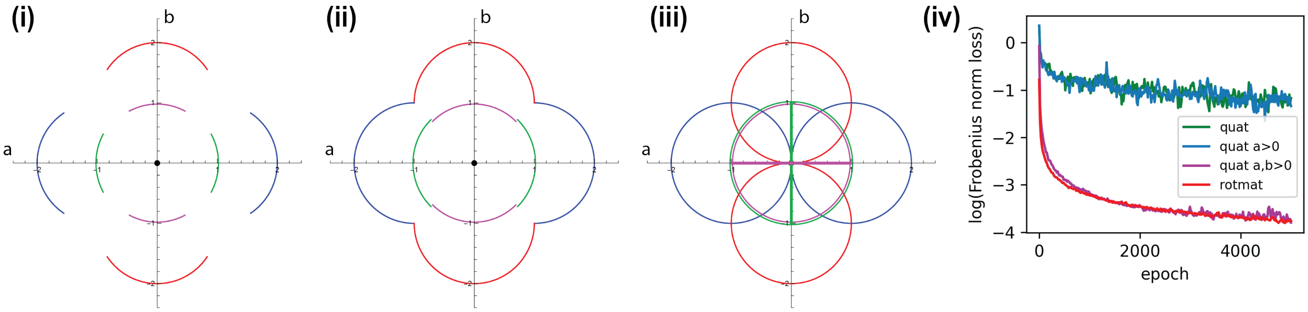

Figure 2. **Normalization of the quaternion space results in singularities.** Here we illustrate how quaternion space in 2D, with $0 \leq \theta < 4\pi$, must be covered by multiple alternative solutions (Eq. (8) & Eq. (9)) depending on the actual 2D cosine and sine rotation parameters. The blue circles are the paths of $\pm(1+c,\, s)$, mapping to the green half-circle quaternions in (a, b), failing at $c = -1$, $a = 0$. The red circles are the paths of $\pm(s,\, 1-c)$, mapping to the magenta half-circle quaternions in (a, b), which fail at $c = +1$, $b = 0$. Subfigure (i) shows partial completion, (ii) shows the full four-part map, and (iii) shows the singularities occurring at $c = \pm 1$. (iv) This Zhou-style sanity test, learning the rotation matrix directly from the quaternion, verifies that restricting training to the quaternion sector similar to $(a > 0, b > 0)$ in (i,ii,iii) indeed succeeds in recovering the rotation matrix, while training to all quaternions or just a single hemisphere does not.

Much of the literature determines the maximal eigenvalue using numerical methods [32], though the answer can be computed directly from Cardano's quartic equation solution [1], and also one can bypass quaternions entirely and compute R_{opt} directly from E using singular-value-decomposition [28] or the matrix-square-root method [15]. We note that there are subtle problems in determining the maximal eigenvector from the maximal eigenvalue due to the appearance of zeros in the eigenvector components. Resolving these singularities is closely related to the selection of optimal variables in the classic rotation to quaternion method of Sheppard [29]. One of the main objectives of this paper is to advocate the use of the full quaternion adjugate matrix as the training target in the neural network context to avoid the effects of such zeros.

Learning Quaternions in Neural Networks. Several recent machine-learning implementations of quaternion-based point-cloud alignment problems have reported that quaternions are a deficient representation for this task. Zhou et al [43], for example, describe singularities at 180°, discontinuous behavior, and an error spectrum that appears incompatible with the effective employment of quaternions as a learning target. They propose an alternative way of encoding the output rotation using 5D or 6D subsets of the 3D rotation matrix itself. Peretroukhin et al [24], in contrast, train to an intermediate matrix and appear to improve on the results of [43]. Fig. (1) compares the angular error of the best performing of these [24] to the closed-form quaternion solution, which is an order of magnitude more accurate. Other authors studying rotation-output problems, including [27, 37, 38], have also observed problems with using single quaternion outputs, and suggest alternatives such as multiple-valued representations, and other approaches that circumvent using the pure quaternion representation. As the linear-algebra based quaternion approach outlined above has been used successfully since at least 1968, we were led to examine how these machine learning approaches used quaternions and to clarify what was and wasn't working.

