Deep Projective Rotation Estimation through Relative Supervision

Anonymous Author(s)
Affiliation
Address
email

Abstract: Orientation estimation is the core to a variety of vision and robotics tasks such as camera and object pose estimation. Deep learning has offered a way to develop image-based orientation estimators; however, such estimators often require training on a large labeled dataset, which can be time-intensive to collect. In this work, we explore whether self-supervised learning from unlabeled data can be used to alleviate this issue. Specifically, we assume access to estimates of the relative orientation between neighboring poses, such that can be obtained via a local alignment method. While self-supervised learning has been used successfully for translational object keypoints, in this work, we show that naively applying relative supervision to the rotational group $SO(3)$ will often fail to converge due to the non-convexity of the rotational space. To tackle this challenge, we propose a new algorithm for self-supervised orientation estimation which utilizes Modified Rodrigues Parameters to stereographically project the closed manifold of $SO(3)$ to the open manifold of $\mathbb{R}^3$, allowing the optimization to be done in an open Euclidean space. We empirically validate the benefits of the proposed algorithm for rotational averaging problem in two settings: (1) direct optimization on rotation parameters, and (2) optimization of parameters of a convolutional neural network that predicts object orientations from images. In both settings, we demonstrate that our proposed algorithm is able to converge to a consistent relative orientation frame much faster than algorithms that purely operate in the $SO(3)$ space. Additional information can be found on our anonymized website.

1 Introduction

Pose estimation is a critical component for a wide variety of computer vision and robotic tasks. It is a common primitive for grasping, manipulation, and planning tasks. For motion planning and control, estimating an object's pose can help a robot avoid collisions or plan how to use the object for a given task. The current top performing methods for pose estimation use machine learning to estimate the object's pose from an image; however, training these estimators tends to rely on direct supervision of the object orientation [1, 2, 3]. Obtaining such supervision can be difficult and requires either time-consuming annotations or synthetic data, which might differ from the real world. In this work, we explore whether self-supervised learning can be used to alleviate this issue by training an object orientation estimator from unlabeled data. Specifically, we assume that we can estimate the relative rotation of an object between neighboring object poses in a self-supervised manner. Such relative supervision can be easily obtained in practice, for example through a local registration method such as ICP [4] or camera pose estimation.

Relative self-supervision has been previously used for representation learning in estimating translational keypoints [5, 6, 7]. These methods use only relative supervision to ensure that the keypoints are consistent across views of the object, and do not directly supervise the keypoint locations. In this work, we explore whether such relative self-supervision can similarly be used in estimating object orientations. We show that naively applying such relative supervision to rotations on the $SO(3)$ manifold will often fail to converge. Unlike self-supervised learning of translational keypoints, the rotational averaging problem [8] is inherently non-convex, with many local optima. While there exist global optimization algorithms which jointly optimize all pairs of rotations for this problem [9, 10],

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

they are not easily integrated into the iterative, stochastic gradient descent methods used to train neural network-based pose estimators.

To address this issue, we propose a new algorithm, Iterative Modified Rodrigues Projective Averaging, which uses Modified Rodrigues Parameters to map from the closed manifold of $SO(3)$ to the open space of $\mathbb{R}^3$. In doing so, we obtain faster convergence with a lower likelihood of falling into local optima. Our experiments show that our method converges faster and more consistently than the standard $SO(3)$ optimization and can easily be integrated into a neural network training pipeline. Additionally, in the supplement, we include an intuitive theoretical example describing how, while not all local optima are removed, the dimensionality of a set of problematic configurations is greatly reduced when optimizing using our algorithm, as compared to optimizing in the space of $SO(3)$.

The primary contributions of this work are:

- We propose a new algorithm, Iterative Modified Rodrigues Projective Averaging, which is an iterative method for learning rotation estimation using only relative supervision and can be applied to neural network optimization.
- We empirically investigate the convergence behavior of our algorithm as compared to optimizing on the $SO(3)$ manifold.
- We demonstrate that our algorithm can be used to train a neural network-based pose estimator using only relative supervision.

2 Related Work

Averaging and Consensus Estimation: Consensus methods, sometimes referred to as averaging methods, have a long history of research. The goal of these methods is, given a distributed set of estimates, to produce a consistent prediction of a value using relative information. While there are iterative algorithms with good convergence properties in Euclidean space [11, 12, 13, 14, 15], optimizing over the closed manifold of $SO(3)$ can be more difficult, as the region is non-convex, with many local minima. Hartley et al. [8, 16] describe several methods of finding a consistent set of rotations, though their convergence is similarly not guaranteed outside of a radius $r \leq \frac{\pi}{2}$ ball in $SO(3)$. Wang and Singer [10] find an exact solution to this problem, using a combination of a semidefinite programming relaxation and a robust penalty function. More recently, Shonan Rotation Averaging [9] shows that projecting to higher dimensional spaces allows for the recovery of a globally optimal solution using semidefinite programming. Chatterjee and Govindu [17, 18] use iterative re-weighted least-squares to recover a global optimal solution using global error estimates. Shi and Lerman [19] extends this work, using cycle consistency and message passing. Chen et al. [20] tackle the problem through an hybrid appracoh of obtaining a global solution via semidefinite programming, then refining the solution through iterative $SO(3)$ log space update. These solutions require global error estimates or semidefinate programming, which are incompatible with the stochastic gradient descent methods used to train neural networks. These methods are infeasible for our problem, as they do not well integrate with the SGD training frameworks used for neural networks.

Supervised Orientation Estimation: Past work has explored using a neural network to predict an object's orientation. Traditionally, these methods rely on supervising the rotations using a known absolute orientation, whether in the form of quaternions [21, 1, 22], axis-angle [23], or Euler angles [24]. More recently, 6D [25, 2], 9D [26], and 10D [27] representations have been developed for continuity and smoothness. Recently, Terzakis et al. [28] introduced Modified Rodrigues Parameters, a projection of the unit quaternion sphere $\mathbb{S}^3$ to $\mathbb{R}^3$ used in attitude control [29], to a range of common computer vision problems. Terzakis et al. [28] does not, however, address the unique problems found in the rotation averaging problem.

Some methods, such as DeepIM [30], have posed the rotation estimation problem purely as a relative problem, computing the transform to rotate from one object pose to another. Similarly, $se(3)$-TrackNet [31] tracks object pose using a Lie Algebra-based orientation update. While these methods do remove the need for absolute supervision, the resulting estimates are only useful when compared to an anchor image with an absolute orientation given. In practice, obtaining an absolute pose can be useful for both planning and joint learning of orientation representation and control. For this reason, we seek to estimate an absolute pose using relative supervision.

Recently, there has been research into mapping the Riemannian optimization to the Euclidean optimization used for network training [32, 33, 34, 35, 36]. These methods focus on applying tangent space gradients from losses in 3D transformation groups. Specifically, Projective Manifold Gradient Layer [32] ensures that the gradients take into account any projection operations, such that the gradients point towards the nearest valid representation in the projection's preimage. While this does map the Riemannian optimization into a Euclidean problem, it does not solve the problems caused by the closed manifold of $SO(3)$, as this does not alter the underlying topology of this manifold.

3 Problem Definition

We formally describe the problem of self-supervised orientation estimation below. We assume that we are given a set of inputs observations $\{I_1, \ldots, I_N\}$, of an object where, in each input observation I_i, the object is viewed from an unknown orientation R_i. These inputs could be in the form of images, point clouds, or some other object representation. While we do not know the absolute object orientations R_i in any reference frame, we assume that we do know a subset of the relative rotations R_j^i, possibly from a local registration method like ICP, between the object in images I_j and I_i, such that $R_i = R_i^j R_j$. Our goal is to learn a function $f(I_i)$ that estimates an orientation of the object in each image, $f(I_i) = \hat{R}_i$ that minimizes the pairwise error of all input pairs, with respect to the geodesic distance metric $d(R_i, R_j) = \| \log(R_i^\top R_j)\|^2$. Given a set of rotations $\mathcal{R} = \{R_1, \ldots, R_N\}$, the core optimization objective is thus:

$$\min_{\hat{R}_i, \hat{R}_j \in \mathcal{R}} \sum_{i,j} d(\hat{R}_i, R_i^j \hat{R}_j) \tag{1}$$

Note that this optimization does not have a unique solution, since the solution $\hat{R}_i := SR_i, \forall i$ minimizes this error for any constant rotation S.

In many robotics tasks, relative rotations can be accurately estimated only when their magnitude is small as many registration algorithms, such as ICP, requires a good initialization near the optimum. Following this observation, we assume that we can only accurately supervise relative rotations when they are small in magnitude. This leads to a local neighborhood structure where each rotation R_i is connected to R_j only in a local neighborhood around R_i, when $d(R_i, R_j) < \epsilon$, and the set of all R_j's connected to R_i form the neighborhood set of $\mathcal{N}_i$. While the algorithms described in this manuscript do not rely on this angle ϵ, it can be scaled as needed based on the accuracy of the relative rotation estimation method (e.g. ICP, etc).

Our eventual goal is to represent the function $f(I_i) = \hat{R}_i$ as a neural network. Thus, we restrict the methods with which we compare to iterative methods that are updated using only a sampled subset of the rotations (as opposed to methods that perform a global optimization over the entire set of rotations $\{R_1, \ldots, R_N\}$). This requirement is to match the conditions required by stochastic gradient descent, the primary method of training neural networks.

4 Baselines

Preliminaries. The 3D rotational space of $SO(3) \triangleq \{R \in \mathbb{R}^{3\times3} : R^\top R = \mathbb{I}_{3\times3}, \det(R) = 1\}$ is a compact matrix Lie group, which topologically is a compact manifold. Due to the compactness of the $SO(3)$ manifold, there exist configurations of pairs of points where multiple, non-unique geodesically minimal paths exist between them; for instance, there are two unique geodesically minimal paths for a pair of antipodal points on a circle, and there are infinitely many for a pair of antipodal points on a sphere. This is not the case in an open manifold like the 3D Euclidean space of $\mathbb{R}^3$, over which there exists a unique geodesically minimal path between any arbitrary pair of points. The distinction in compactness between the 3D rotational space of $SO(3)$ and 3D Euclidean space makes optimization over $SO(3)$ more ill-conditioned than over the space of $\mathbb{R}^3$. This results in the optimization over the rotational space being non-convex. These properties of the $SO(3)$ manifold will affect the convergence of self-supervised orientation estimation, which we discuss below.

