

# Inferring Space Object Orientation with Spectroscopy and Convolutional Networks

**Matthew Phelps**

*United States Space Force*

**J. Zachary Gazak, Thomas Swindle, Justin Fletcher**

*United States Space Force*

**Ian McQuaid**

*Air Force Research Laboratory*

## ABSTRACT

Accurate inference of a space object's orientation is imperative for deriving its operational status and coordinating effective space traffic management at large. To formulate the framework necessary for solving the problem of orientation inference, we analyze several standard mathematical representations of rotation with an emphasis on continuity, uniqueness, and deep learning efficacy. On this basis, we are naturally led to the implementation of a lesser known but well-behaved 6D representation of rotation. For the input of our inference models, we employ a distance-invariant observational technique that has long been used to probe the furthest reaches of the universe at the smallest scales – spectroscopy. Facilitated by deep convolutional neural networks (CNN's), we investigate the viability of using simulated raw, long slit spectroscopic images to infer the orientation of space objects in the nonresolved regime of large orbital radii. We present methods and results of training CNN's on spectral images of several space objects with an aim to i) standardize the measures used in rotation analysis, ii) establish an upper bound on spectral-based performance, and iii) provide a simple-scenario baseline for the extension of future work in the application of spectroscopy to space domain awareness.

## 1. INTRODUCTION

One of the fundamental goals in the field of space domain awareness (SDA) is to provide an increasingly complete specification of the physical state and trajectory associated with a target space object orbiting Earth. While the set of six Keplerian elements (commonly encoded as 'two-line elements') form a sufficient basis to describe the unperturbed orbit of the center of mass, additional degrees of freedom remain concerning the motion of a body about its center of mass. To continue advancing towards a more complete description of the physical state, one must also specify the rotational kinematics of the body, namely its orientation and angular velocity relative to a given reference frame. Increasing the complexity further, if the body itself is not rigid, yet even more degrees of freedom must be determined to account for arbitrary articulations of mass extended by the space object.

In this work, we focus on a specific subset of the aforementioned degrees freedom – rotational orientation. We approach orientation inference as a computer vision task implemented by deep convolutional networks. The description of rotation itself and its generalization to both  $n$  dimensions and complex fields has a strikingly rich foundation in both mathematics and physics, serving as an integral component in the study of group theory (Lie groups and Lie algebras) and the description of fundamental symmetries that form the basis of modern theoretical physics. In light of this vast groundwork, in Sect. 3 we pay due diligence to investigate numerous mathematical representations of rotation and assess their utility within deep learning applications. The selection of the appropriate representation of rotation will serve to define the framework by which we approach the task of learning orientation and will provide a basis for measuring model performance.

The physical scenario we target broadly concerns ground based observation of resident space objects located at orbital radii too large to form an optically 'resolved' image. For the purposes of this work, our observations are modeled after the 3.67 meter Advanced Electro Optical System Telescope (AEOS) stationed at the summit of Mt. Haleakala, Maui. To provide an intuition of how optical resolvability varies with altitude, Fig. 1 (right) shows two images of the Hubble spacecraft as simulated through a high-resolution adaptive optics sensor on AEOS. In Fig. 1 (top right) we

**DISTRIBUTION A. Approved for public release: distribution is unlimited.**  
**Public Affairs release approval #AFRL-2021-2863**

observe Hubble at low earth orbit (LEO) where distinguishing features are resolvable and orientation can be inferred to a reasonable approximation. As we increase the orbital radii out to geosynchronous orbit (GEO), we endure a significant loss of information as the Hubble spacecraft as shown in Fig. 1 (bottom right) now comprises merely a handful of pixels, effectively removing the ability to resolve distinguishing features, let alone infer its orientation.

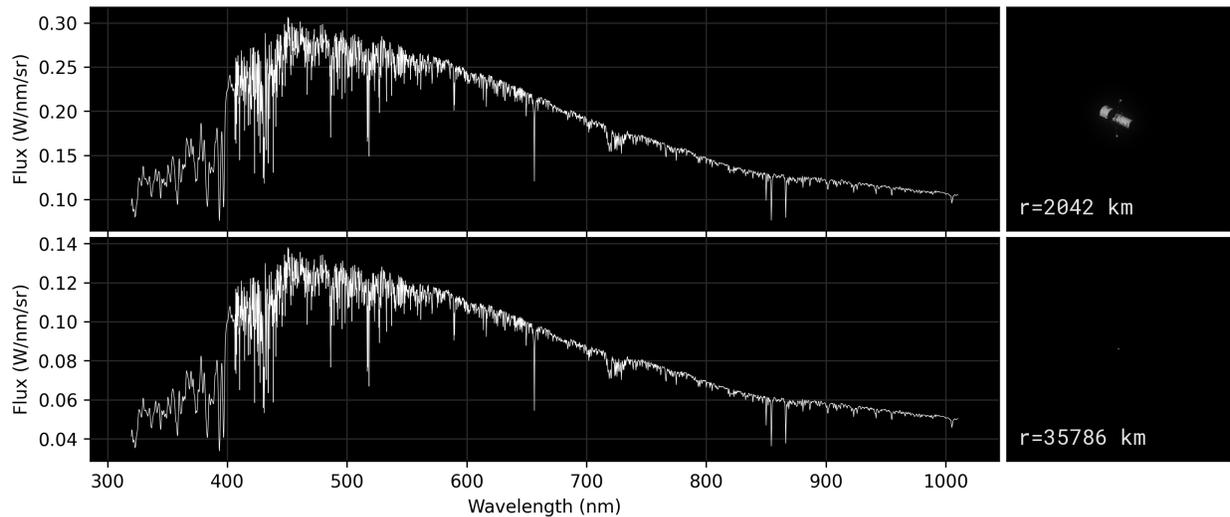


Fig. 1: Reflection spectrum associated with the Hubble spacecraft at LEO ( $r = 2042\text{km}$ , top left) and GEO ( $r = 35786\text{km}$ , bottom left) along with respective optical image renders (right) as simulated through the 3.67 meter AEOS telescope. As we observe spectra while varying the radial distance, to an excellent approximation the flux is simply scaled by  $r^{-2}$ . Consequently, the structure of Hubble’s spectrum is invariant to radial separation. In contrast, as we reposition Hubble from LEO to GEO, we observe a critical loss of information in the associated optical images (right).

In order to overcome the ever increasing loss of information inherent to optical images as object-telescope separation is increased, we seek a method of measurement that is, to a good approximation, scale-invariant. To find such a source of physical information, one need only to look at the method used to probe the atomic structure of celestial bodies residing at the deepest extents of space: spectroscopy. The key invariance afforded by spectroscopy is that as telescope-object separation increases, the spectral flux, to an excellent approximation, *uniformly* scales by  $1/r^2$ , thus preserving the functional form of its wavelength dependence. Fig. 1 (left) illustrates the constant scaling of Hubble’s wavelength structure as it moves from an optically resolvable region (LEO) to an optically unresolvable region (GEO).