3. Geometry of the Quaternion Manifold

In this section, we discuss the origin of quaternion discontinuities in the context of two dimensions that exhibits all the relevant properties. We then show how the Bar-Itzhack method [2] in 3D leads directly to the quaternion adjugate as a complete and rigorous resolution of the discontuity problem. Finally, we review the existing neural network approaches in the context of the algebraic solution to the pose estimation problem and the quaternion adjugate approach.

3.1. The 2D Rotation-to-Quaternion Map

We begin by considering the 2D rotation-to-quaternion map, as it has sufficient complexity to reveal our main points. Suppose we have a numerical 2D rotation matrix R, which by definition is orthogonal with determinant equal to one; we can write R in two ways,

$$R = \begin{bmatrix} a^2 - b^2 & -2ab \\ 2ab & a^2 - b^2 \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \qquad (7)$$

$$\det R = 1 \;\rightarrow\; c^2 + s^2 = 1,\;\; (a^2 + b^2)^2 = 1\,.$$

The first equation is a 2D version of the quaternion expression in Eq. (2) with $a = \cos(\theta/2)$, $b = \sin(\theta/2)$, and the second is the noise-free rotation matrix parameterized by

$c = \cos\theta$, $s = \sin\theta$. Our task is now, thinking of (a,b) as unknown symbols, to *derive* them using *only* the ideal measured rotation data (c,s). We can see that we have two ways of solving for (a,b). First we use the dependence $a^2+b^2 = 1$ to eliminate b in the upper left component,

$$\left.\begin{array}{rcl} a^2-b^2=2a^2-1 &=& c\\ 2ab &=& s \end{array}\right\}$$

$$\text{Solve for } (a,b) \;\rightarrow\; a = \pm\frac{\sqrt{1+c}}{\sqrt{2}},\quad b = \pm\frac{s}{\sqrt{2}\sqrt{1+c}}\,. \tag{8}$$

Then we similarly eliminate a,

$$\left.\begin{array}{rcl} a^2-b^2=1-2b^2 &=& c\\ 2ab &=& s \end{array}\right\}$$

$$\text{Solve for } (a,b) \;\rightarrow\; a = \pm\frac{s}{\sqrt{2}\sqrt{1-c}}\quad b = \pm\frac{\sqrt{1-c}}{\sqrt{2}}\,. \tag{9}$$

Equations (8) and (9) are in fact identical *in principle*. But clearly the first solution is impossible for $a \sim 0$, or $c \sim -1$, a perfectly legal rotation, and the second solution is impossible for $b \sim 0$, or $c \sim +1$, also perfectly legal. Thus the unnormalized components must be retained in order to choose an expression that avoids the singularities. The set of four locally non-singular charts covering the entire a, b quaternion domain is illustrated in Fig. (2)(i,ii,iii). In Fig. (2)(iv), we use the Zhou sanity test, learning the rotation matrix directly from the quaternion, to show that training to rotations with all quaternion signs fails because that quaternion is not a function, while training to just positive-signed quaternions, analogous to $a > 0$, $b > 0$, succeeds because that restriction admits a unique function as a solution.

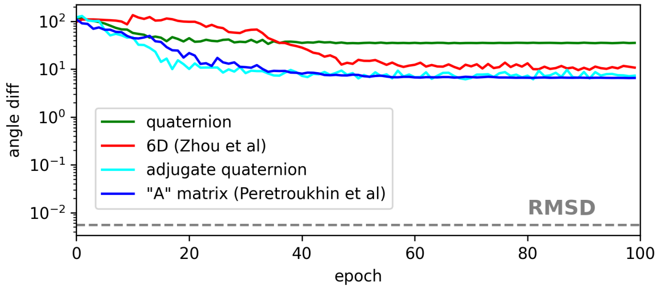

Figure 3. **Learning using the quaternion adjugate performs among the best quaternion-based methods for NN rotation estimation.** Using a PointNet architecture [25] and training with simulated random point clouds, the quaternion adjugate (cyan) evades singularities and substantially outperforms the lone quaternion training target (green), slightly better than the 6D rotation representation of Zhou et al (red) [43], and on par with Peretroukhin et al (blue) [24], but still an order of magnitude away from the analytical RMSD solution (grey dotted line).