While self-supervised learning for objects translation, specifically in the form of object keypoints [5, 6, 7], has shown great success, in this work, we show that naively applying such an iterative self-supervised formulation to the rotational group $SO(3)$ will often fail to converge. Below we discuss two approaches to self-supervised orientation estimation in $SO(3)$.

Quaternion Averaging: A standard objective in rotation estimation is to minimize the geodesic distance between a predicted unit quaternion and its corresponding ground-truth orientation [37, 8], $\theta = \arccos(2\langle \hat{q}_i, q_{gt}\rangle^2)$ where $\hat{q}_i$ is the predicted orientation for image i and q_{gt} is the ground-truth orientation. An objective function is often defined to directly minimize this geodesic distance [37].

In our task, defined above (Section 3), we are given the relative rotation q_i^j between some pairs of rotations q_i and q_j. Using this relative supervision, we can use the geodesic distance between a sampled estimate, $\hat{q}_i$, its desired relative position with respect to a sampled neighbor and a known relative rotation q_i^j, $\tilde{q}_i = q_i^j \otimes \hat{q}_j$, leading to the loss $\mathcal{L}_q = 1 - \langle \hat{q}_i, q_i^j \otimes \hat{q}_j\rangle^2$, where $\otimes$ denotes the quaternion multiplication. Note that this loss is monotonically related to the geodesic distance when using unit quaternions, while avoiding the need to compute an $\arccos$.

$SO(3)$ **Averaging:** To optimize the rotations with respect to the non-Euclidean geometry of the rotational manifold of $SO(3)$, one approach is described by Manton [38]. Each orientation is iteratively updated in the tangent space using the logmap of $SO(3)$ and projected back to $SO(3)$ using the exponential map. Specifically, we can take the gradient of the loss

$$\mathcal{L}_{SO(3)} = \left\| \log\left(R_i^\top R_i^j R_j\right) \right\|^2 \qquad (2a) \qquad \nabla_{\hat{r}_i} \mathcal{L}_{SO(3)} = r_\Delta = \log\left(R_i^\top R_i^j R_j\right) \qquad (2b)$$

which gives the update step $\hat{R}_i \leftarrow \hat{R}_i \exp(\gamma r_\Delta)$, where γ is the learning rate and $\log$ is the logmap of $SO(3)$. When optimizing the full set of orientations, this algorithm can fall into local optima due to the closed nature of the space which allows any orientation to be reached by two unique straight paths, as the space wraps around on itself.

5 Method

We propose an alternative that projects the optimization to an open image and optimizes the distances in that space. Specifically, we use the Modified Rodriguez Projection to minimize the relative error between neighboring poses in $\mathbb{R}^3$. We provide experiments in Section 6 that show that self-supervised orientation estimation using Modified Rodriguez Projection converges much faster than self-supervised orientation estimation in $SO(3)$, with theoretic analysis of an illustrative example available in the supplement.

5.1 Iterative Modified Rodrigues Projective Averaging

As mentioned previously, optimizing on a closed space, such as $SO(3)$ or $\mathbb{S}^3$ can be problematic, since the relative distance between two points can eventually be minimized by moving them in the exact opposite direction of the minimum path between them. To alleviate this issue, we would like to instead perform self-supervised learning in an open space, where this symmetry is broken. This can be done using Modified Rodrigues Parameters (MRP) [39, 28]. MRP is the stereographic projection of the closed manifold of the quaternion sphere $\mathbb{S}^3$ to $\mathbb{R}^3$, and has been widely used in attitude estimation and control [29]. In combining this projection with the mapping between $SO(3)$ and $\mathbb{S}^3$, this projection can be used to optimize rotations.

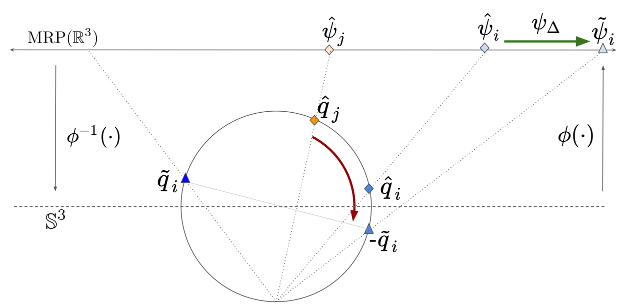


Figure 1: Projection of relative supervision, q_i^j, shown in red, from back-projected rotation $\hat{q}_j := \phi^{-1}(\hat{\psi}_j)$ to $\hat{q}_j$ into the MRP space update, ϕ_Δ, shown in green. While $\tilde{q}_i$ could have been selected as the the goal rotation, it would have induced a much larger movement in the projected space.

We define a unit quaternion $q = [\rho \quad \nu] \in \mathbb{S}^3 \triangleq \{x \in \mathbb{R}^4 : \|x\| = 1\}$, where $\rho \in \mathbb{R}$ defines the scalar component and $\nu \in \mathbb{R}^3$ defines the imaginary vector component of the unit quaternion. The projection operator $\phi(q) = \psi \in \mathbb{R}^3$ and its inverse $\phi^{-1}(\psi) = q \in \mathbb{S}^3$ are given by [39, 28] where $\psi = \phi([\rho \quad \nu]) = \frac{\nu}{1+\rho}$ and $[\rho \quad \nu] = \phi^{-1}(\psi) = \left[\frac{1-\|\psi\|^2}{1+\|\psi\|^2} \quad \frac{2\psi}{1+\|\psi\|^2}\right]$. Given this projective orientation space, we need to map our relative rotation R_i^j into the projective space in order to use these relative rotations for the self-supervised learning task. This projection is required, as the relative supervision is in $SO(3)$, and the direction and magnitude of this relative measurement are distorted differently in different regions of the projective MRP space. Given a pair of estimated projected rotations $\hat{\psi}_i := \phi(\hat{R}_i)$ and $\hat{\psi}_j := \phi(\hat{R}_j)$, we project $\hat{\psi}_j$ back to a unit quaternion $\phi^{-1}(\hat{\psi}_j) = \hat{q}_j \in \mathbb{S}^3$ and rotate it according to R_i^j, $\tilde{q}_i = q_i^j \otimes \hat{q}_j$, where $\otimes$ is quaternion multiplication and q_i^j is the quaternion form of R_i^j. The resulting unit quaternion $\tilde{q}_i$ is then projected back into the Modified Rodrigues Parameter space, $\tilde{\psi}_i$. A simplified visual analogy of this process is shown in Figure 1.

While this relative rotation could be applied and projected at either the sampled point $\hat{\psi}_i$, or the neighboring location $\hat{\psi}_j$, we select the neighboring location $\hat{\psi}_j$, as it does not require us to compute gradients through the forward or inverse projections $\phi(\cdot)$ and $\phi^{-1}(\cdot)$, respectively. This projected rotation $\tilde{\psi}_i$ represents the value $\hat{\psi}_i$ should hold, relative to the current predicted rotation $\hat{\psi}_j$. It should be noted that $\psi(q) \neq \psi(-q)$, while q and $-q$ represent the same rotation. In terms of the projective space, this means that the sign of $\tilde{q}_i$ matters. To remove this ambiguity, we select the nearest projection to $\hat{\psi}_i$ in the projective MRP space. It should be noted that this is different from selecting the closer antipode on $\mathbb{S}^3$, as the large deformations found near the south pole[1] can cause the nearer antipode in $\mathbf{S}^3$ to be further in MRP space. In contrast, if we were to select a consistent sign for the scalar component $\tilde{q}_i$, for example ensuring the scalar component is always positive, a small change in $\hat{\psi}_j$ can cause large changes in $\tilde{\psi}_i$. While this change is required to stabilize our optimization, it does add some ambiguity to the direction of optimization. However, the directions to each of the projected locations, $\psi(\tilde{q}_i)$ and $\psi(-\tilde{q}_i)$, are only anti-parallel (pulling in exactly opposite directions) when $\tilde{\psi}_i - \hat{\psi}_i$ intersects the origin.

The loss with respect to a given estimate, $\hat{\psi}_i$, can then be written as the l_2 distance between its current value and the projected relative location, $\tilde{\psi}_i$, relative to a given neighbor, $\hat{\psi}_j$:

$$\mathcal{L}_{\Psi+} = \left\| \hat{\psi}_i - \phi(\tilde{q}_i) \right\|^2 \quad (3a) \quad \mathcal{L}_{\Psi-} = \left\| \hat{\psi}_i - \phi(-\tilde{q}_i) \right\|^2 \quad (3b) \quad \mathcal{L}_{\Psi} = \min(\mathcal{L}_{\Psi-}, \mathcal{L}_{\Psi+}) \quad (3c)$$

where we recall that, $\tilde{q}_i = q_i^j \otimes \hat{q}_j$, and $\hat{q}_j = \phi^{-1}(\hat{\psi}_j)$.

Note that, while $\hat{\psi}_j$ is a predicted value, we do not pass gradients through it, allowing it to anchor the update to a consistent orientation. The gradient update[2] is then given by:

$$\nabla_{\hat{\psi}_i} \mathcal{L}_{\Psi} = \psi_{\Delta} = \begin{cases} \hat{\psi}_i - \phi\left(\tilde{q}_i\right), & \text{if } \mathcal{L}_{\Psi+} < \mathcal{L}_{\Psi-} \\ \hat{\psi}_i - \phi\left(-\tilde{q}_i\right), & \text{otherwise} \end{cases} \tag{4}$$

Additionally, a maximum gradient step, η, in the projective space is imposed, $\psi_{\Delta} \leftarrow \eta \frac{\psi_{\Delta}}{\|\psi_{\Delta}\|}$, if the gradient exceeds a defined amount.This prevents extremely large steps from being taken, as the projective transform can distort the space.

6 Experiments

Next, we perform experiments to show that our method converges faster and more consistently than the alternative approaches. Our empirical results are grouped into two settings: (1) direct optimization of randomly generated rotations, Section 6.1, and (2) optimization of the parameters of a convolutional neural network using synthetically rendered images, Section 6.2. In both cases, relative orientations between elements in a neighborhood are provided. We show Iterative Modified Rodrigues Projective Averaging is able to converge faster and more often than alternative approaches. We further show in Section 6.2 that our method can easily be used to supervise convolutional neural networks, when only relative orientation information is available.

[1]The south pole in this case is described by the quaternion $-1 + 0i + 0j + 0k$

[2]We omit a constant factor for brevity, and integrate it into the learning rate, γ.