Although spectroscopy may possess the desired scaling properties, the information encoded in spectral images is of an entirely different nature altogether in comparison to resolved imagery. While the SpectraNet model of [7] demonstrates that raw spectra possess enough semantic structure to distinguish satellite classes from one another (with excellent accuracy), we point out that SpectraNet was trained to classify *through* random orientations of space objects, i.e. it learned a rotationally invariant mapping. In contrast, in this work we aim the solve the rather different problem of regressing spectral images to the continuous and higher dimensional space of 3D orientations. As such, it is not apriori obvious whether i) spectra will possess enough variation to construct a sufficiently complete mapping from spectra to orientation or whether ii) such a mapping will have a large enough signal to noise ratio to enable an efficient learning by CNN’s (convolutional neural networks).

To address these open-questions, we design a series of two experiments. In the first experiment, we fix two rotational degrees of freedom and form a dataset of object orientations corresponding to rotation about a single fixed axis (2D rotation). The particular rotation axis is selected to lie along one of the principle axes of inertia in order to provide highly regular and predictable motion of the space object. This experiment aims to provide a scenario of absolute minimum complexity that still encompasses the core challenge of constructing a mapping from spectra to a continuous interval with periodic behavior. The outcome of this experiment provides strong guidance as to whether spectral information has a sufficiently strong correlation to orientation. In the second experiment, we model the full task of orientation inference (3D rotation) and seek to establish a baseline measure of performance to be used in future spectra-based applications of rotation.

**DISTRIBUTION A. Approved for public release: distribution is unlimited.**  
**Public Affairs release approval #AFRL-2021-2863**

## 2. RELATED WORK

The application of spectroscopy to SDA was first introduced via SpectraNet [7], where simulated spectrograph observations were used to train CNN's for classifying satellites. Achieving accuracies exceeding 80% on a 64-class problem of randomly orientated satellites, SpectraNet is the source of inspiration for our task and serves as the foundation upon which many of our methods are based. Using similar techniques, [8] is an additional spectroscopic application, tuned towards the detection of closely separated objects (CSO's).

The use of computer-vision based methods of pose inference in SDA is slightly more widespread, the most relevant being [15], which demonstrated success in training CNN's to infer satellite orientation from degraded images in the optical regime (LEO). Other pose-related SDA applications include attitude estimation using light curves [2], [3] and [4] which optimizes CNN's for the task of low power, onboard sensing of satellite orientations using optical imagery. Computer vision applications of SDA beyond pose estimation are numerous, one of the most promising being SatNet [6], which performed object detection of GEO satellites on a large dataset of ground-based imagery, achieving a max F1-score of 0.971.

An appreciable body of work outside SDA exists in applying computer-vision to the task of pose-estimation, often formulated as the inference of both translational and rotational degrees of freedom [16]. Notable work in this domain includes [17], which provides techniques to infer orientation of objects possessing rotational symmetries and [20] which uses an autoencoder trained on augmented synthetic object orientations to learn an implicit representation of rotation allowing inference without copious amounts of labeled data.

Regarding the theoretical foundation upon which the task of orientation estimation is formulated, an analysis of the representations of rotation from a continuity and topological perspective was explored in [23], leading to the proposition of new representations of rotation, a result by which much of our discussion of Sect. 3 is centered. For the purposes of constructing performance metrics for pose estimation, one may refer to [12] which provides mathematically well-defined metrics on standard 3D rotation representations.

## 3. ROTATION REPRESENTATIONS

A number of mathematical representations of rotation are available to unambiguously describe the orientation of an object in three dimensional space such as Euler angles,  $SO(3)$ , axis-angle, or quaternions. In order to progress towards selecting the representation most suitable for our problem statement, in the following we i) establish  $SO(3)$  as the fundamental representation ii) analyze candidate representations based on their properties of continuity and faithfulness to the underlying fundamental representation and iii) provide potential distance measures which may serve as performance metrics. In this context of this paper, we follow [23] and consider a representation as continuous if a set of connected elements in  $SO(3)$  remains connected when mapped to the new representative space. We may formalize this notion as follows [23]: given a representative rotation space  $Y$  and the fundamental representation space  $SO(3)$  there must exist mappings  $f$  and  $g$  which obey

$$f : Y \rightarrow SO(3), \quad g : SO(3) \rightarrow Y, \quad f(g(x)) = x, \quad g \text{ is continuous.} \quad (1)$$

While the first three relations in (1) establish the definition of a valid representation, the continuity of  $g$  ensures that an arbitrary set of connected elements in  $SO(3)$  remains connected under  $g$ .

### 3.1 $SO(3)$

As each orientation of an object in three-dimensional Euclidean space can be associated by a unique element in  $SO(3)$ , this representation serves as the fundamental rotation representation for our task. By definition,  $SO(3)$  is the group of special orthogonal  $3 \times 3$  matrices. To help deconstruct this concise definition, a group is simply a set with the property that any two elements within the set may be composed together via some operation (matrix multiplication in the case of  $SO(3)$ ) to form a new element that is also a member of the group<sup>1</sup>. Representing an element of  $SO(3)$  as the  $3 \times 3$  matrix  $\mathbf{R}$ , the notion of orthogonality and 'special'-ness respectively refer to matrices that behave as

$$\mathbf{R}\mathbf{R}^T = \mathbb{I}, \quad \det \mathbf{R} = 1. \quad (2)$$

---

<sup>1</sup>Additionally, members of the group must obey the axioms of associativity, identity, and invertability.

**DISTRIBUTION A. Approved for public release: distribution is unlimited.  
Public Affairs release approval #AFRL-2021-2863**

These constraints translate into limiting ourselves to transformations that preserve both length (a rotated vector does not change in magnitude) and coordinate system orientation (a right handed coordinate system remains right handed). Counting the degrees of freedom, we note that orthonormality provides 5 constraints and the unit determinate 1 constraint, thus reducing the 9 components of a 3x3 matrix down to 3, just as expected for describing 3D rotation.

To determine whether two SO(3) rotations are close or far apart from one another, the notion of a metric on SO(3) is in fact mathematically well defined<sup>2</sup> and given by a quantity occasionally referred to as the SO(3) geodesic distance [12]  $\Phi_0 : SO(3) \times SO(3) \rightarrow \mathbb{R}^+$ ,

$$\Phi_0(\mathbf{R}_1, \mathbf{R}_2) = \|\log(\mathbf{R}_1 \mathbf{R}_2^T)\|_F, \quad (3)$$

where  $\|\cdot\|$  denotes the element-wise Frobenius norm  $\|\mathbf{A}\| = (\sum_i \sum_j |A_{ij}|^2)^{1/2}$ . To avoid the complexity of the matrix logarithm, one can transform (3) into the computationally simpler form (see Appendix 9.1) of

$$\Phi_1(\mathbf{R}_1, \mathbf{R}_2) = \cos^{-1} \left[ \frac{1}{2} (\text{Tr}(\mathbf{R}_1 \mathbf{R}_2^T) - 1) \right]. \quad (4)$$

### 3.2 Euler Angles

Ostensibly the most intuitive representation, Euler angles are characterized by a set of three angles that signify the magnitude of rotation along a specified set of coordinate axes. The axes themselves may be fixed relative to an external observer (extrinsic) or fixed to the body of the moving object (intrinsic). Importantly, an object's orientation is given by a *composition* of Euler angles, namely a sequence of three transformations, with each transformation effecting a rotation about one axis.