3.2. The 3D Rotation-to-Quaternion Map: The Bar-Itzhack Method and the Quaternion Adjugate

Now we move on to 3D rotations. An appealing and mathematically-sound approach to the 3D version of the problem we just solved in 2D is the Bar-Itzhack method [2], which we will now briefly outline. The fundamental idea is that if one computes the Fröbenius norm of the difference of two rotation matrices, one, say $R(r)$ in terms of an unknown quaternion $r = (r_0, r_1, r_2, r_3)$ or their quadratic adjugates, $r_{ij} = r_i r_j$, and the other, say Q, a possibly noisily measured 3×3 rotation matrix candidate, one recovers Eq. (3) with Q^{t} replacing the cross-covariance matrix E . We easily verify this by observing that Bar-Itzhack's Fröbenius loss expands as

$$\begin{aligned}\mathrm{loss}_{\mathrm{B.I.}} &= \mathrm{tr}\,(R(r)-Q)\cdot(R(r)-Q)^{\mathrm{t}}\\ &= \mathrm{tr}\left(R(r)\cdot R(r)^{\mathrm{t}} + Q\cdot Q^{\mathrm{t}}\right) - 2\,\mathrm{tr}\,R(r)\cdot Q^{\mathrm{t}}\,.\end{aligned}$$

We convert the problem to the maximization of the cross-term,

$$\Delta(Q(\text{measured})) = \mathrm{tr}\,R(r)\cdot Q^{\mathrm{t}} \;=\; r\cdot M(Q^{\mathrm{t}})\cdot r\,, \tag{10}$$

where $M(Q^{\mathrm{t}})$ is just the 4×4 symmetric, traceless, real matrix Eq. (4) with the elements of E replaced by the corresponding measured matrix numbers of Q^{t}. Using Eq. (2) as ideal data we can write Q as $Q(q)$ and M as $M(q)$, where the maximal eigenvalue of M(q) is $3(q\cdot q)$. This leads to our main insight that the adjugate of the characteristic matrix,

$$\chi = [M(q) - 3(q\cdot q)I_4]\,, \tag{11}$$

is simply, up to a scale factor,

$$\mathrm{Adjugate}(\chi) = \begin{bmatrix} {q_0}^2 & q_0q_1 & q_0q_2 & q_0q_3\\ q_0q_1 & {q_1}^2 & q_1q_2 & q_1q_3\\ q_0q_2 & q_1q_2 & {q_2}^2 & q_2q_3\\ q_0q_3 & q_1q_3 & q_2q_3 & {q_3}^2 \end{bmatrix}\,, \tag{12}$$

in agreement with Eq. (6), where $q_iq_j = q_{ij}$.

We next observe the essential fact that there are fourteen potential families of singularities in the quaternion mapping that would obstruct the computation of a normalized quaternion, namely any combination of up to three of $q_0 = 0$, $q_1 = 0$, $q_2 = 0$, and $q_3 = 0$. Thus not only do we recover the 180° singularity at $q_0 = 0$ noted by Zhou et al [43], but in fact several additional entire families. Comparing to the training functions proposed by Xiang & Li [37, 38], we see that their form intermixes the singularities, which achieves the desired non-singular result, but lacks our theoretical context. The final conclusion is that one needs to train to the quadratic form in Eq. (6), not the quaternion itself, to solve the pose estimation problem. One must apply a post-network-evaluation procedure on the output 4×4 matrix; one chooses the column of Eq. (6) with the maximum norm, and normalizes that to obtain the desired non-singular quaternion rotation q_{opt} up to a sign. Finally, $R_{\mathrm{opt}} \equiv R(q_{\mathrm{opt}})$ is obtained directly from Eq. (2).