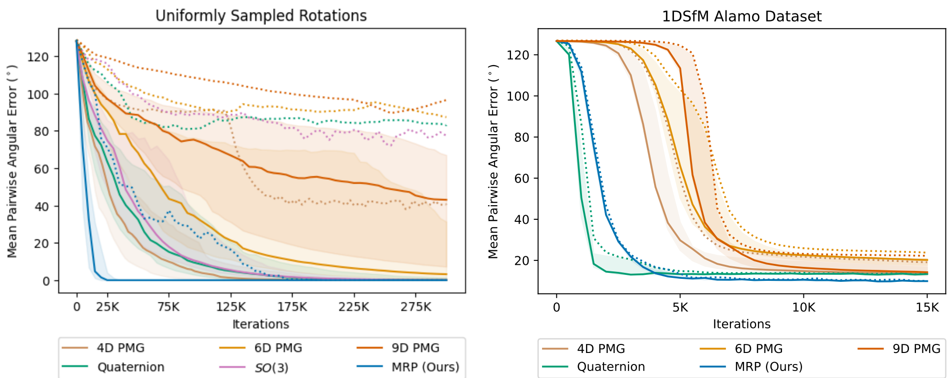


Figure 2: Relative rotation consensus with direct optimization of rotation parameters over 50 unique environments with 100 random generated orientations each (**left**) and Alamo 1DSfM [40] (**right**). Median average-pair-wise angular error (°) between each estimated rotations is shown, with shaded region representing the first and third quartile for each method. The max average-pair-wise angular error for each algorithm at each iteration is shown as a dashed line.

	Avg Pairwise Angular Error $< 5°$			*Normalized AUC*		
Algorithm	Mean Steps	Max Steps	Min Steps	Mean	Max	Min
$SO(3)$	157.7K	Not Converged	85.0K	24.47	82.92	7.55
4D PMG [32]	126.1K	Not Converged	27.0K	15.67	52.40	3.06
6D PMG [25]	235.9K	Not Converged	80.0K	43.53	89.15	11.34
9D PMG [26]	284.5K	Not Converged	150.0K	62.94	101.77	17.77
Quaternion	160.3K	Not Converged	40.0K	23.55	84.85	3.47
MRP (Ours)	**37.5K**	**160.0K**	**15.0K**	**5.08**	**15.56**	**2.18**

Table 1: Number of iteration steps until convergence and Normalized Area Under Curve (nAUC) over 50 unique environments of 100 randomly generated orientations. 300K optimization steps are taken for each experiment.

6.1 Direct Parameter Optimization

We evaluate the convergence behaviour of our Iterative Modified Rodrigues Projective Averaging method, **MRP (Ours)** , described in Section 5.1, as well as the $SO(3)$ averaging method, described in Section 4. For the $SO(3)$ averaging method, we implement both the pure Riemannian optimization , $\mathbf{SO(3)}$, as well as a method using a Projective Manifold Gradient Layer [32] to map the Riemannian gradient of the $SO(3)$ averaging loss, Equation 2a, to a Euclidean optimization in $\mathbb{R}^D$, where we test $D = 4$, $D = 6$ [25], and $D = 9$ [26], **4D PMG [32]**, **6D PMG [25]**, **9D PMG [26]**, respectively. Additionally, we evaluate direct quaternion optimization, described in Sections 4, **Quaternion**.

Uniformly Sampled Rotations. We test the performance of each algorithm when directly optimizing the rotation parameters of a set of size $N = 100$ with known relative rotations R_i^j, and local neighborhood structure. Ground truth and initial estimated rotations are both randomly sampled from a uniform distribution in $SO(3)$. Each rotation, R_i, has a neighborhood, $\mathcal{N}_i$, consisting of the closest $|\mathcal{N}_i| = 3$ rotations with respect to geodesic distance. The connectivity of this neighborhood graph is checked to ensure the graph contains only a single connected component. We test all algorithms over 50 sets of unique environments, each with $N = 100$ randomly generated orientations as described above. The estimated rotations are updated by each algorithm in batches of size 8, for 300K iterations.

As the goal of our algorithm is to improve the convergence properties of iterative averaging methods, we analyze each algorithm at various stages of optimization. We are particularly interested in the average number of update steps until the algorithm has converged, which we define as when the average angular error between all pairs of rotations is below $5°$. As we can see in Figure 2, the Iterative Modified Rodrigues Projective Averaging method, **MRP (Ours)**, converges before the standard $SO(3)$ averaging method. On average, our method converged to within $5°$ in 37K steps. The next best method, **4D PMG [32]**, which takes over three times as many iterations to converge to the same

Algorithm	% Avg Pairwise Angular Error $< 5°$					Final Error(°)	
	30K	70K	100K	150K	300K	Mean	Median
$SO(3)$	0%	0%	6%	57%	94%	2.056	0.10
4D PMG [32]	2%	32%	46%	72%	90%	1.969	0.14
6D PMG [25]	0%	0%	4%	20%	52%	20.096	3.20
9D PMG [26]	0%	0%	0%	2%	20%	40.125	43.02
Quaternion	0%	12%	30%	56%	82%	9.72	0.04
MRP (Ours)	**66%**	**88%**	**96%**	**98%**	**100%**	**0.004**	**0.004**

Table 2: Percentage of experiments converged and final angular errors over 50 unique environments of 100 randomly generated orientations. 300K optimization steps are taken for each experiment.

level of accuracy. Further, Table 1 shows that our method is the only one to converge across all environments within 300K iterations. For each method, we also compute the mean area under the pairwise error curve, with the number of steps normalized to between zero and one (nAUC), also shown in Table 1. We find that in the best, average, and worst case scenarios, our method has the best convergence behavior. To quantify convergence behavior, we also compute the percentage of trials that achieve average pairwise angular error below 5° at different stages of training, as shown on the left in Table 2. We find that at each stage of training, the Iterative Modified Rodrigues Projective Averaging, **MRP (Ours)**, training has a lower average pairwise error, shown in Table 2. Our method also converged far more often at each stage of training, also shown in Table 2.

Algorithm	Mean Relative Error (°)		Mean Absolute Error (°)		Mean nAUC	
	E. Island	Alamo	E. Island	Alamo	E. Island	Alamo
4D PGM [32]	11.94	15.00	7.34	9.94	25.60	47.20
6D PGM [25]	11.26	18.84	6.90	13.09	27.77	58.04
9D PGM [26]	10.22	16.32	6.32	11.43	29.31	60.14
Quaternion	11.58	13.40	7.23	8.93	**16.01**	**22.57**
MRP (Ours)	**8.84**	**9.89**	**5.49**	**6.56**	16.21	25.61
IRLS-GM[17]	-	-	3.04	3.64	-	-
IRLS-$\ell_{\frac{1}{2}}$ [18]	-	-	2.71	3.67	-	-
MLP[19]	-	-	2.61	3.44	-	-

Table 3: **Rotation Averaging Results on 1DSfM [40] dataset.** Results before the double lines are comparisons of local method by mean relative error (°), mean absolute error (°) and normalized area under curve (nAUC) after 20K iterations. Results under the double lines are obtained from global methods which require optimizing over global set of relative orientations data at each step. Results for sections with dashed line are not available from global methods [19].

Structure from Motion Dataset. To test our algorithms under natural noise conditions, we also evaluate our algorithm on the 1DSfM [40] structure from motions datasets. These datasets contain full transforms for each sample; however, we are only concerned with optimizing the rotations. Each environment is tested with 5 random initializations and the estimated rotations are updated by each algorithm in batches of size 64, for 20K iterations. The results of a subset of the environments are shown in Table 3 and the remainder can be found in the supplement. The noise characteristics of relative rotations in this dataset are similar to those found when capturing relative poses, but, unlike the environments found in the previous section (Uniformly Sampled Rotations), the distribution of rotations does not fully cover the orientation space. As a result, all methods converge relatively quickly. Our algorithm outperforms the baselines in terms of accuracy. While the **Quaternion** optimization converges slightly faster, it consistently finds a lower accuracy configuration, resulting in a low nAUC, but higher relative and absolute accuracy. More details can be found in the supplemental.

6.2 Neural Network Optimization

To show that the Iterative Modified Rodrigues Projective Averaging method, **MRP (Ours)**, can be used to learn orientation using neural networks by optimizing the parameters of a simple CNN, specifically a ResNet18 [41], we follow the procedure as in Section 6.1 with some minor changes. Instead of operating directly on a set of rotation parameters, we learn a function $\hat{\psi}_i = f(I_i)$ from

Algorithm	Mean Error (°)	Median Error (°)	5° Acc (%)
4D PMG [32]	123.84	123.96	0
Quaternion	28.83	21.74	50
MRP (Ours)	**3.71**	**3.73**	**100**
Oracle	1.58	1.56	100

Table 4: **Final results for image based rotation estimation.** Final mean and median angular error (°) after 10K steps over 8 unique environments of 100 images associated with randomly generated orientations are shown. Percentage of runs converged below 5° angular error is also showed at 10K steps.

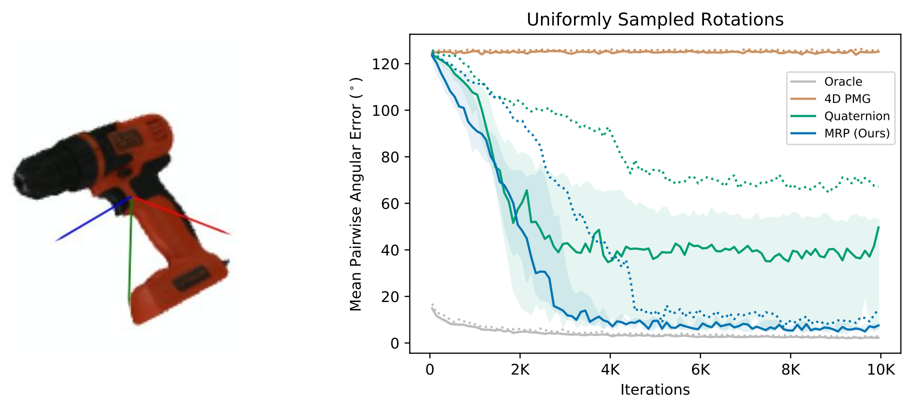


Figure 3: Estimated rotation frame learned for the YCB [1] drill model using Iterative Modified Rodrigues Projective Averaging and relative rotations (x, y, z) (left). Results for rotations estimated by neural networks given images of the YCB drill [1] rendered at each of 100 random rotations with various supervisions, (right). Median average-pairwise angular error (°) is shown with shaded areas representing the first and third quartile over all training sessions. The max average-pairwise angular error for each algorithm at each iteration is shown as a dashed line.

rendered images of the YCB drill [1] model, shown in Figure 3, rendered at each of 100 random orientations R_i. We continue to only supervise each method described in Section 6.1 using the relative rotations between each image. We compare the best performing methods, and, as a lower bound, we also train an oracle network, **Oracle**, with the ground truth rotations, R_i and cosine quaternion loss. We use the Adam [42] optimizer, batch size of 32 and learning rate of 1×10^{-4} for all experiments, and with a maximum training time of 10K steps, trained over 8 environments, each with 100 images associated with randomly generated rotations. We report final mean and median pairwise angular error, and the percentage of runs converged below 5° pairwise angular error as 5° Acc. We find that **MRP (Ours)** is able to converge to a rotational frame consistent with the relative rotations used for supervision relatively quickly, with a significantly lower average-pairwise-error than all other relative methods, shown in Figure 3 and Table 4.