To expose some consequences of composing rotations and provide a connection to our fundamental representation, we can construct the map between Euler angles and SO(3) by considering each individual transformation as a rotation matrix parameterized by one Euler angle. For example, defining Euler angles as  $(\alpha, \beta, \gamma)$  with respective extrinsic axes  $(\hat{x}, \hat{y}, \hat{z})$  we have,

$$\mathbf{R}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}, \quad \mathbf{R}_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}, \quad \mathbf{R}_z(\gamma) = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (5)$$

in which a final orientation  $\mathbf{R}$  can be composed by multiplying rotation matrices

$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z. \quad (6)$$

If we imagine applying the composite rotation  $\mathbf{R}$  to a 3-vector  $\mathbf{v}$ , note that the transformed vector  $\mathbf{v}' = \mathbf{R}\mathbf{v}$  is reached by first rotating around the z-axis by an angle  $\gamma$ , then the y-axis by angle  $\beta$ , and lastly around the x-axis by angle  $\alpha$ . Insight into the dependence on the order of rotations can be gained by looking at the commutation relation between the infinitesimal generators  $\mathbf{K}_i$  of the SO(3) group, viz.

$$[\mathbf{K}_i, \mathbf{K}_j] = \mathbf{K}_i \mathbf{K}_j - \mathbf{K}_j \mathbf{K}_i = \sum_k \epsilon_{ijk} \mathbf{K}_k, \quad (7)$$

where  $\epsilon_{ijk}$  is the antisymmetric Levi-Civita tensor. With nonvanishing commutation, (7) tells us that the order by which rotations are carried out matters. Specifically, in order to define a unique orientation, information about the Euler angles and axes themselves is not sufficient; one must also specify the ordering of transformations. As a result of this ordering, a myriad of conventions exist for specifying the choice of coordinate axes<sup>3</sup>. In the experiments to follow, we use the convention most commonly adopted in aerospace engineering, namely, an intrinsic coordinate system with axes defined by the principle axes of inertia of the rotating body and referred to as roll, pitch, and yaw.

In addition to the ordering of rotations, another interesting behavior can be revealed by representing Euler angle transformations as SO(3) matrices. Using the coordinate system and angles as defined in (5), consider a rotation with

<sup>2</sup>As a compact Lie group, the inner product of the Lie algebra generators (i.e. elements of  $so(3)$ ) can be used to define an infinitesimal length. By exponentiating the generators to form group elements and integrating over the shortest path between the two points, we can recover (3).

<sup>3</sup>In fact, only two of the specified rotation axes need to be distinct. It can then be shown that a total of twelve different rotation sequences can be defined for *each* reference frame type (intrinsic or extrinsic).

$\beta = \pi/2$ . Evaluating the final rotation matrix we have

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ \sin(\alpha + \gamma) & \cos(\alpha + \gamma) & 0 \\ -\cos(\alpha + \gamma) & \sin(\alpha + \gamma) & 0 \end{pmatrix}. \quad (8)$$

Inspecting (8), we see that the composite rotation is not defined by  $\alpha$  and  $\gamma$  independently, but rather only through their sum ( $\alpha + \gamma$ ). As a result, there are many (infinite) combinations of  $\alpha$  and  $\gamma$  yield the underlying SO(3) matrix, and thus the mapping between Euler angles and SO(3) is many-to-one, i.e. non-unique.<sup>4</sup>

In regards to continuity, one may note that for  $\alpha$  defined in the range  $[0, 2\pi]$ , (5) implies

$$\lim_{\alpha \rightarrow 0^-} \mathbf{R}_x(\alpha) = \lim_{\alpha \rightarrow 2\pi^+} \mathbf{R}_x(\alpha) = \mathbb{I}. \quad (9)$$

With  $\mathbf{R}_x$  representing a two-dimensional transformation along the x-axis, we see that a connected set of elements that lie in the in the space spanned by  $\mathbf{R}_x$  (i.e.  $S^1$ , the lower right block matrix of  $\mathbf{R}_x$ ) become disconnected when mapped to the  $[0, 2\pi]$  Euler angle representation. Consequently, not only is the mapping from Euler angles to SO(3) non-unique, it is also discontinuous.

Despite these properties, the use of Euler angles within deep learning is not entirely infrequent, and thus, to provide a basis of comparison to SO(3), we present a few sensible choices of functions to measure the degree of similarity between two Euler angle rotations. Here we define  $E \in \{(\alpha, \beta, \gamma) \mid \alpha \in [0, 2\pi], \beta \in [0, \pi], \gamma \in [0, 2\pi]\}$ , with each metric representing the map  $E \times E \rightarrow \mathbb{R}^+$ . To handle the periodic nature of angular variables, we must also define an angular difference function  $d(a, b) = \min\{|a - b|, 2\pi - |a - b|\}$ .

L2 Distance (range  $[0, \sqrt{3}\pi]$ ):

$$\Phi_2((\alpha_1, \beta_1, \gamma_1), (\alpha_2, \beta_2, \gamma_2)) = [d(\alpha_1, \alpha_2)^2 + d(\beta_1, \beta_2)^2 + d(\gamma_1, \gamma_2)^2]^{1/2} \quad (10)$$

L1 Distance (range  $[0, 3\pi]$ ):

$$\Phi_3((\alpha_1, \beta_1, \gamma_1), (\alpha_2, \beta_2, \gamma_2)) = d(\alpha_1, \alpha_2) + d(\beta_1, \beta_2) + d(\gamma_1, \gamma_2) \quad (11)$$

Mean Absolute Error (range  $[0, \pi]$ ):

$$\Phi_4((\alpha_1, \beta_1, \gamma_1), (\alpha_2, \beta_2, \gamma_2)) = \frac{1}{3} [d(\alpha_1, \alpha_2) + d(\beta_1, \beta_2) + d(\gamma_1, \gamma_2)] \quad (12)$$

To illustrate why the properties of continuity and uniqueness may be important when deciding what representation one would like to cast a deep learning problem into, consider Fig. 2 which plots a mean-squared error loss and two metrics over a number of training steps. This figure represents one example training run from Sect. 6.2 in which the output of the last network layer is a predicted SO(3) matrix.

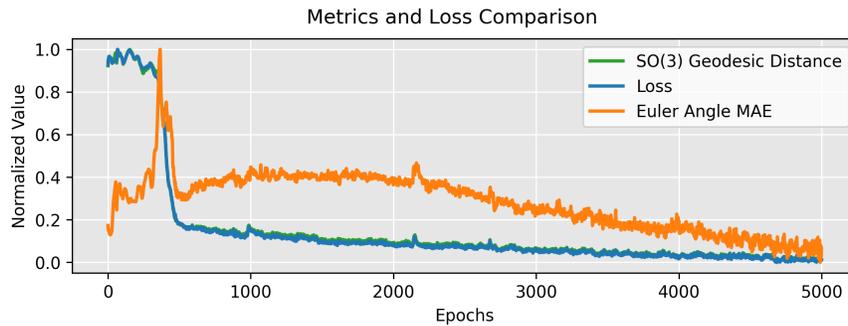


Fig. 2: Example training run evaluating loss (mean squared error), SO(3) geodesic distance, and Euler angle mean absolute error over a validation dataset. To solely illustrate the relative change in magnitudes, each series is normalized via min-max rescaling. While the MSE loss closely follows the ideal measure of rotation error, SO(3) geodesic distance, Euler angle MAE deviates significantly and exhibits pathological behavior at early epochs.