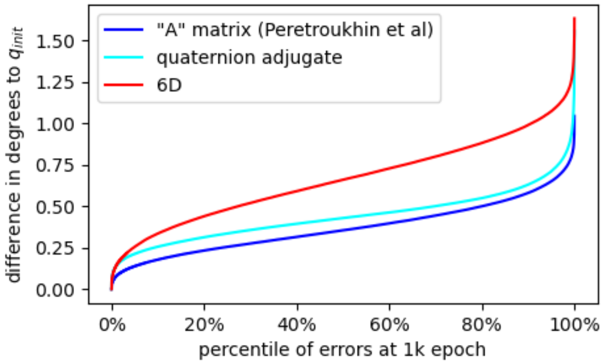

Figure 4. **Training using many rotations of a single source.** Training in the style of [38], with only a single source, shows the quaternion adjugate training framework also works well in this scenario – effectively reducing anomalous max error in the final epoch of testing on 10,000 held out point clouds. Here the mean error in degrees is 0.66, 0.37, and 0.44 for the 6D, A matrix, and quaternion adjugate representations, respectively.

3.3. NN approaches in context of RMSD

We have now established the fundamental mathematical properties of the extraction of a quaternion from rotation data. We conclude this section with a short discussion of how this influences the way we understand neural network approaches to the pose estimation problem. First, we agree with Zhou et al [43] that the rotation itself, in 5D or preferably 6D form, can be appropriately employed as a training target because in fact it is computable from the measured rotation data; however, lone quaternions are not appropriate as machine learning targets in their procedure, and the quaternion subtleties that we have explored explain the issues they uncovered.

Next, we observe that the method of Peretroukhin et al [24] trains to a symmetric 4×4 matrix they denote as "A" and calculates the eigenvector of the minimal eigenvalue of A as the quaternion solution they train to. This is exactly equivalent to the Faugeras & Hebert [6, 7, 13] alternative to the method of Horn (Eq. (4)). The A matrix of Peretroukhin is equivalent to their $B = \sum_{k=1}^{N} B_k$ where k is the index within the point cloud and B_k is defined as follows:

$$B_k = \begin{bmatrix} {a_1}^2 + {a_2}^2 + {a_3}^2 & a_3 s_2 - a_2 s_3 & a_1 s_3 - a_3 s_1 & a_2 s_1 - a_1 s_2 \\ a_3 s_2 - a_2 s_3 & {a_1}^2 + {s_2}^2 + {s_3}^2 & a_1 a_2 - s_1 s_2 & a_1 a_3 - s_1 s_3 \\ a_1 s_3 - a_3 s_1 & a_1 a_2 - s_1 s_2 & {a_2}^2 + {s_1}^2 + {s_3}^2 & a_2 a_3 - s_2 s_3 \\ a_2 s_1 - a_1 s_2 & a_1 a_3 - s_1 s_3 & a_2 a_3 - s_2 s_3 & {a_3}^2 + {s_1}^2 + {s_2}^2 \end{bmatrix}_k \tag{13}$$

where $a_{\{1,2,3\}} = \{x_1 - y_1,\, x_2 - y_2,\, x_3 - y_3\}$ and $s_{\{1,2,3\}} = \{x_1 + y_1,\, x_2 + y_2,\, x_3 + y_3\}$.

Of additional interest is the work of Xiang & Li [38, 37] who propose a set of four different training functions, configured to exclude specific neighborhoods of zeroes of q_0, q_1, q_2, and q_3. This *ad hoc* approach, which first explored 2-fold and 3-fold quaternion representations before landing on the successful 4-fold representation, implements an escape from the discontinuous behavior in the work of [43] only if they alternate among *all four* options for the quaternion training function. In this approach, a 'self-selecting' neural network classifier is used to choose between the various four-fold representations, with sixteen conditions established to select one of the four options for the quaternion training function, concealing the simple relationship between the different unnormalized quaternions that are revealed clearly and unambiguously by the adjugate matrix. In contrast, we use rigorous linear algebra to discover a complete understanding of this multiplicity and the selection process via the adjugate matrix. Thus we have traced the recent advances in quaternion-based neural network approaches to the pose estimation problem to their roots in the RMSD literature, ending with our novel description of the quaternion adjugate approach to resolving the singularities inherent in learning on the quaternion manifold.