We also perform experiments on generalization to unseen poses and find that a curriculum is required (see supplement for details). For the generalization experiments, we found that **MRP (Ours)** achieves a mean pairwise angular error or $5.19°$, **Quaternion** achieves $12.41°$, and **4D PMG [32]** never converged, with final error of $125.09°$.

7 Limitations

While this parameterization of the rotational space is valuable for learning rotations using only relative supervision, it is not without limitations. One of the primary ones is the need for a curriculum for generalizability to unseen relative rotations. Without this, our experiment show that all representations fall into the local optima of outputting a constant orientation. Additionally, in generalization experiments, we are only able to achieve a final error of 5 degrees. This may not be accurate enough for many fine motor tasks, though an additional refinement network that is trained to handle rotations within a sub-region of the whole rotation space could reduce this error.

8 Conclusion

In this paper, we show that through the use of Modified Rodrigues Parameters, we are able to open the closed manifold of $SO(3)$, improving the convergence behavior of the rotation averaging problem. We show that Iterative Modified Rodrigues Projective Averaging is able to outperform the naive application of relative-orientation supervision in both direct parameter optimization and image-based rotations estimation from neural networks. We hope our method allows more systems to convert the relative supervision of relative methods, like ICP, to consistent and accurate absolute poses.

References

[1] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018.

[2] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020.

[3] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. 2019.

[4] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *European conference on computer vision*, pages 766–782. Springer, 2016.

[5] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. *International Symposium on Robotics Research (ISRR) 2019*, 2019.

[6] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In *ECCV*, 2018.

[7] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *NeurIPS*, 2018.

[8] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013.

[9] F. Dellaert, D. M. Rosen, J. Wu, R. Mahony, and L. Carlone. Shonan rotation averaging: Global optimality by surfing $so(p)^n$. In *European Conference on Computer Vision*, pages 292–308. Springer, 2020.

[10] L. Wang and A. Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2(2):145–193, 2013.

[11] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69 (345):118–121, 1974.

[12] S. Chatterjee and E. Seneta. Towards consensus: Some convergence theorems on repeated averaging. *Journal of Applied Probability*, 14(1):89–97, 1977.

[13] A. Olshevsky and J. N. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM journal on control and optimization*, 48(1):33–55, 2009.

[14] W. J. Russell, D. J. Klein, and J. P. Hespanha. Optimal estimation on the graph cycle space. *IEEE Transactions on Signal Processing*, 59(6):2834–2846, 2011.

[15] A. Beck and S. Sabach. Weiszfeld's method: Old and new results. *Journal of Optimization Theory and Applications*, 164(1):1–40, 2015.

[16] R. Hartley, K. Aftab, and J. Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *CVPR 2011*, pages 3041–3048. IEEE, 2011.

[17] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 521–528, 2013.

[18] A. Chatterjee and V. M. Govindu. Robust relative rotation averaging. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):958–972, 2017.

[19] Y. Shi and G. Lerman. Message passing least squares framework and its application to rotation synchronization. In *International Conference on Machine Learning*, pages 8796–8806. PMLR, 2020.

[20] Y. Chen, J. Zhao, and L. Kneip. Hybrid rotation averaging: A fast and robust rotation averaging approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10358–10367, 2021.

[21] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[22] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5974–5983, 2017.

[23] T.-T. Do, M. Cai, T. Pham, and I. Reid. Deep-6dpose: Recovering 6d object pose from a single rgb image. *arXiv preprint arXiv:1802.10367*, 2018.

[24] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. 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*, pages 2686–2694, 2015.

[25] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[26] J. Levinson, C. Esteves, K. Chen, N. Snavely, A. Kanazawa, A. Rostamizadeh, and A. Makadia. An analysis of svd for deep rotation estimation. *Advances in Neural Information Processing Systems*, 33:22554–22565, 2020.

[27] V. Peretroukhin, M. Giamou, D. M. Rosen, W. N. Greene, N. Roy, and J. Kelly. A smooth representation of belief over so(3) for deep rotation learning with uncertainty. *In Proceedings of Robotics: Science and Systems (RSS'20)*, 2020.

[28] G. Terzakis, M. Lourakis, and D. Ait-Boudaoud. Modified rodrigues parameters: an efficient representation of orientation in 3d vision and graphics. *Journal of Mathematical Imaging and Vision*, 60(3):422–442, 2018.

[29] J. Crassidis and F. Markley. Attitude estimation using modified rodrigues parameters. In *Flight Mechanics/Estimation Theory Symposium*, pages 71–86. NASA, 1996.

[30] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.

[31] B. Wen, C. Mitash, B. Ren, and K. E. Bekris. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10367–10373. IEEE, 2020.

[32] J. Chen, Y. Yin, T. Birdal, B. Chen, L. Guibas, and H. Wang. Projective manifold gradient layer for deep rotation regression. *arXiv preprint arXiv:2110.11657*, 2021.

[33] Z. Teed and J. Deng. Tangent space backpropagation for 3d transformation groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10338–10347, 2021.

[34] R. Brégier. Deep regression on manifolds: a 3d rotation case study. In *2021 International Conference on 3D Vision (3DV)*, pages 166–174. IEEE, 2021.

[35] R. Arora. On learning rotations. *Advances in neural information processing systems*, 22:55–63, 2009.

[36] S. Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[37] S. Mahendran, H. Ali, and R. Vidal. 3d pose regression using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2174–2182, 2017.

[38] J. H. Manton. A globally convergent numerical algorithm for computing the centre of mass on compact lie groups. In *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference, 2004.*, volume 3, pages 2211–2216. IEEE, 2004.

[39] T. F. Wiener. *Theoretical analysis of gimballess inertial reference equipment using delta-modulated instruments*. PhD thesis, Massachusetts Institute of Technology, 1962.

[40] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[41] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[42] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

Deep Projective Rotation Estimation through Relative Supervision - Supplement

Anonymous Author(s)
Affiliation
Address
email

A Intuitive Example

We present an intuitive example of when optimizing a set of orientations to solve the rotation averaging problem described in Equation (1), in the main text, can fail. In this example, we show the benefits of the Iterative Modified Rodrigues Projective Averaging approach over the baseline approach. We show that, while both $SO(3)$ averaging and Iterative Modified Rodrigues Projective Averaging share a class of non-optimal critical points, in the projective case, these critical points are a subset of the problematic configurations for $SO(3)$ averaging.

A.1 Examples of Critical Points

In this section, we analyze a class of critical points shared by both standard $SO(3)$ averaging and Iterative Modified Rodrigues Projective Averaging. For simplicity, we will examine the $N = 3$ rotation case, where $\mathcal{R} = \{R_1, R_2, R_3\}$ with relative rotations of $R_i^j := R_i R_j^\top$. As this is an iterative algorithm, we need to initialize our predicted rotations to some values $\hat{\mathcal{R}} = \{\hat{R}_1, \hat{R}_2, \hat{R}_3\}$. In this case, we initialize each predictions to $\hat{R}_i := R_i R_0 \exp\left(\left(\theta_0 + i\frac{2\pi}{N}\right)\omega_0\right)$ where R_0 is an arbitrary but constant rotational offset, ω_0 and θ_0 define an arbitrary, but constant axis and constant rotation, about which each initial estimate $\hat{R}_i$ is rotated an additional angle of θ_i. We find that if we use the previously described methods to update this initial configuration, under certain values of $\mathcal{R}$, R_0, θ_0, and ω_0, the expected update at each value $\hat{R}_i$ is $\mathbf{0}$, forming a critical point for each algorithm.

A.1.1 Critical Point for $SO(3)$ Averaging

Given the initial predictions of $\hat{\mathcal{R}}$ defined above, for all values of $\mathcal{R}$, R_0, θ_0, and ω_0, we find that the expectation of the gradient of $SO(3)$ averaging loss, $\mathbb{E}_{i,j}\left[\nabla_{\hat{r}_i}\mathcal{L}_{SO(3)}\right]$, is $\mathbf{0}$. The gradient of any sampled pair i, j is given by

$$\begin{aligned}
\nabla_i \mathcal{L}_{SO(3)}^{i,j} := & \nabla_{\hat{r}_i}\mathcal{L}_{SO(3)}\left(\hat{R}_i, \hat{R}_j, R_i^j\right) \\
&= \log\left(\hat{R}_i^\top R_i^j \hat{R}_j\right) \\
&= \log\left(\left(R_i R_0 \exp\left(\theta_i\omega_0\right)\right)^\top R_i^j R_j R_0 \exp\left(\theta_j \omega_0\right)\right) \\
&= \log\left(\exp\left((\theta_j - \theta_i)\omega_0\right)\right) \\
&= \text{wrap}_{[-\pi,\pi)}\left[(\theta_j - \theta_i)\right]\omega_0 \\
&= \text{wrap}_{[-\pi,\pi)}\left[\frac{2\pi}{N}(j-i)\right]\omega_0 \\
&= \frac{2\pi}{N}(j-i)\omega_0.
\end{aligned}$$

This lead to an expected gradient of each estimate rotation $\hat{R}_i$ of

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

$$\mathbb{E}_j \left[\nabla_{\hat{r}_i} \mathcal{L}_{SO(3)} \left(\hat{R}_i, \hat{R}_j, R_i^j \right) \middle| i = 1 \right] = \frac{1}{2} \text{wrap}_{[-\pi,\pi)} \left[\sum_{j \neq i} \frac{2\pi}{N} (j - i) \right] \omega_0 = \mathbf{0}.$$

For all estimates $\hat{R}_i$, this sums to an integer multiple of $2\pi\omega_0$, which, due to the definition of the $SO(3)$ exponential map, wraps to $\mathbf{0}$.