<sup>4</sup>From a topological perspective, this pathology arises due to there existing no local homeomorphism from the three-torus ( $T^3$ ) of Euler angles to the real projective space ( $RP^3$ ) of SO(3).

**DISTRIBUTION A. Approved for public release: distribution is unlimited.**  
**Public Affairs release approval #AFRL-2021-2863**

Here we observe that as the validation loss decreases, the SO(3) geodesic metric of (4) follows in lock-step, whereas the Euler angle MAE of (12) exhibits large fluctuations and, even more undesirable, opposes the change in direction of loss at early epochs. Recalling that SO(3) geodesic distance is the natural metric induced on SO(3) space and thus our ideal measure of similarity between labels and inferred predictions, this figure indicates that i) Euler Angle MAE can be an unreliable measure of error and ii) MSE appears to serve as an excellent loss function for minimizing SO(3) geodesic distance. While Euler angle metrics may give rise to erroneous behavior, one can observe that as the loss begins to converge, the MAE tends to also decrease approximately monotonically and shows reasonable limiting behavior for small Euler angles.

### 3.3 Quaternions

Analogous to how 2D rotations can be represented as complex numbers of unit norm ( $e^{i\theta}$ ), 3D rotations may be represented as four dimensional complex quantities of unit norm called unit quaternions. A quaternion  $q$  is defined as

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}, \quad (13)$$

where  $a, b, c, d$  are real numbers and  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  represent the generalization of complex numbers to higher dimension. In order to define a multiplication between quaternions, the complex ‘units’ obey

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1, \quad \mathbf{ijk} = -1. \quad (14)$$

To recover the three degrees of freedom necessary for describing rotation, we restrict ourselves to the set of unit quaternions which satisfy

$$q^\dagger q = 1 \quad (15)$$

$$\implies (a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k})(a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}) = a^2 + b^2 + c^2 + d^2 = 1. \quad (16)$$

Just as the SO(3) matrices equipped with a binary composition operation (matrix multiplication) form a group, the unit quaternions form a group under quaternion multiplication called SU(2). From a practical perspective, unit quaternions are particularly attractive because one only needs to specify four real numbers that satisfy (16) in order to describe a 3D rotation. Additionally, their mathematical structure is quite convenient as the set of unit quaternions are isomorphic to  $S^3$ , the three-sphere. With respect to rotations, the most important property of unit quaternions is that their group, SU(2) forms a double cover of SO(3). In other words, there is a 2:1 mapping between unit quaternions and members of SO(3). The degeneracy in the mapping is in fact quite simple: for every unit quaternion  $q$  there exists another unit quaternion  $-q$  that represents the same underlying SO(3) rotation.

To explore the continuity properties of the quaternion representation, let’s first define a map  $g$  from SO(3) to quaternions and a map  $f$  from quaternions to SO(3). One such choice of mappings is [23]

$$g_q(R) = \begin{cases} [R_{32} - R_{23}, R_{13} - R_{31}, R_{21} - R_{12}, Tr(R) + 1]^T & \text{if } Tr(R) - 1 \neq 0 \\ [\sqrt{R_{11} + 1}, c_2\sqrt{R_{22} + 1}, c_3\sqrt{R_{33} + 1}, 0]^T & \text{if } Tr(R) - 1 = 0 \end{cases} \quad (17)$$

$$c_i = \begin{cases} 1 & \text{if } R_{i,1} + R_{1,i} > 0 \\ -1 & \text{if } R_{i,1} + R_{1,i} < 0, \\ (\text{sgn}(R_{3,2}))^i & \text{otherwise} \end{cases} \quad (18)$$

$$f_q(q \in (x, y, z, w)) = \begin{pmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2zw & 2xz + 2yw \\ 2xy + 2zw & 1 - 2x^2 - 2z^2 & 2yz - 2xw \\ 2xz - 2yw & 2yz + 2xw & 1 - 2x^2 - 2y^2 \end{pmatrix}. \quad (19)$$

Now, consider the special case of a set of SO(3) elements that effect a rotation of  $\pi$  around an arbitrary axis, e.g. for a  $\pi$  rotation around the z-axis

$$\mathbf{R}_z = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (20)$$

**DISTRIBUTION A. Approved for public release: distribution is unlimited.  
Public Affairs release approval #AFRL-2021-2863**

Generically,  $SO(3)$  elements representing rotations of  $\pi$  are characterized by  $Tr(\mathbf{R}) = -1$  and  $R = R^{-1} = R^T$  (a rotation by  $\pi$  is equivalent to a rotation by  $-\pi$ ). One may then ask, is the set of quaternions that map to these  $SO(3)$  elements connected? Inspecting (17), we see that for arbitrary  $\pi$  rotations we have  $Tr(R) - 1 = 0$  and thus  $g_q(R)$  yields a quaternion  $q = (a \neq 0, b \neq 0, c \neq 0, 0)$ . Meanwhile, if we look at rotations close to  $\pi$  and evaluate their limit as they approach  $\pi$ , (17) yields a quaternion  $q$  with the first three components strictly vanishing by virtue of  $R$  being symmetric viz.  $R = R^{-1} = R^T$ ; that is,  $q = (0, 0, 0, 0)$ . Consequently, we see continuity of  $g_q$  does not hold as required in (1) and thus quaternions do not serve as a continuous representation of rotation.

Considering quaternions simply as unit vectors in  $\mathbb{R}^4$ , one can define an Euclidean inner product between two unit quaternions  $q_1 = (a_1, b_1, c_1, d_1)$  and  $q_2 = (a_2, b_2, c_2, d_2)$  as  $q_1 \cdot q_2 = a_1 a_2 + b_1 b_2 + c_1 c_2 + d_1 d_2$ . With an inner product in hand, we can then proceed to construct measures of distance between quaternions (and thus 3D rotations) in the same manner as usual real vector spaces, keeping in mind the 2:1 mapping of quaternions to  $SO(3)$ . One such metric is the L2 distance between quaternions

$$\Phi_5(q_1, q_2) = \min\{\|q_1 - q_2\|, \|q_1 + q_2\|\}. \quad (21)$$

The L2 norm  $\|\cdot\|$  is related the inner product as  $\|q\| = (q \cdot q)^{1/2}$  and we note that by construction  $\Phi_5(q, -q) = 0$ . Similarly, we can form a quaternion metric measuring the ‘angular’ cosine similarity of two vectors as

$$\Phi_6(q_1, q_2) = \cos^{-1}(|q_1 \cdot q_2|), \quad (22)$$

which is functionally related to  $\Phi_5$  through  $\Phi_6 = [2(1 - \cos(\Phi_5))]^{1/2}$ . When mapping quaternions to elements of  $SO(3)$ , we find that the quaternion metric  $\Phi_6$  of (22) is in fact equivalent to the  $SO(3)$  geodesic distance  $\Phi_0$  of (3) up to a constant viz.  $\Phi_0 = \frac{1}{2}\Phi_6$  [12].