4. Training to the Quaternion Adjugate

We now show the practical consequences of applying machine learning with the four-fold unnormalized adjugate matrix. Note that this consists of equivalent, but scaled, copies of the optimal quaternion eigenvector. The code used to conduct these experiments is available at `https://github.com/flatironinstitute/PointCloud_Regression`.

4.1. Simulated random point clouds

For our first experiment, we employ an elementary data model of random 3D point clouds and take advantage of the open source PyTorch implementation of a simplified PointNet architecture [25] used in [24]. For each convolutional layer in the feature extractor network, we use batch normalization without dropout. We used the Adam optimizer and a learning rate of 0.001. For the simulated data, an initial reference cloud of N random (x, y, z) values is transformed by a random quaternion rotation q_{init} plus Gaussian noise σ to generate a test cloud. Our initial experiments were carried out with $N = 100$ points in range ± 1 and $\sigma = 0.1$. We created 1000 reference clouds, each with a matching noisy rotated target partner: 900 pairs were used for training and 100 for testing for each batch. The loss function used for the quaternion-based learning was the chordal distance between the learned and the target quaternion [24]. To learn using the adjugate matrix instead of a lone quaternion we used the initial framework of [24] in which the 10-dimensional output is used to construct a 4×4 symmetric matrix. However, we used the Fröbenius norm as a loss function between the

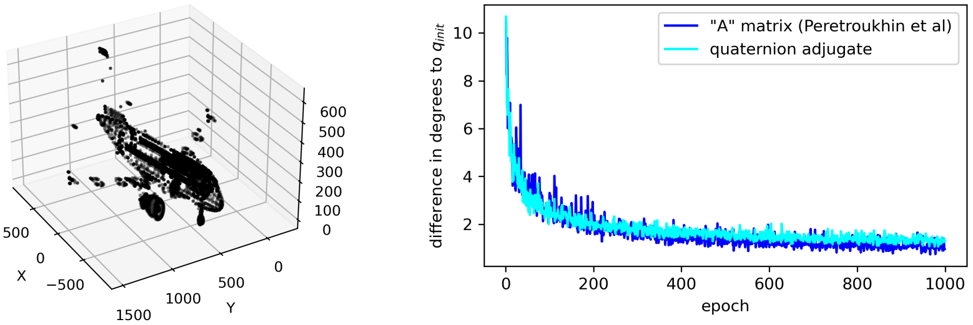

Figure 5. Training on multiple rotations for a single ModelNet airplane reveals similar performance for the "A" matrix and the quaternion adjugate.

learned adjugate matrix and the target adjugate matrix,

$$\text{loss}(\text{Adj}_{\text{target}}, \text{Adj}_{\text{learned}}) = \sum_{k=1}^{K} \left(\sum_{i \leq j} \left({\text{Adj}_{\text{target}}}_{ij}^{\ k} - {\text{Adj}_{\text{learned}}}_{ij}^{\ k} \right)^2 \right) . \tag{14}$$

After learning the best adjugate matrix using this loss, the quaternion result is extracted from this by normalizing the adjugate element with the maximal norm. Note that using the Fröbenius norm between the target and learned adjugate as the loss has the beneficial side-effect of *guaranteeing a good approximation to a rank 1 adjugate matrix*. This is essential since we must extract a column with non-singular normalization from the adjugate, and these will match, up to a possibly-vanishing scale, only if we have a rank 1 matrix.

Using the adjugate method instead of the lone quaternion to train our neural net, we saw learning improve drastically, as shown in Fig. (3), to the point where the mean distance to the target quaternion was closer than for the 6D method of Zhou et al [43] and on par with the "A" matrix method of Peretroukhin et al [24].

In the final experiment with simulated point cloud data, we tested the quaternion adjugate learning performance in a manner that replicates the procedure of Xiang & Li [38]. In this experiment, we used a single point cloud with 100 points as the source, applied 100,000 random rotations, and added Gaussian noise to generate the target point clouds. Throughout this process, we maintained the same hyperparameters and network configurations used in the initial experiment and similarly used 90% of the data for training. In this scenario we again see that the quaternion adjugate learning framework performs well (Fig. (4)).