A.1.2 Critical Point for Iterative Modified Rodrigues Projective Averaging

When optimizing using our Iterative Modified Rodrigues Projective Averaging method, we find that this configuration is only a critical point when the relative orientations between each pair of rotations are equal and opposite, i.e., $R_i^j = R_i^{k\top} \to R_i^j = \exp\left(\pm \frac{2\pi}{N} \omega_0\right)$ and the predicted orientations are initialized at identity: $R_0 = \mathbf{I}$. This only happens when the true orientations $\mathcal{R}$ are evenly spaced about an axis of rotations: $R_i := \exp\left(\left(\theta_0 - i\frac{2\pi}{N}\right)\omega_0\right)$, leaving only axis of rotation ω_0 and the constant angular offset θ_0 about that axis as free parameters.

As we are trying to update these rotations using a method compatible with stochastic gradient descent, we are concerned with the expectation of our update with respect to a sampled pair. In this case, the expected loss and update, defined in Equations (3c) (4) in the main text, respectively, for any projected rotation $\hat{\psi}_i$ and its neighbor $\hat{\psi}_j$ is $\mathcal{L}_{\Psi+}^{i,j} := \left\| \hat{\psi}_i - \phi(q_i^j \otimes \phi^{-1}(\hat{\psi}_j)) \right\|^2$ where q_i^j is the quaternion associated with R_i^j. As all $\hat{\psi}_i$ are initialized to the identity, i.e., $\phi(q_I) = \mathbf{0}$ where q_I is the identity quaternion, we get

$$\mathcal{L}_{\Psi+}^{i,j} := \left\| -\phi^{-1}(q_i^j) \right\|^2 \qquad \nabla_i \mathcal{L}_{\Psi+}^{i,j} := -\phi^{-1}(q_i^j)$$

$$\mathcal{L}_{\Psi-}^{i,j} := \left\| -\phi^{-1}(-q_i^j) \right\|^2 \qquad \nabla_i \mathcal{L}_{\Psi-}^{i,j} := -\phi^{-1}(-q_i^j)$$

The relative rotations in this configuration are

$$R_i^j := \exp\left(\pm \frac{2\pi}{3} \omega_0\right)$$

with relative quaternions $q_i^j := \begin{bmatrix} \cos(\frac{\pi}{3}) & \pm \sin(\frac{\pi}{3})\omega_0 \end{bmatrix}$, which leads to

$$\phi(q_i^j) = \frac{\pm \sin(\frac{\pi}{3})\omega_0}{1 + \cos(\frac{\pi}{3})} = \frac{\pm\omega_0}{\sqrt{3}} \qquad \phi(-q_i^j) = \frac{\mp \sin(\frac{\pi}{3})\omega_0}{1 - \cos(\frac{\pi}{3})} = \pm\sqrt{3}\omega_0.$$

This results in the potential losses for the positive and negative antipodes of

$$\mathcal{L}_{\Psi+}^{i,j} = \|\phi(q_i^j)\| = \frac{1}{3} \qquad \mathcal{L}_{\Psi-}^{i,j} = \|\phi(-q_i^j)\| = 3$$

for all pairs of i, j. Selecting the minimum loss antipodes, we get gradients of

$$\nabla_i \mathcal{L}_{\Psi}^{i,j} = \frac{\mp 1}{\sqrt{3}} \omega_0 \qquad \nabla_i \mathcal{L}_{\Psi}^{i,j} = \frac{\pm 1}{\sqrt{3}} \omega_0,$$

for $j = i + 1$ and $j = i - 1$, respectively. The final expectation of the gradients with respect neighborhood sampling is

$$\mathbb{E}_j \left[\nabla_{\hat{\psi}_i} \mathcal{L}_{SO(3)}(\hat{\psi}_i, \hat{\psi}_j, R_i^j) | i = 1 \right] = \frac{1}{2} \sum_{j \neq i} \nabla_i \mathcal{L}_{\Psi}^{i,j} = \frac{1}{2} \left(\frac{1}{\sqrt{3}} \omega_0 - \frac{1}{\sqrt{3}} \omega_0 \right) = \mathbf{0}.$$

While this demonstrates that our method is not without critical points, even in this simple example, it shows that this configuration is only problematic when the true rotations are equally spaced around an axis of rotation, ω_0, and the estimates are initialized at identity. This compares very favorably to the $SO(3)$ algorithm, which can be in a critical point for any set of relative rotations, R_i^j, and with initialization that can vary with an additional arbitrary constant rotation R_0.

B 1DSfM Datasets

We report results on all structure from motions datasets available in the 1DSfM [1]. Each environment is tested with 5 random initializations and the estimated rotations are updated by each algorithm in batches of size 64, for 20K iterations. While Iterative Modified Rodrigues Projective Averaging, **MRP (Ours)** outperform all **PMG [2]** based methods, the direct **Quaternion** optimization regularly converges to relatively accurate local optima more quickly than ours, as shown in Table S3 and Figure S.1. That being said, our method converges to a more accurate final configuration for most datasets, with respect to mean relative error, Table S4, mean absolute error, Table S1, and median absolute error, Table S2. Our method, as well as the baselines, do not appear to perform well on the larger datasets. As a reminder, this algorithm is specifically designed for training deep learned methods, not for direct rotation optimization. When training deep learned methods, all of the weights are shared, allowing the network to use a single example to improve the accuracy of all rotations near that example. Additionally, we see poor performance on datasets with extremely large observation noise, specifically Gendarmenmarkt, whose median observation error is over 12 degrees. All dataset statistics can be found in Table S5. These datasets do not fully cover the orientation space, and tend to largely cover only variations in yaw. For results on datasets that represent full coverage of the orientation space, see the Uniformly Sampled Rotations dataset or the Neural Network Optimization dataset.

	Mean Absolute Error (°)							
Dataset	4D PGM [2]	6D PGM [2, 3]	9D PGM [2, 4]	Quat	MRP (Ours)	IRLS-GM [5]	IRLS-$\ell_{\frac{1}{2}}$ [6]	MLP [7]
Ellis Island	7.5	7.03	6.41	7.44	**5.59**	3.04	2.71	2.61
NYC Library	9.23	8.32	7.38	8.92	**6.03**	2.71	2.66	2.63
Piazza del Popolo	16.37	16.1	15.88	15.24	**10.03**	4.10	3.99	3.73
Madrid Metropolis	13.55	13.23	11.78	13	**11.25**	5.30	4.88	4.65
Yorkminster	9.13	8.34	7.48	8.56	**5.3**	2.60	2.45	2.47
Montreal Notre Dame	8.17	7.65	6.24	7.76	**4.02**	2.63	2.26	2.06
Tower of London	8.02	8.12	8.36	7.44	**5.58**	3.42	3.41	3.16
Notre Dame	8.71	7.96	7.03	8.55	**5.80**	2.63	2.26	2.06
Alamo	9.41	11.98	10.98	8.74	**6.42**	3.64	3.67	3.44
Gendarmenmarkt	66.41	73.7	68.29	**46.63**	48.82	39.24	39.41	44.94
Union Square	32.46	40.86	40.92	13.44	**10.22**	6.77	6.77	6.54
Vienna Cathedral	29.18	31.42	32.94	18.67	**13.60**	8.13	8.07	7.21
Roman Forum	63.23	64.85	60.51	**18.11**	55.65	2.66	2.69	2.62
Piccadilly	53.35	84.37	106.84	**26.29**	29.98	5.12	5.19	3.93
Trafalgar	121.93	124.18	125.15	**69.65**	91.67	-	-	-

Table S1: **Final Mean Absolute Rotation Error Results on 1DSfM [1] dataset.** Results on the left before the double lines are comparisons of local method after 20K iterations. Results on the right after the double lines are obtained from global methods which require optimizing over global set of relative orientations data at each step. Results associated sections with dashed line are not available from global methods [7].

Dataset	*Median Absolute Error (°)*							
	4D PGM [2]	6D PGM [2, 3]	9D PGM [2, 4]	Quat	MRP (Ours)	IRLS-GM [5]	IRLS-$\ell_{\frac{1}{2}}$ [6]	MLP [7]
Ellis Island	3.68	3.25	3.12	4.04	**2.96**	1.06	0.93	0.88
NYC Library	6.11	5.52	4.85	6.11	**4.04**	1.37	1.30	1.24
Piazza del Popolo	9.51	9.32	9.32	9.29	**6.12**	2.17	2.09	1.93
Madrid Metropolis	9.37	9.06	7.86	9.07	**6.99**	1.78	1.88	1.26
Yorkminster	6.44	5.77	4.56	6.11	**3.29**	1.59	1.53	1.45
Montreal Notre Dame	3.86	3.56	2.86	3.90	**2.30**	0.58	0.57	0.51
Tower of London	4.87	5.84	6.36	4.64	**3.59**	2.52	2.50	2.20
Notre Dame	4.39	3.73	3.09	4.48	**2.61**	0.78	0.71	0.67
Alamo	4.73	5.77	5.16	4.90	**3.48**	1.30	1.32	1.16
Gendarmenmarkt	64.08	71.57	62.9	**43.91**	45.92	7.07	7.12	9.87
Union Square	27.75	34.68	34.84	9.75	**6.85**	3.66	3.85	3.48
Vienna Cathedral	13.80	13.77	16.73	11.67	**6.34**	1.92	1.76	2.83
Roman Forum	53.78	62.46	57.71	**16.56**	41.95	1.58	1.57	1.37
Piccadilly	42.34	79.74	107.32	19.67	**15.09**	2.02	2.34	1.81
Trafalgar	126.71	129.57	130.45	**65.54**	89.09	-	-	-

Table S2: **Final Median Absolute Rotation Error Results on 1DSfM [1] dataset.** Results on the left before the double lines are comparisons of local method after 20K iterations. Results on the right after the double lines are obtained from global methods which require optimizing over global set of relative orientations data at each step. Results associated sections with dashed line are not available from global methods [7].