### 3.4 Stiefel 6D

To motivate a representation that satisfies our definition (1) of continuity, namely a connected set of elements in  $SO(3)$  remains connected when mapped to a new representative space, let us first consider the mechanics involved in implementing an  $SO(3)$  regression task within a network. To form a 3x3 matrix as the output, we can construct a fully connected final layer in the network comprising 9 output units with linear activations. At this stage, the output matrix does not yet represent a valid rotation as it does not satisfy the ‘S’ and ‘O’ properties that define the  $SO(3)$  group. To impose orthogonality, i.e.  $\mathbf{R}\mathbf{R}^T = \mathbb{I}$ , we can arrange the 9 outputs into a 3x3 matrix and apply Gram-Schmidt orthogonalization to the column vectors. However, before carrying out the full orthogonalization, we note that one can actually satisfy both orthogonality and unit determinant simultaneously by using a slightly modified Gram-Schmidt procedure according to the map  $h_{GS} : \mathbb{R}^{3 \times 3} \rightarrow SO(3)$  defined as

$$h_{GS} \left( \left( \begin{array}{c|c|c} | & | & | \\ \hline a_1 & a_2 & a_3 \\ \hline | & | & | \end{array} \right) \right) = \left( \begin{array}{c|c|c} | & | & | \\ \hline b_1 & b_2 & b_3 \\ \hline | & | & | \end{array} \right), \quad (23)$$

where

$$b_i = \left( \begin{array}{c} N(a_1) \\ N(a_2 - (b_1 \cdot a_2)b_1) \\ b_1 \times b_2 \end{array} \right) \begin{array}{l} \text{if } i = 1 \\ \text{if } i = 2 \\ \text{if } i = 3 \end{array} \quad , \quad N(v_i) = \frac{v_i}{\|v_i\|}. \quad (24)$$

Here  $a_i$  represent the column vectors of the output units and  $b_i$  represent a set of orthonormal column vectors that, when formed into a matrix  $\mathbf{R}$ , satisfy  $\det(\mathbf{R}) = 1$ . The deviation from standard Gram-Schmidt orthogonalization occurs at the last step in forming  $b_3 = b_1 \times b_2$ , a product which ensures  $\det(\mathbf{R}) = 1$ .

Inspection of (24) reveals a curiosity: from the matrix of column vectors  $a_i$ , the final  $SO(3)$  matrix of column vectors  $b_i$  only depends on  $a_1$  and  $a_2$  – the elements of  $a_3$  are entirely discarded. With only 6 elements playing a role, this suggests that instead of mapping  $\mathbb{R}^{3 \times 3} \rightarrow SO(3)$ , a map of  $\mathbb{R}^{3 \times 2} \rightarrow SO(3)$  is entirely sufficient in defining a rotation representation.

With this insight, we can now formulate a new 6D rotation representation by defining the map  $f_{GS} : \mathbb{R}^{3 \times 2} \rightarrow SO(3)$  as

$$f_{GS} \left( \left( \begin{array}{c|c} | & | \\ \hline a_1 & a_2 \\ \hline | & | \end{array} \right) \right) = \left( \begin{array}{c|c|c} | & | & | \\ \hline b_1 & b_2 & b_3 \\ \hline | & | & | \end{array} \right), \quad b_i = \left( \begin{array}{c} N(a_1) \\ N(a_2 - (b_1 \cdot a_2)b_1) \\ b_1 \times b_2 \end{array} \right) \begin{array}{l} \text{if } i = 1 \\ \text{if } i = 2 \\ \text{if } i = 3 \end{array} \quad , \quad N(v_i) = \frac{v_i}{\|v_i\|}, \quad (25)$$

**DISTRIBUTION A. Approved for public release: distribution is unlimited.  
Public Affairs release approval #AFRL-2021-2863**

along with an inverse mapping  $g_{GS} : SO(3) \rightarrow \mathbb{R}^{3 \times 2}$  defined by

$$g_{GS} \left( \begin{pmatrix} | & | & | \\ a_1 & a_2 & a_3 \\ | & | & | \end{pmatrix} \right) = \begin{pmatrix} | & | \\ a_1 & a_2 \\ | & | \end{pmatrix}. \quad (26)$$

In constructing the map  $g_{GS}$  from elements of the fundamental  $SO(3)$  rotation representation to our 6D representation, (26) affirms that not only is  $g_{GS}$  continuous, it is remarkably straightforward and computationally efficient – simply drop the last column vector! Equipped with a continuous  $g_{GS}$ , [23] additionally details a proof showing that for every  $\mathbf{R} \in SO(3)$ ,  $f_{GS}(g_{GS}(\mathbf{R})) = \mathbf{R}$ , to thus establish the 6D representation as a truly continuous rotation representation according to our definition (1).

To expound upon some of the underpinnings of the 6D representation, we note that the intermediate representation spanned by the set  $g_{GS}(\mathbf{R})$  is the Stiefel manifold  $V_2(\mathbb{R}^3)$ , the set of all ordered pairs of orthonormal 3-vectors. We can view this manifold as a submanifold in 6D Euclidean space [5]. The ability of our representative manifold to be embedded in Euclidean space is not a special feature but in fact a generic property that must hold for any representation of rotation that is to be continuous by (1). Specifically, [23] points out that there exists no homeomorphism from  $SO(3)$  to any subset of  $\mathbb{R}^d$  for  $d < 5$ . Consequently, our 6D representation as formulated serves as the most compact<sup>5</sup> representation possible without loss of continuity.

With respect to measuring the distance between points  $\mathbf{V} \in V_2(\mathbb{R}^3)$  lying on the Stiefel manifold, one may use the inner product structure of the underlying Euclidean space to form the Euclidean metric [21]

$$\Phi_7(\mathbf{V}_1, \mathbf{V}_2) = Tr[(\mathbf{V}_1 - \mathbf{V}_2)^T(\mathbf{V}_1 - \mathbf{V}_2)] = 2Tr \left[ \mathbb{I} - \frac{1}{2}(\mathbf{V}_1^T \mathbf{V}_2 + \mathbf{V}_2^T \mathbf{V}_1) \right]. \quad (27)$$

Other metrics, such as the canonical and Procrustes distance may also be defined using more involved methods of differential geometry. [21, 11].

## 4. METHODS

With the rotation representations and properties of continuity, uniqueness, and measures of closeness having been explored, the natural next step in framing the problem of orientation estimation is to realize an implementation of a model that takes in spectra and regresses to orientation. In the context of deep learning, realizing an implementation amounts to choosing a rotation representation, formulating a loss function, selecting appropriate metrics, architecting a neural network, and finally training the network. In this section, we review the methods by which we instantiate a model tuned to the task of pose estimation.

### 4.1 Representation

With the target domain being a continuous space of rotations in a given representation, we formulate the problem as a regression task<sup>6</sup>. To select which representation to use, we use the basis of continuity and uniqueness to serve as a primary guiding principle. Support for continuous rotation representations is empirically demonstrated in [23], where it is shown that discontinuous representations can have up to 6-14 times higher mean errors in practical applications. Recalling that Euler angles and quaternions are both discontinuous and many-to-one representations of  $SO(3)$ , from the representations given in Sect. 3 we are then left with the choice between  $SO(3)$  and Stiefel 6D.

The simplest and most reasonable choice would seem to be  $SO(3)$  as it already graced as the fundamental representation of 3D rotation and has a naturally defined distance measure. However, as eluded to in Sect. 3, an oddity arises during its actual implementation within a neural network. After regressing to an output layer of nine units with linear activations and arranging into a 3x3 matrix, in order to constrain the matrix to unit determinant and enforce orthogonality (the ‘S’ and ‘O’ of  $SO(3)$ ), one must further employ a variant of the Gram-Schmidt orthogonalization procedure as defined by (24). Notably, only the first six output units are used within orthogonalization and thus a final  $SO(3)$  output is formed which has no dependence on the last three original output units. Hence, exactly three output units in

<sup>5</sup>A minimum of five dimensions are strictly necessary but with the formulation in six dimensions being computationally simpler and shown to yield better empirical performance [23], we only discuss the six dimensional case here.