4.2. Exploring ModelNet40 Data

We further experimented on the ModelNet40 dataset [36], a meshed CAD model with 40 categories of 3D models. We continued to use the standard PointNet [25] architecture to conduct two kinds of experiments on this real-world dataset with a learning rate of 0.0001. In both cases we compared the results using our quaternion adjugate representation to Peretroukhin's "A" matrix model [24]. Here we show results for σ = 0.1, but experiments conducted with σ = 0.01 and 0.05 produced similar results. For each experiment, in addition to the loss, we also measured the angular difference with respect to q_{init}. We first used a single model from the airplane category, and during each iteration of training, we transformed it with random rotations. In this task, 1000 noisy rotated target airplanes are generated to pair with the single initial model, and 90 percent of them are used for training. Fig. (5) displays a sample of an airplane model and the results of training for a single airplane model: both representations achieved a small angle error of around 1 degree.

For the second experiment, we picked 33 airplane models with sizes ranging from 30,000 to 35,000 points. In each iteration of training, we randomly sampled one model out of 33 and used similar settings to those used in PointNet [25] by uniformly sampling 2048 points from each selected airplane. After each downsampling, we applied random rotations and noise to generate target data. We generated 2000 pairs of point clouds and used 90 percent for training. Fig. (6) shows a selection of these airplanes and compares the performance of the two training targets. The final angular errors, around five degrees, were larger than for those trained with just one model, but clearly show that the quaternion adjugate has improved performance over the "A" matrix model. The larger angular error in this experiment is likely due to the ModelNet models being generated from real-world scans from the SUN database [39], which have much greater variation in their shape and size even in a single category. Thus to achieve better performance, more complicated feature extractors, such as self-attention [40] or transformer-based [35] architectures, appear to be needed to learn local and global context information of the

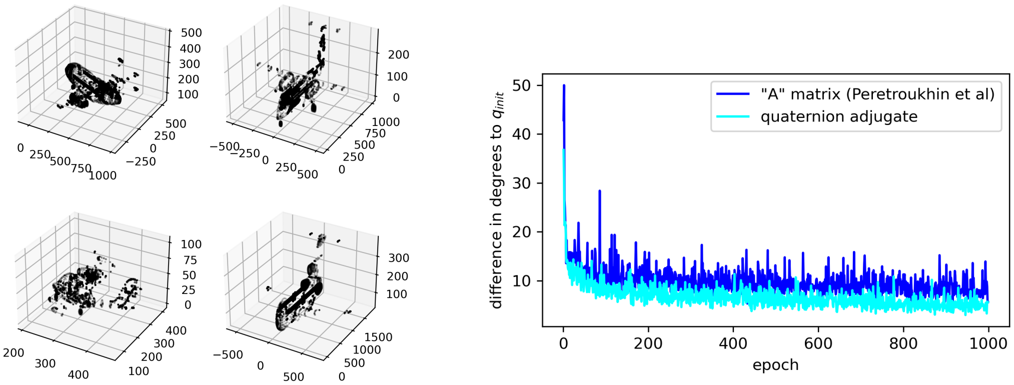

Figure 6. Training on the more complex case of multiple models from ModelNet reveals superior performance of the quaternion adjugate.