Dataset	*Mean nAUC*				
	4D PGM [2]	6D PGM [2, 3]	9D PGM [2, 4]	Quat	MRP (Ours)
Ellis Island	22.56	24.07	25.02	15.05	**14.58**
NYC Library	28.53	31.12	32.07	18.20	**16.84**
Piazza del Popolo	37.36	44.18	43.98	25.13	**22.21**
Madrid Metropolis	35.91	38.49	39.15	**24.34**	24.48
Yorkminster	36.82	42.37	44.91	18.71	**18.43**
Montreal Notre Dame	33.97	37.54	40.37	17.69	**16.19**
Tower of London	39.98	45.99	49.54	**18.14**	18.85
Notre Dame	38.77	43.04	46.05	**20.78**	21.10
Alamo	39.87	49.08	50.22	**20.47**	22.05
Gendarmenmarkt	97.45	101.77	100.11	74.76	**71.39**
Union Square	77.22	87.01	89.76	**34.60**	46.20
Vienna Cathedral	72.25	81.07	83.48	**38.74**	42.94
Roman Forum	103.59	105.73	108.88	**52.05**	82.30
Piccadilly	115.83	123.41	126.16	**62.87**	78.31
Trafalgar	126.43	126.49	126.5	**108.19**	115.90

Table S3: Final Mean Normalized AUC on all 1DSfM [1] datasets after 20K iterations

Dataset	*Mean Relative Error (°)*				
	4D PGM [2]	6D PGM [2, 3]	9D PGM [2, 4]	Quat	MRP (Ours)
Ellis Island	12.21	11.49	10.37	11.87	**9.03**
NYC Library	14.29	12.94	11.51	13.67	**9.30**
Piazza del Popolo	21.91	21.24	20.64	20.74	**13.49**
Madrid Metropolis	20.43	19.84	17.85	19.62	**17.09**
Yorkminster	13.73	12.64	11.58	12.97	**8.35**
Montreal Notre Dame	12.5	11.59	9.58	11.93	**6.22**
Tower of London	12.41	12.24	12.44	11.56	**8.71**
Notre Dame	14.15	13.1	11.65	13.86	**9.66**
Alamo	14.23	17.47	15.75	13.17	**9.78**
Gendarmenmarkt	84.21	89.61	84.77	**60.25**	62.98
Union Square	44.44	55.4	55.94	19.98	**15.52**
Vienna Cathedral	41.8	45.62	44.18	26.64	**20.32**
Roman Forum	79.24	77.18	78.03	**25.04**	64.25
Piccadilly	74.25	105.15	122.06	**38.61**	46.21
Trafalgar	126.18	126.42	126.49	**81.28**	97.53

Table S4: Final Mean Relative Error (°) on all 1DSfM [1] datasets after 20K iterations

Dataset	# Nodes	# Edges	Mean Error	Median Error
Ellis Island	227	20K	12.52	2.89
NYC Library	332	21K	14.15	4.22
Piazza del Popolo	338	25K	8.4	1.81
Madrid Metropolis	341	24K	29.31	9.34
Yorkminster	437	28K	11.17	2.68
Montreal Notre Dame	450	52K	7.54	1.67
Tower of London	472	24K	11.6	2.59
Notre Dame	553	104K	14.16	2.7
Alamo	577	97K	9.1	2.78
Gendarmenmarkt	677	48K	33.33	12.3
Union Square	789	25K	9.03	3.61
Vienna Cathedral	836	103K	11.28	2.59
Roman Forum	1084	70K	13.84	2.97
Piccadilly	2152	309K	19.1	4.93
Trafalgar	5058	679K	8.64	3.01

Table S5: Dataset sizes and observation accuracies (°) for all 1DSfM [1] datasets

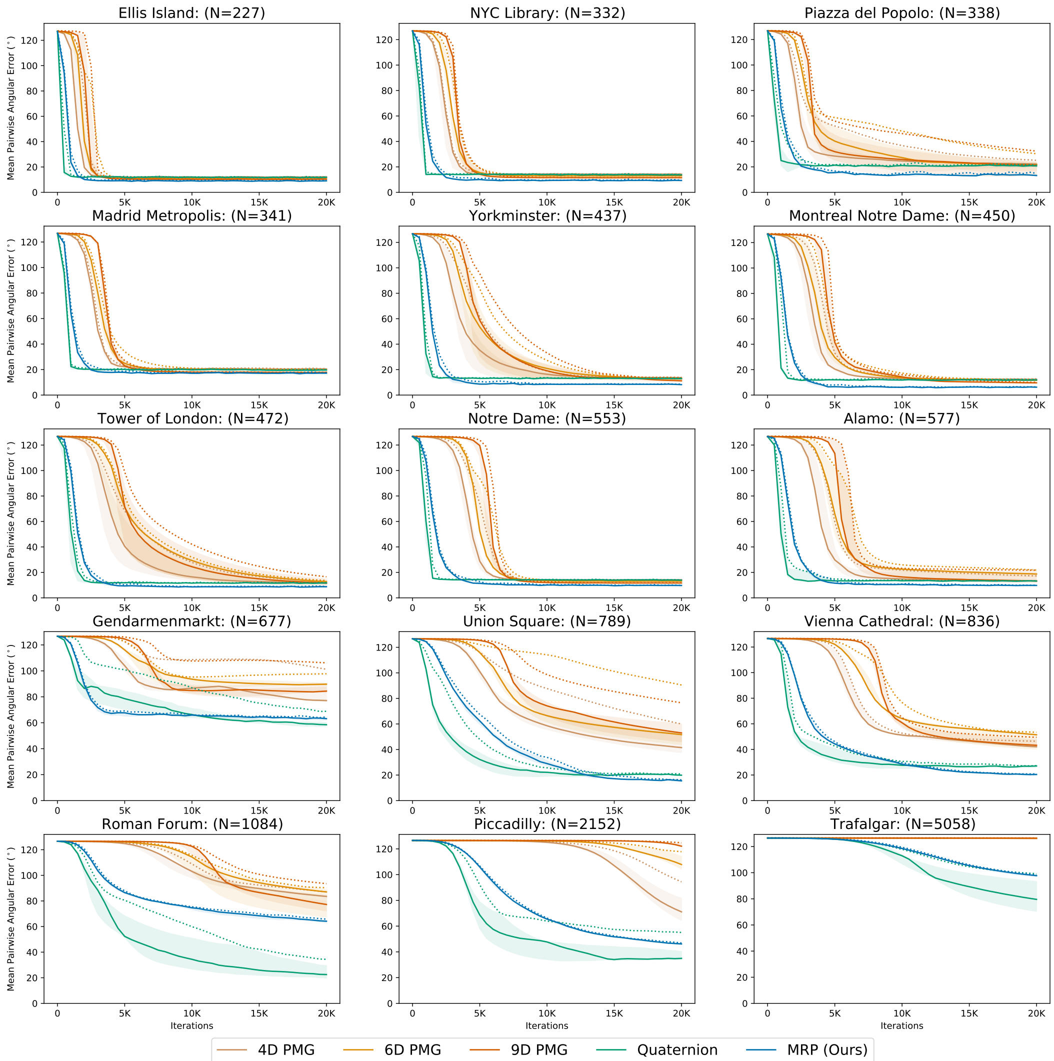
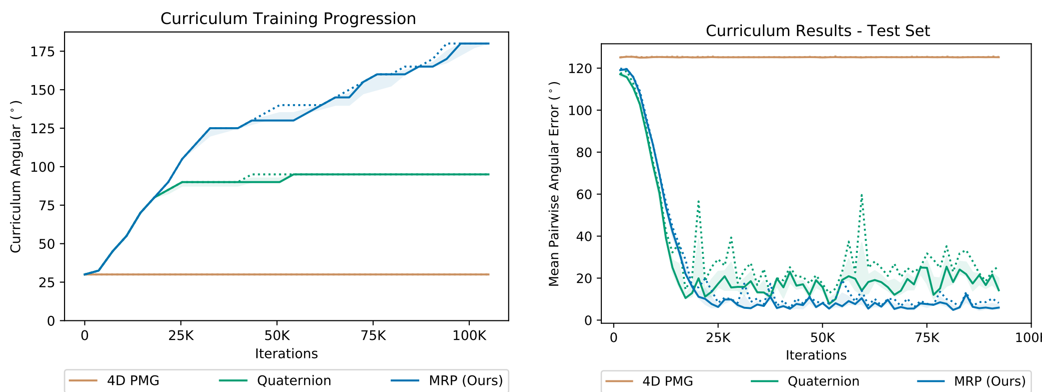


Figure S.1: Optimization results for all 1DSfM [1] datasets, ordered by number of cameras (N). Median average-pairwise angular error (°) is shown with shaded areas representing the first and third quartile over all training sessions. The max average-pairwise angular error for each algorithm at each iteration is shown as a dashed line.

C Curriculum for Neural Network Optimization

We find that a curriculum is required for any relatively supervised method to generalized to unseen orientation. This curriculum training involves starting with a initial base rotation. The model is rendered at this base rotation and a random rotation within $30°$ of this base rotation. This base rotation is initially sampled with $\theta = 30°$ of a constant anchor orientation, until the average training angular error of the previous epoch drops below a given threshold, in this case, $5°$. Once the error drops below this threshold, the angular range, θ, from which this base rotation is sampled is increased by $5°$. This process is repeated, increasing the value of θ by $5°$ each time the error threshold is reached. We find that **MRP (Ours)** is able to complete the curriculum in a reasonable number of iterations, about 100K, achieving a median final pairwise accuracy of $5.19°$ over three training sessions. This test error is sampled from two random rotations across the $SO(3)$, differing from the training error, which are sampled based on the curriculum and are always, at most, $30°$ apart. The quaternion optimization method, **Quaternion**, stalls out at curriculum angle of $90°$, achieving a final pairwise accuracy of $12.41°$ and the **4D PMG [2]** method never gets past the first level of the curriculum, with a final error of $125.09°$. The full training progression of each method, over three random initialization each, can be seen in Figure C

One way this curriculum could be applied to captured data as follows: given a video, a curriculum could be established based on temporal proximity in the video. Choosing an arbitrary initial frame of the video as a anchoring frame, a curriculum can be generate by increasing temporal distance to neighboring frames until the entire video has been used in training.



Curriculum Angle (left) and Average Pairwise Error (right), sampled over the full orientation space for three training sessions with each method. Median average-pairwise angular error (°) is shown with shaded areas representing the first and third quartile over all training sessions. The max average-pairwise angular error for each algorithm at each iteration is shown as a dashed line.

Curriculum Angle (left) and Average Pairwise Error (right), sampled over the full orientation space for three training sessions with each method. Median average-pairwise angular error (°) is shown with shaded areas representing the first and third quartile over all training sessions. The max average-pairwise angular error for each algorithm at each iteration is shown as a dashed line.

Figure S.2:
Curriculum Angle (left) and Average Pairwise Error (right), sampled over the full orientation space for three training sessions with each method. Median average-pairwise angular error (°) is shown with shaded areas representing the first and third quartile over all training sessions. The max average-pairwise angular error for each algorithm at each iteration is shown as a dashed line.