<sup>6</sup>While it may appear obvious for orientation inference to be defined as a regression, other non-regressive techniques also exist such as continuous interval binning [9], effectively turning the inference task into a non-trivial classifier.

**DISTRIBUTION A. Approved for public release: distribution is unlimited.  
Public Affairs release approval #AFRL-2021-2863**

the final layer are made completely redundant and never receive weight updates from backpropagation of the gradient. This redundancy was in fact the very detail that motivated the original construction of the Stiefel 6D representation.

To detail our design of the final network layer, instead of forming an output layer of nine output units, we remove the extraneous inactive units and implement a layer of six output units in total. We then proceed with orthogonalization to generate the desired 3x3 SO(3) outputs<sup>7</sup>. Fascinatingly, this six degree of freedom construction as just described can be elevated to a bona-fide representation of rotation in its own right, with the map between the 6D elements and SO(3) defined by Gram-Schmidt-like orthogonalization and the (continuous) inverse map from SO(3) to 6D elements defined by dropping the last column vector.

If we view the neural network as a parameterized mapping from input images to output SO(3) matrices, we see that the SO(3) and 6D construction are actually functionally equivalent as the output has no dependence on the three extraneous units of the output layer. Thus, in the end, the distinction between SO(3) and the 6D representation is of no consequence at the implementation level; however, our decision to use an output layer of six units obliges us to proclaim the 6D Stiefel representation as the ideal choice of rotation representation.

## 4.2 Metrics and Loss

In discussing the choice of metrics and loss, we note that since orthogonalization is done in-network, the outputs of the model are elements in SO(3). Naturally, we are led to use the SO(3) geodesic distance as our metric, a choice supported by virtue of this distance function being the natural Riemannian metric induced on SO(3) space. While other possibly more intuitive metrics may be used, such as Euler angle MAE, we note that such metrics may exhibit undesirable pathological behavior, as shown in Fig. 2. For these reasons, we use the SO(3) geodesic distance implementation of (4) as our basis by which to measure performance, restated here as

$$\Phi_1(\mathbf{R}_1, \mathbf{R}_2) = \cos^{-1} \left[ \frac{1}{2} (\text{Tr}(\mathbf{R}_1 \mathbf{R}_2^T) - 1) \right]. \quad (28)$$

If we imagine the set of all SO(3) elements lying on the unit hemisphere  $S^3/\{\pm 1\}$ , then (28) has the convenient interpretation as measuring the angular difference (geodesic distance) between any two points on the unit hemisphere; namely,  $\Phi_1$  maps two rotations to the interval  $[0, \pi]$ , with two rotations being maximally separated at  $180^\circ$ . To provide an elementary, albeit very useful baseline, the SO(3) geodesic distance corresponding to a random regressor evaluates to  $126.75^\circ$ .

Concerning loss functions, we use the mean squared error defined for a dataset of size  $m$  as

$$MSE = \frac{1}{m} \sum_i \sum_j \| \mathbf{y}_{ij}^{pred} - \mathbf{y}_{ij}^{target} \|^2. \quad (29)$$

While one could use the SO(3) geodesic distance as a loss function, MSE by comparison avoids the trigonometric functions and matrix logarithms of (4) and (3). Given the unknown difficulty of the task of regressing spectra to orientations, our primary interest lies not in fine tuning performance but rather in providing the most stable training dynamics, a provision which MSE is better equipped to serve, given its smooth and well-behaved properties. In addition, we observe from Fig. 2 that MSE follows our base metric, SO(3) geodesic distance, nearly identically, further cementing its competence as the chosen loss function.

## 4.3 Architecture

As discussed in Sect. 5, the spectral input tensor takes one of two forms. In the case of 2D reflection spectra, the input tensor has shape (1340, 200) representing the pixel resolution of a single channel 16-bit image. Although an increasing number of advanced computer vision architectures may be adaptable to this input (with vision transformers gaining notable traction), for the purposes of establishing a fundamental baseline, we select our architecture on the basis of maximizing training stability. Naturally, we are led towards use of the ResNet architecture, where batch norm and skip connections have long been utilized to improve training speed and stabilize training dynamics [10].

To tune the ResNet architecture to the non-standard dimensions associated with the 2D spectra, we modify the initial convolutional layer by increasing the kernel size of  $7 \times 7$  to  $7 \times 28$  and initial stride of  $2 \times 2$  to  $2 \times 12$ . These modifications slightly reduce model capacity while significantly reducing GPU memory allocation, permitting larger input batch

<sup>7</sup>Importantly, these two steps detail the design of performing orthogonalization *in-network* rather than post-process. We expect an increase in error with any methods that do not backpropagate through the orthogonalization procedure.

**DISTRIBUTION A. Approved for public release: distribution is unlimited.  
Public Affairs release approval #AFRL-2021-2863**

sizes and hence increased training speed and stability. In agreement with [7], we find that a larger initial kernel had no impact on performance, suggesting that the most pertinent features are associated with wide spectral ranges. Following the same architecture depth and building block patterns of [10], we construct the full series of six model sizes ranging from 10 layers to 152 layers, and denote this set of width-modified models as ‘ResNet-NW’, N representing layer depth. We use [1] to compute the receptive field, observing that all ResNet-W models sans ResNet-10W have a receptive field greater than the input image size, affirming that our architectures achieve global context.

In the case of 1D reflection spectra, the input has tensor shape (2301, 1) representing 2301 (approximately even spaced) wavelength-flux pairs. By virtue of being single dimensional and spanning an appreciable range of wavelengths, an investigation of the architecture best suited to this data format lends itself well to the body of research concerning time series classification with deep neural networks. In particular, [22] analyzes time series classification performance of a variety of deep learning models over 44 distinct datasets and shows that the FCN and ResNet convolutional networks yield the highest performance. Encouraged by this study we also invoke the ResNet architecture for the 1D spectral input, allowing a convenient unification of architectures at the implementation level. Mirroring the 2D ResNet-W construction, we modify the initial kernel to size 28 and stride to size 12, and construct a series of six 1D ResNet-W models spanning from depths ResNet-10W to ResNet-152W.

#### 4.4 Training

Given that our experiments require the training of no less than twelve distinct models, and given our constrained computational resources, a systematic method of selecting the batch size, learning rate, and regularization which minimize training time while yielding performant results was deemed paramount. We fix batch size according to the largest possible batch that may fit in the GPU memory. Though smaller batch sizes may act a method of regularization, larger batch sizes result in smoother gradient updates improving training stability while significantly reducing training time. In order to fix the learning rate and regularization, we turned to the use of dynamic learning rate scheduling through the method of OneCycle [18] [19]. Although the OneCycle prescription as detailed in [18] appeared rather straightforward in theory, we found the practical implementation of OneCycle in our setting was accompanied by numerous challenges and important subtleties. We discuss our approaches and findings below.