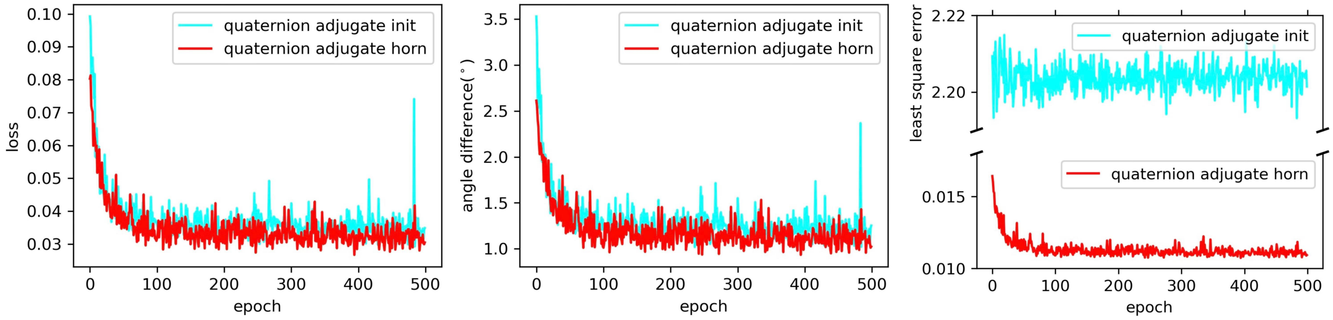

Figure 7. Training on the simulated point cloud dataset with $\sigma = 0.1$ using q_{horn} or q_{init} to produce the target adjugate matrix.

point clouds. We intend to explore this further in the future.

5. Issues to be considered

The tasks and frameworks we have considered here have many nuances that need to be taken into account depending on the context. For example, in the bulk of the literature training is done to the initial simulation-defining rotation, R_{init}, or quaternion, q_{init}. However, the optimal quaternion solution, q_{horn}, (using the RMSD method of Horn and others) remains the correct optimal rotation with or without added noise, and as shown in Fig. (7) this is the more reliable training target.

6. Conclusions

In this paper, we have addressed how the decades old literature solving the quaternion-based 3D point-cloud alignment problem is relevant to learning 3D rotations via neural networks. We have taken the adjugate quaternion framework, originally developed in the context of an analytical solution to the 3D to 2D rotation estimation problem [11], and found that this approach improves the performance of machine learning for this class of problems. We have pointed out that the closed form algebraic (RMSD) solution for the point-cloud alignment problem gives the ideal result, and we encourage others to take advantage of this method in appropriate stages of their analysis. We hope to emphasize with this paper that it is not quaternions that are a deficient representation of rotations, but rather that neural networks with single-function outputs are a deficient representation for multi-valued objects like quaternions, a topic which has been addressed in other problem areas [31, 42]. We note that there are many more subtleties to be understood and explored concerning the use of neural networks for rotation estimation problems. Further work in fact needs to be done even in the limited domain we have explored here, for example, directly incorporating SVD into neural networks [22]. Finally, we would like to highlight the interplay in this paper between standard linear algebra tools and improved development of neural networks, as the most significant results described here lie on both sides of this divide.

References

[1] M. Abramowitz and I. Stegun. *Handbook of mathematical functions*. Dover Publications Inc., New York, 1970. Pages 17–18. 4

[2] Itzhack Y. Bar-Itzhack. New method for extracting the quaternion from a rotation matrix. *Journal of Guidance, Control, and Dynamics*, 23(6):1085–1087, 2000. 2, 4, 5

[3] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization techniques for 3d slam: A survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4597–4604. IEEE, 2015. 2

[4] P.B. Davenport. A vector approach to the algebra of rotations with applications. Technical Report TN D-4696, NASA: Goddard Space Flight Center, Greenbelt, Maryland, 1968. 3

[5] R. Diamond. A note on the rotational superposition problem. *Acta Crystallogr.*, A44:211–216, 1988. 3

[6] Olivier Faugeras and Martial Hebert. A 3D recognition and positioning algorithm using geometrical constraints between primitive surfaces. In *Proc. 8th Joint Conf. on Artificial Intell.*, IJCAI'83, pages 996–1002. Morgan Kaufmann, August 1983. 1, 3, 6

[7] Olivier Faugeras and Martial Hebert. The representation, recognition, and locating of 3D objects. *International Journal of Robotic Research (IJRR)*, 5:27–52, 09 1986. 3, 6

[8] J. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 1

[9] Andrew J. Hanson. *Visualizing Quaternions*. Morgan-Kaufmann/Elsevier, 2006. 2

[10] Andrew J. Hanson. The quaternion-based spatial-coordinate and orientation-frame alignment problems. *Acta Crystallographica Section A*, 76(4):432–457, Jul 2020. 1, 3