D 3D Object Rotation Estimation via Relative Supervision from Pascal3D+ Images

D.1 Experimental Setup

Pascal3D+ [8] is a standard benchmark for categorical 6D object pose estimation from real images. We follow similar experimental settings as in [2, 4] for 3D object pose estimation from single images. Following [2, 4], we discard occluded or truncated objects and augment with rendered images from [9]. We report 3D object pose estimation via relative orientation supervision results on two object categories of Pascal3D+ image dataset: *sofa* and *bicycle*. We compare our method **MRP** with five baselines: **Quaternion**, **4D PMG** [2], **6D PMG** [2, 3], **9D PMG** [2, 4] and **10D PMG** [2, 10].

We use ResNet18 [11] as the model backbone to predict object rotation from single images. The model is supervised by the geodesic error between the induced relative orientation between the predicted absolute orientations for a pair of images, and the relative orientation between the ground truth absolute orientations for the image pair.

Specifically, **MRP** is supervised by the geodesic distance on the MRP manifold as described in Equation 3 and 4 in the main paper. **Quaternion** is supervised by quaternion geodesic distance as described in Equation 2 in the main paper. While **4D/6D/9D/10D PMG** are supervised by the geodesic error derived from projective manifold gradients as in [2]. We use the same batch size of 20 as in [2, 4], and use Adam [12] with learning rate of 1e-4.

D.2 Result Analysis

Results for *sofa* showed in Figure S.3 and Table S6. Results for *bicycle* showed in Figure S.4 and Table S7. **Pascal3D+ Sofa.** For *sofa* category, as seen in Table S6, we find that after 50K training iterations, **MRP (Ours)** achieves a mean angular pairwise error of $14.09°$ on the test set, outperforms all other baselines. **Quaternion** achieves the worst error out of all methods, with final angular pairwise error of $26.35°$. Besides achieving the lowest test angular error, we also find that **MRP (Ours)** has the fastest convergence speed, as seen in Figure S.3.

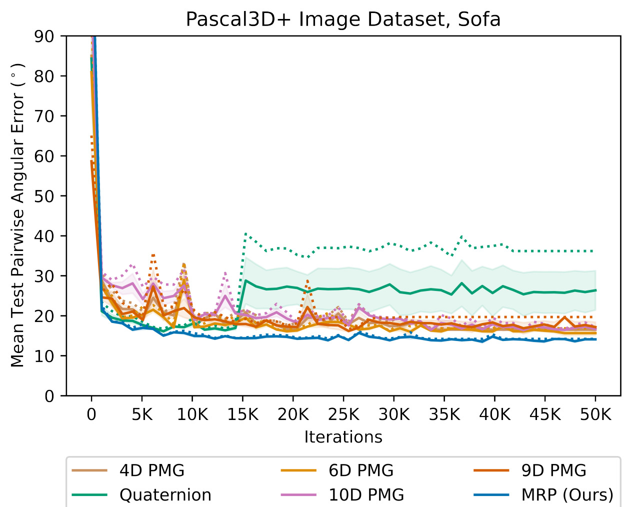


Figure S.3: **3D Object Pose Estimation via Relative Supervision on Pascal3D+ *Sofa* Images.** Mean test pairwise angular error in degrees of *sofa* at different iterations of training. Trained over 50K training steps for 2 random seeds per method. Solid lines stand for mean errors, dashed line stand for max errors, and shaded area represents error standard deviation.

Algorithm	Mean Test Angular Pairwise Error (°)
4D PMG [2]	16.53
6D PMG [2, 3]	15.65
9D PMG [2, 4]	17.17
10D PMG [2, 10]	16.67
Quaternion	26.35
MRP (Ours)	**14.09**

Table S6: **Final Mean Test Angular Pairwise Error on Pascal3D+ *sofa* Images after 50K training iterations.**

Pascal3D+ Bicycle. For *bicycle* category, as seen in Table S7, we find that after 50K training iterations, **MRP (Ours)** achieves a mean angular pairwise error of $29.21°$ on the test set, outperforms all other baselines. Besides achieving the lowest test angular error, we also find that **MRP (Ours)** has the fastest convergence speed, as seen in Figure S.4.

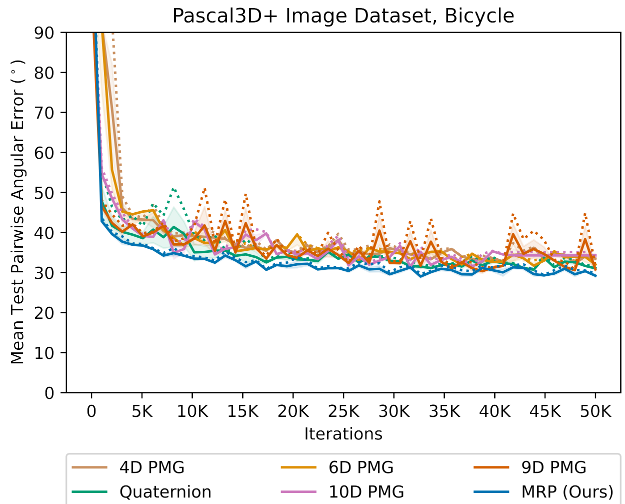


Figure S.4: **3D Object Pose Estimation via Relative Supervision on Pascal3D+ *Bicycle* Images.** Mean test pairwise angular error (°) of *bicycle* at different iterations of training. Trained over 50K training steps for 2 random seeds per method. Solid lines stand for mean errors, dashed line stand for max errors, and shaded area represents error standard deviation.

Algorithm	Mean Test Angular Pairwise Error (°)
4D PMG [2]	33.48
6D PMG [2, 3]	31.73
9D PMG [2, 4]	30.78
10D PMG [2, 10]	35.30
Quaternion	31.06
MRP (Ours)	**29.21**

Table S7: **Final Mean Test Angular Pairwise Error on Pascal3D+ *bicycle* Images after 50K training iterations.**

E 3D Object Rotation Estimation via Relative Supervision from ModelNet40 Point Clouds

E.1 Experimental Setup

ModelNet40 [13] is a standard benchmark for categorical 6D object pose estimation from 3D point clouds. We follow similar experimental settings as in [2]. We follow the same train/test data split as in [2] and report 3D object pose estimation via relative orientation supervision results on the *airplane* category of ModelNet40 dataset. We compare our method **MRP** with four baselines: **Quaternion**, **4D PMG** [2], **6D PMG** [2, 3], **9D PMG** [2, 4] and **10D PMG** [2, 10] . We use PointNet++ [14] as the model backbone to predict 3D absolute object rotation from single point cloud generated from the ModelNet40 3D CAD models, as in [2]. The model is supervised by the geodesic error between the induced relative orientation between the predicted absolute orientations for a pair of point clouds, and the relative orientation between the ground truth absolute orientations for the point cloud pair.

We sample 1024 points per point cloud as in [2, 4], use a batch size of 14. As for training, we use Adam [12] with learning rate of 1e-3, and run over 1 trial for each method.

We find that for any of the compared methods to generalize to unseen test point cloud instances, a curriculum is needed. We train with a curriculum over the rotation space, the curriculum details can be found in Section C. Specifically we start with base rotation range with $\theta = 30°$ of a constant anchor orientation, and θ is increased by $5°$ whenever the previous mean epoch train angular error drops below the curriculum threshold, $5°$. To speed up the training procedure, we increase this curriculum threshold to $8°$ once θ gets to $125°$.

E.2 Result Analysis

Results on the *airplane* object class from ModelNet40 dataset is shown in Figure S.5 and Table S8.

As seen in Figure S.5 and Table S8, **MRP (Ours)** is able to go through the curriculum in 250K iterations, reaching final test pairwise angular error of $5.49°$. **Quaternion** goes through the curriculum much slower, reaching curriculum angle $\theta = 90°$ at the end of 250K steps. **4D PMG**, **6D PMG**, **9D PMG** and **10D PMG**, on the other hand, is not able to progress beyond the original curriculum angle of $\theta = 30°$, reaching final test pairwise angular error around $35°$ after 200K iterations. In summary, **MRP (Ours)** achieves faster convergence rate than all baselines, and is able to achieve final test angular error on the order of $5°$ after progressing through the curriculum.

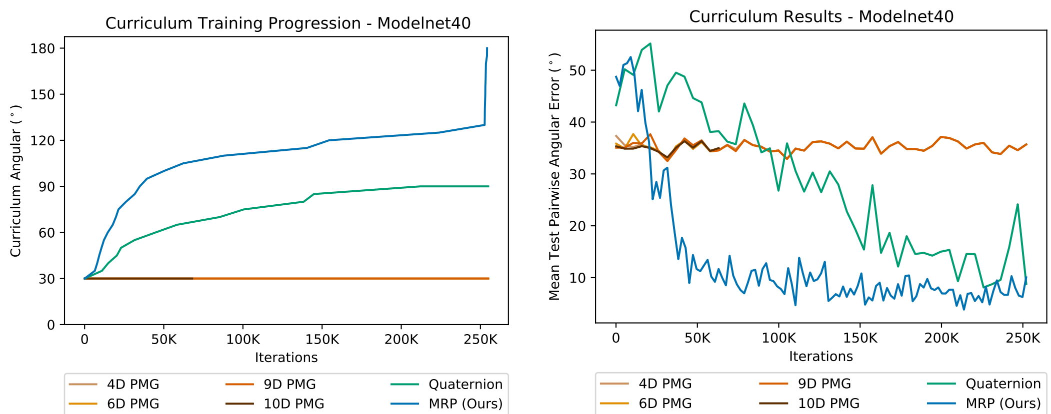


Figure S.5: **3D Object Rotation Estimation via Relative Supervision from ModelNet40 Point Clouds - *airplane*. Left:** Curriculum angle progression through training iterations. **Right:** Average test pairwise angular error (°), sampled over the full orientation space for 1 training session with each method.

Algorithm	Mean Test Angular Pairwise Error (°)
4D PMG [2]	35.35
6D PMG [2, 3]	34.12
9D PMG [2, 4]	35.80
10D PMG [2, 10]	35.26
Quaternion	12.86
MRP (Ours)	**5.49**

Table S8: **Final Mean Test Angular Pairwise Error on ModelNet40 *airplane* Point Clouds after at most 250K training iterations.**

F Absolute Orientation Supervision

F.1 Experimental Setup

In this paper, we are assuming that only relative orientation supervision is available; however, in this section we explore how different orientation representations perform if absolute orientation supervision is available, and specifically how Modified Rodriguez Parameters (MRP) [15] used in this paper compare. To explore this, we perform an experiment on rotation estimation from 2D images of rendered YCB drill supervised with absolute orientation instead of relative supervision. We follow the same experimental setup as in Section 6.2 in the main paper, utilizing ResNet18 [11] as the model backbone to predict absolute 3D object orientations from sets of 2D rendered object images, rendered at 100 random rotations each. The neural network model is supervised by the geodesic error between the predicted absolute orientation and the ground truth absolute orientation. We compare the performance of different rotation parameterizations on this task. Specifically, we compare the Modified Rodriguez Parameters (MRP) [15] (**Oracle-MRP**) with Quaternions (**Oracle-Quaternion**). Each method is trained for 10K steps, over 8 different rendered image sets.