As the first step, in order to construct a learning rate range test (LRRT) plot on a given model/dataset, one must estimate the appropriate range of learning rates to sweep. Without any apriori knowledge on what learning rates are most conducive, we constructed a large sweep spanning the range  $[10^{-9}, 10^{-1}]$ . However, to implement the LRRT, the number of training steps must also be specified. While the general guidance of [18] suggests that LRRT need only be carried out over a small number of epochs, we found that this prescription does not hold on models with rather large batch sizes. Rather, one should target the step count as the minimum number of steps required to partition the selected learning rate interval into bins of size large enough to avoid training instabilities induced by too rapid of a change in learning rate. Since we, once again, did not have any apriori guidance on what is a good learning rate bin size, we generated a series of LRRT curves with decreasing step counts to pinpoint the threshold step number at which a ‘phase change’ occurs from stable to unstable behavior.

While one could consider using a large overestimate of the number of steps to land one safely in the stable training regime, we have observed that the computational resources required to generate LRRT curves are non-trivial and, in models with very large batch sizes, are on par with the resources required for an actual training run. This matter is exacerbated when plotting the validation loss in the LRRT curve with the computational complexity scaling as  $O(n_{steps\_val})$ . The effort extolled in determining the minimal LR bin size is thus a worthy investment since, once it is fixed, a LRRT curve will be required for each choice of weight decay and thus LRRT’s will be great in number during the course of hyperparameter selection. In light of these computational burdens, we employed a number of techniques to reduce the resource use of the LRRT task.

Regarding the minimum LR bin size, we found through experiment that this size is approximately invariant to model capacity. Thus, for the ResNet-W architectures used in this paper, the subtask of ascertaining the minimum number of steps need not be repeated for each model, reducing computation by a factor of the number of models queued (six in our case). Moreover, this bin size may be readily used to calculate the minimum number of steps for any range of learning rates.

In a similar vein, we also found that the ideal window of learning rates is approximately the same over all model capacities, though we suggest padding the range by an additional order of magnitude on both endpoints to account for the small variability observed. To hone in on a candidate range of learning rates in the first place, we suggest

**DISTRIBUTION A. Approved for public release: distribution is unlimited.  
Public Affairs release approval #AFRL-2021-2863**

performing a large learning rate sweep without any validation steps, followed by an analysis of the LRRT curve to capture an optimal sub-range of learning rates. Once this sub-range is determined, validation loss can then be plotted in subsequent LRRT curves. We found that this ‘minimal window method’ reduced the number of steps of all subsequent LRRT curves from  $7.5 \times 10^3$  to  $3.1 \times 10^3$ , slicing computational complexity in half.

Returning the discussion to generating LRRT curves, at this point we have detailed our methods for fixing the number of training steps and the learning rate range. The last parameter that must be fixed in the LRRT is the number of validation steps. We consider the selection of this number equivalent to determining the minimum sample size of the validation dataset that approximately captures the underlying distribution. As such, this quantity will depend on the initial train-validation partitioning of the dataset. Absent practitioner judgement, one can also fix this number empirically by successively decreasing the number of validation steps until one arrives at the threshold of phase-change-like behavior, just as we did in minimizing the training steps.

Having specified all the parameters necessary to generate LRRT curves, we move the discussion to the next step within OneCycle, namely regularization. To fine tune the regularization, we implement the weight decay defined in [14], which has shown particular efficacy in conjunction with the Adam optimizer [13]. Using the OneCycle method, one should be able to fix the regularization by generating an LRRT curve for each value of weight decay and analyzing which values yield the ‘best’ interval of learning rates as plotted through the validation loss. Additionally, [18] suggests that the magnitude of the validation loss minimum can also be used as a basis to select a weight decay. In practice, we find the selection of the ‘best’ learning rate interval at different weight decays to be much more of an art than a science and becomes a rather nontrivial decision task. After much experimental tribulation, we arrived at a formulation of the task of selecting the optimal learning rate range as the optimization of three features: smoothness, slope, and extent. Even with this more refined basis, we note that in practice many LRRT curves could not be sufficiently ranked and thus required empirical comparison at the level of a completed training run.

We finish this section with some final remarks on our OneCycle implementation. Contrary to the official AdamW implementation of PyTorch, we do not scale the weight decay directly by the learning rate, but simply keep it fixed throughout the learning rate schedule. This is motivated by the insight of [19], which notes that large learning rates serve as a means of regularization in and of themselves, and thus we propose that if weight decay is to be made dynamic, it should be *inversely* proportional to the learning rate so as to keep the total effective regularization conserved. The final learning rate schedule itself is formed by a gluing of half cosines together and is specified by four parameters: max learning rate, warm-up percentage, starting learning, and final learning rate. We found that choosing the ‘best’ learning rate interval on the aforementioned basis resulted in selections of parameters that deviated significantly from the prescription of [18]. This could be attributed to the results of [18] lacking generalization to non-benchmark datasets or to the observation that our orientation task demanded significantly lower learning rates, with maximum values on the order of  $10^{-6}$ .

## 5. DATASETS

We design our datasets around space object scenarios of minimal complexity in order to i) provide an upper bound on the expected performance of CNN’s using spectral images and ii) form the simplest baseline possible to serve as a footing in which to add additional sources of noise and variation for possible future work. Each scenario consists of a target space object located at a telescope-object separation large enough to fall into the regime of optical non-resolvability. Relative to the 3.67 meter AOES telescope, the advanced optical system by which all our simulations are modeled, this amounts to positioning our space objects at a geosynchronous orbit (GEO) of radii  $r = 35768$  km. Space objects are fixed at a constant azimuth and elevation angle to lie directly overhead the ground-based optical system. The light source comprises parallel rays between the imaging system and target (an idealized ‘headlight’ illuminator), with a wavelength dependence mirroring the solar spectrum. We model three objects of varying bidirectional reflectance distribution functions (BRDFs) and moments of inertia as shown in Fig. 3.

**DISTRIBUTION A. Approved for public release: distribution is unlimited.  
Public Affairs release approval #AFRL-2021-2863**

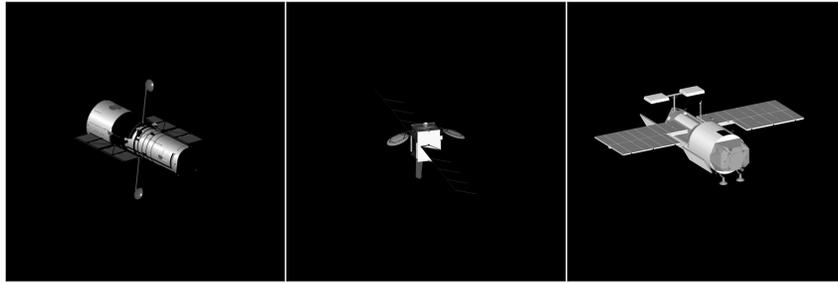


Fig. 3: Pristine 3D CAD renders of the space objects modeled within the datasets: Hubble telescope (left), Directv satellite (middle), and Almaz (right).

Using radiometric tools, 1D reflection spectra of shape (2301, 1) are generated for each space object, representing an integration of light spectra reflected towards our optical system over one spatial axis of a spectrograph (Fig. 4 (top)). To simulate the raw, uncalibrated frames as observed through a long slit spectrograph, the 1D reflection spectra are propagated through in house spectroscopic focal plane array (FPA) modeling software to output 16-bit single channel images of pixel resolution (200, 1340). Characterized by a horizontal strip of exposed pixels (Fig. 4 (bottom)), the 2D spectra are simulated with a signal-to-noise-ratio (SNR) of 100 and omit distortions induced by atmospheric turbulence.