[11] Andrew J. Hanson and Sonya M. Hanson. Exploring the adjugate matrix approach to quaternion pose extraction. *arXiv:2205.09116*, 2022. 2, 8

[12] Andrew J. Hanson and Sidharth Thakur. Quaternion maps of global protein structure. *Jour. Molec. Graphics and Modelling*, 38:256–278, Sept 2012. 2

[13] Martial Hebert. Reconnaissance de formes tridimensionnelles, 1983. Ph. D. thesis, University of Paris South. Available as INRIA Tech. Rep. ISBN 2-7261-0379-0. 3, 6

[14] John Gilbert Hocking and Gail S Young. *Topology*. Courier Corporation, 1988. 2

[15] Berthold K.P. Horn, Hugh M. Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A*, 5(7):1127–1136, July 1988. 1, 4

[16] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4:629–642, April 1987. 1, 2, 3

[17] Markus Horn, Nico Engel, Vasileios Belagiannis, Michael Buchholz, and Klaus Dietmayer. Deepclr: Correspondence-less architecture for deep end-to-end point cloud registration. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020. 2

[18] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. 2

[19] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr.*, A32:922–923, 1976. 1

[20] S. K. Kearsley. On the orthogonal transformation used for structural comparisons. *Acta Crystallogr.*, A45(2):208–210, Feb 1989. 3

[21] Gerald R. Kneller. Superposition of molecular structures using quaternions. *Molecular Simulation*, 7(1–2):113–119, 1991. 3

[22] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 8

[23] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7193–7203, 2020. 2

[24] Valentin Peretroukhin, Matthew Giamou, David M. Rosen, W. Nicholas Greene, Nicholas Roy, and Jonathan Kelly. A Smooth Representation of SO(3) for Deep Rotation Learning with Uncertainty. In *Proceedings of Robotics: Science and Systems (RSS'20)*, Jul. 12–16 2020. 1, 2, 3, 4, 5, 6, 7

[25] Charles R. Qi, Hao Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 5, 6, 7

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[27] A. Saxena, J. Driemeyer, and A. Y. Ng. Learning 3D object orientation from images. In *IEEE Inter. Conf. Robotics and Automation (ICRA '09)*, pages 794–800, 2009. 2, 4

[28] P.H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1– 10, 1966. 1, 4

[29] S. W. Shepperd. Quaternion from rotation matrix. *Journal of Guidance and Control*, 1(3):223–224, 1978. 4

[30] K. Shoemake. Animating rotation with quaternion curves. In *Computer Graphics*, volume 19, pages 245–254, 1985. Proceedings of SIGGRAPH 1985. 2

[31] Kshitij Tayal, Chieh-Hsin Lai, Vipin Kumar, and Ju Sun. Inverse problems, deep learning, and symmetry breaking. *arXiv:2003.09077*, 2020. 8

[32] Douglas Theobald. Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. *Acta Crystallogr.*, A61:478–480, 2005. 4

[33] Nikolas Trawny and Stergios I Roumeliotis. Indirect kalman filter for 3d attitude estimation. *A Tutorial for Quaternion Algebra Multiple Autonomous Robotic Systems Laboratory Technical Report Number 2005-002 , Rev . 57*, 01 2005. 2

[34] Grace Wahba. Problem 65-1, A least squares estimate of spacecraft attitude. *SIAM Review*, 7(3):409, 1965. 3

[35] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019. 2, 7

[36] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1912–1920, 2015. 7

[37] Sitao Xiang. Eliminating topological errors in neural network rotation estimation using self-selecting ensembles. *ACM Trans. Graph.*, 40(4):167:1–167:21, August 2021. 1, 2, 4, 5, 6

[38] Sitao Xiang and Hao Li. Revisiting the continuity of rotation representations in neural networks. *arXiv:2006.06234*, 2020. 1, 2, 4, 5, 6, 7

[39] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119:3–22, 2016. 7

[40] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 107–124. Springer, 2022. 7

[41] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824–11833, 2020. 2

[42] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, December 2000. 8

[43] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5745–5753, 2019. 1, 2, 4, 5, 6, 7