We report the mean global pairwise angular error over the whole set of 100 images over the training process in Table S9.

F.2 Result Analysis

We report results on three metrics: 1) mean global train absolute angular error; 2) median global train absolute angular error; 3) percentage of runs that converge with final pairwise angular error $< 2^\circ$ after 10K steps, which is referred to as 2° *Acc*. Specifically, global relative angular error is calculated as the all-pair relative angular error for all pairs within the image set of 100. As see in Table S9, **Oracle-MRP** achieves comparable but larger mean and median pairwise angular error compared to **Oracle-Quaternion**, while both methods achieves the same 2° *Acc* of 87.5%. In summary, through this simple experiment, we find that MRP is able to achieve comparable but slightly worse train error for absolute orientation supervision compared to quaternions. Thus in the case of direct pose supervision, MRP may not be the best choice of rotation representation; using an open manifold such as in MRP is beneficial only in the case of relative pose supervision.

Algorithm	Mean Error (°)	Median Error (°)	2° Acc (%)
Oracle-Quaternion	**1.58**	**1.56**	**87.5**
Oracle-MRP	1.81	1.86	**87.5**

Table S9: **Absolute Orientation Supervision for Image Based Rotation Estimation from Rendered YCB Drill Images using MRP vs Quaternions Parametrization.** Final mean, median angular train error ($\circ$) and convergence ($< 2^\circ$) percentage for image based rotation estimation from rendered YCB drill images with absolute orientation supervision, after 10K training steps over 8 sets of 100 rendered images.

G Object Orientation Prediction Qualitative Visual Results

We further show some qualitative visual illustrations of the object orientation prediction of trained model at convergence, trained using our iterative MRP averaging method via relative orientation supervision below. Examples from orientation estimation on the rendered YCB drill data as described in Section 6.2 in the main paper is shown in Figure S.6. Examples from orientation estimation on unseen Pascal3D+ *sofa* category data as described in Supplement Section D.1 is shown in Figure G, and prediction on unseen Pascal3D+ *bicycle* category is shown in Figure S.8.

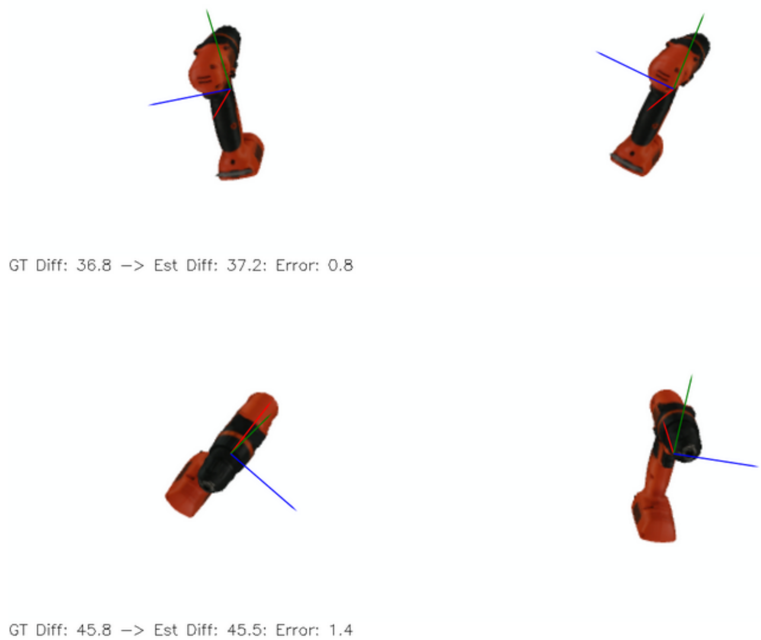


Figure S.6: **Qualitative Visual Examples for Object Orientation Estimation of MRP (Ours) on Rendered YCB Drill Images.** We show qualitative visual examples of predicted object 3D orientation by converged orientation prediction model trained via iterative MRP averaging with relative orientation supervision, the model is evaluated after training for 10K steps from neural net optimization experiment described Sec 6.2 of the main paper. The predicted orientation is shown as coordinate frame (x, y, z). On the bottom of each example, we show in text of the ground truth relative orientation angular difference (°) between the pair of images, and their predicted relative orientation angular difference (°) induced from the absolute object orientation predicted for each image. And finally we show the difference between the predicted relative angular difference and the ground truth relative angular difference as angular error (°).

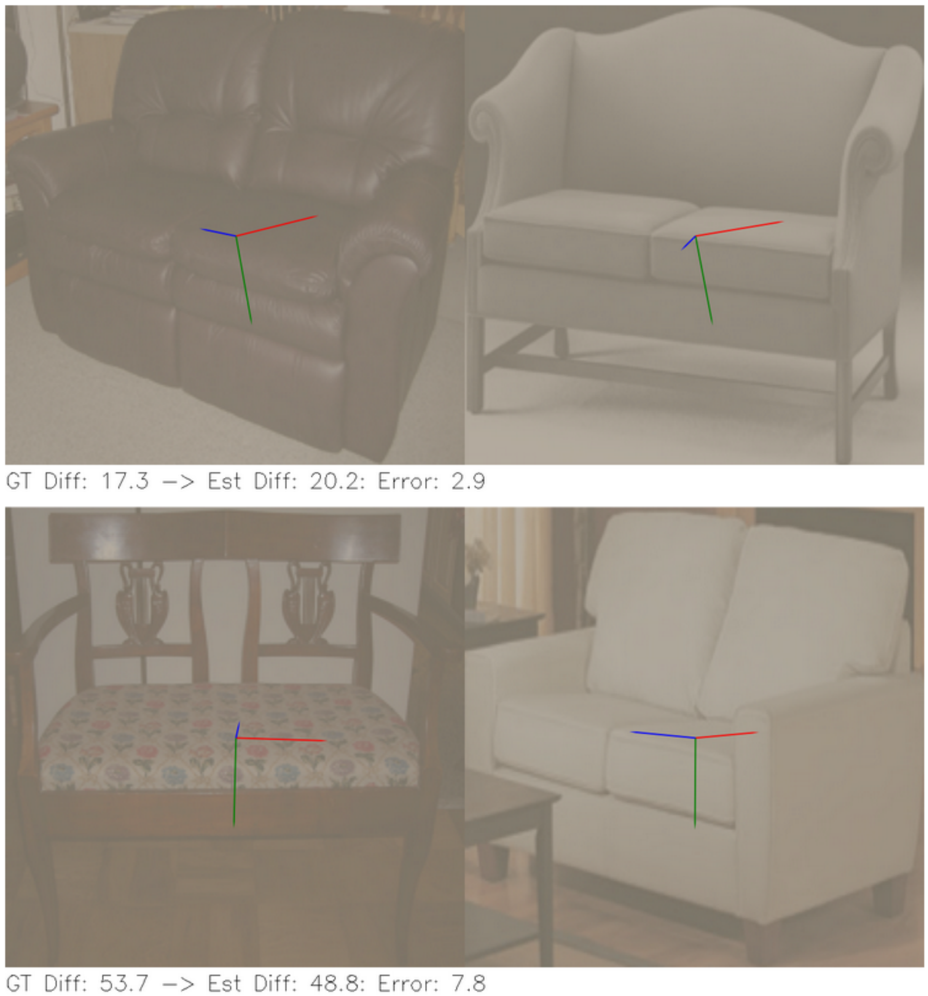


Figure S.7: **Qualitative Visual Examples for Object Orientation Estimation of MRP (Ours) on Unseen Pascal3D+ *Sofa* Images.** We show qualitative visual examples of predicted object 3D orientation by converged orientation prediction model trained via iterative MRP averaging with relative orientation supervision, the model is evaluated after training for 50K steps from 3D object rotation estimation on Pascal3D+ experiment as described Sec D of this supplement. The predicted orientation is shown as coordinate frame (x, y, z). On the bottom of each example, we show in text of the ground truth relative orientation angular difference (°) between the pair of images, and their predicted relative orientation angular difference (°) induced from the absolute object orientation predicted for each image. And finally we show the difference between the predicted relative angular difference and the ground truth relative angular difference as angular error (°).

Figure S.8: **Qualitative Visual Examples for Object Orientation Estimation of MRP (Ours) on Unseen Pascal3D+ *Bicycle* Images** We show qualitative visual examples of predicted object 3D orientation by converged orientation prediction model trained via iterative MRP averaging with relative orientation supervision, the model is evaluated after training for 50K steps from 3D object rotation estimation on Pascal3D+ experiment as described Sec D of this supplement. The predicted orientation is shown as coordinate frame (x, y, z). On the bottom of each example, we show in text of the ground truth relative orientation angular difference (°) between the pair of images, and their predicted relative orientation angular difference (°) induced from the absolute object orientation predicted for each image. And finally we show the difference between the predicted relative angular difference and the ground truth relative angular difference as angular error (°).

References

[1] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[2] J. Chen, Y. Yin, T. Birdal, B. Chen, L. Guibas, and H. Wang. Projective manifold gradient layer for deep rotation regression. *arXiv preprint arXiv:2110.11657*, 2021.

[3] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[4] J. Levinson, C. Esteves, K. Chen, N. Snavely, A. Kanazawa, A. Rostamizadeh, and A. Makadia. An analysis of svd for deep rotation estimation. *Advances in Neural Information Processing Systems*, 33:22554–22565, 2020.

[5] A. Chatterjee and V. M. Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 521–528, 2013.

[6] A. Chatterjee and V. M. Govindu. Robust relative rotation averaging. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):958–972, 2017.

[7] Y. Shi and G. Lerman. Message passing least squares framework and its application to rotation synchronization. In *International Conference on Machine Learning*, pages 8796–8806. PMLR, 2020.

[8] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE, 2014.

[9] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. 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*, pages 2686–2694, 2015.

[10] V. Peretroukhin, M. Giamou, D. M. Rosen, W. N. Greene, N. Roy, and J. Kelly. A smooth representation of belief over so(3) for deep rotation learning with uncertainty. *In Proceedings of Robotics: Science and Systems (RSS'20)*, 2020.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[14] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[15] J. Crassidis and F. Markley. Attitude estimation using modified rodrigues parameters. In *Flight Mechanics/Estimation Theory Symposium*, pages 71–86. NASA, 1996.