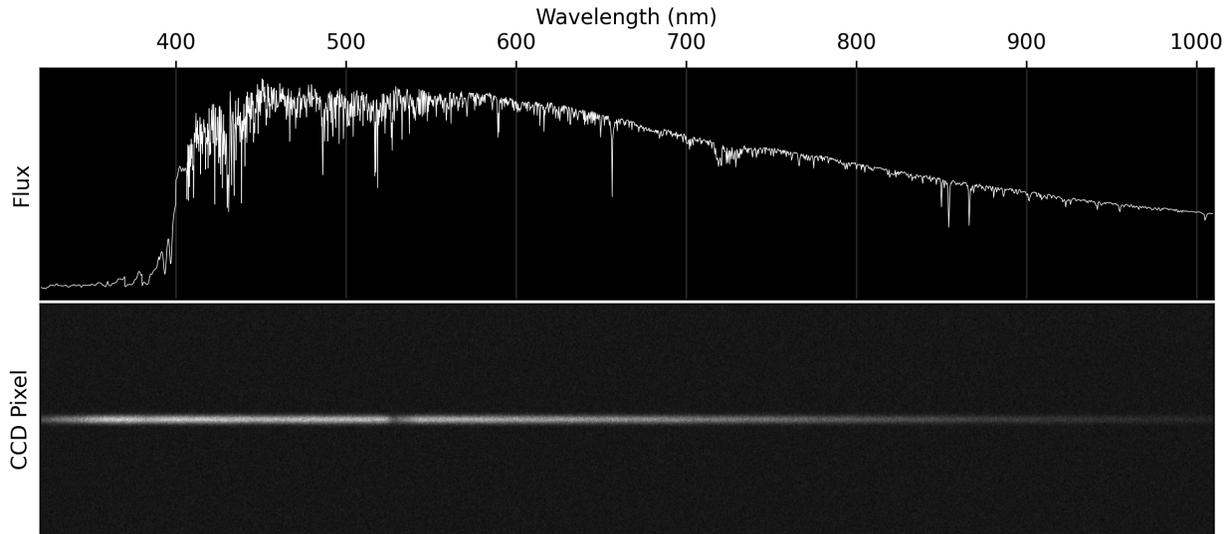


Fig. 4: Spectra simulated through an optical system with Strehl ratio 0.5 for space object Almaz residing at GEO ( $r = 35768$  km). (Top) 1D reflection spectrum. (Bottom) Raw long slit 2D reflection spectrum at 100 SNR.

We construct two dataset series representing 2D and 3D rotational motion. The 2D rotation task aims to establish the strength of correlation between spectral output and orientation, while the 3D rotation task aims to establish a core baseline measure of performance for future spectra-based applications of rotation. Each series comprises a set of 1D pristine spectra and 2D raw FPA spectra for each of the three space objects in Fig. 3. With the orientations of all space objects specified by Euler angles  $\alpha, \beta, \gamma$  (roll, pitch, yaw), the 2D rotation series is characterized by fixing  $\beta, \gamma = 0$  and uniformly sampling  $5 \times 10^3$  elements of roll, i.e.  $\alpha = U(-\pi, \pi)$ . For the 3D rotation series, we generate a dataset of  $5 \times 10^4$  random orientations by uniformly sampling over all Euler angles, i.e.  $\alpha, \gamma = U(-\pi, \pi)$  and  $\beta = U(-\pi/2, \pi/2)$ . In aggregate, this amounts to twelve distinct datasets.

**DISTRIBUTION A. Approved for public release: distribution is unlimited.**  
**Public Affairs release approval #AFRL-2021-2863**

## 6. EXPERIMENTS

### 6.1 2D Rotation

Space Object	Spectra (width, height)	Model	$N_{train}$	$N_{val}$	Validation $\Delta\theta^\circ$
Hubble	(2301, 1)	ResNet-18W	$4.5 \times 10^3$	$2.5 \times 10^2$	<b>15.48</b>
Directv-8	(2301, 1)	ResNet-18W	$4.5 \times 10^3$	$2.5 \times 10^2$	16.65
Almaz	(2301, 1)	ResNet-18W	$4.5 \times 10^3$	$2.5 \times 10^2$	47.52
Hubble	(1340, 200)	ResNet-18W	$4.5 \times 10^3$	$2.5 \times 10^2$	85.41
Directv-8	(1340, 200)	ResNet-18W	$4.5 \times 10^3$	$2.5 \times 10^2$	<b>80.17</b>
Almaz	(1340, 200)	ResNet-18W	$4.5 \times 10^3$	$2.5 \times 10^2$	81.97

Table 1: 2D rotation inference experiment using 1D and 2D spectra. Performance is measured by angular separation error  $\Delta\theta$  (cf. Appendix 9.2). Space objects are positioned at GEO ( $r = 35768$  km).

With the 2D rotation dataset series fixing yaw and pitch and sampling  $5 \times 10^3$  roll orientations, this task amounts to regressing to one angular degree of freedom,  $\theta$ . The results of the 2D rotation experiment are presented in Table 1. As detailed in Appendix 9.2, we regress to a single output unit with linear activation function, use L2 angular distance as the loss function, and measure validation performance using angular difference  $\Delta\theta$ . To provide a standard by which to compare performance, we note that a random regressor would yield a performance of  $\Delta\theta = 90^\circ$ . We observe that the 1D reflection spectra have a mean and standard error of  $26.55 \pm 14.84^\circ$ , far exceeding random performance and providing strong evidence that spectra as a physical mode of information does in fact possess enough variation with respect to orientation to be able to construct a sufficiently complete mapping. With the Hubble telescope being the strongest performer with an error of  $15.48^\circ$ , such a low value of error amounts to an excellent degree of localization of the spacecraft. As we repeat the same study with the more realistic 2D FPA spectra, we observe a systematic increase in error with a mean and standard error of  $82.52 \pm 2.17^\circ$ . The low standard deviation suggests that we may be probing a lower bound in performance error. Although we endure a significant increase in error moving from pristine to raw spectra, our performance is still moderately better than random. Lastly, we note that despite the complex structure inherent to spectra, the model capacity required to efficiently map spectra to  $\theta$  is not large at all, only requiring an 18 layer ResNet architecture.

### 6.2 3D Rotation

Space Object	Spectra (width, height)	Model	$N_{train}$	$N_{val}$	Validation $\Delta\theta^\circ$
Hubble	(2301, 1)	ResNet-18W	$4 \times 10^4$	$5 \times 10^3$	104.74
Directv-8	(2301, 1)	ResNet-18W	$4 \times 10^4$	$5 \times 10^3$	106.24
Almaz	(2301, 1)	ResNet-18W	$4 \times 10^4$	$5 \times 10^3$	<b>95.56</b>
Hubble	(1340, 200)	ResNet-152W	$4 \times 10^4$	$5 \times 10^3$	119.82
Directv-8	(1340, 200)	ResNet-152W	$4 \times 10^4$	$5 \times 10^3$	117.22
Almaz	(1340, 200)	ResNet-152W	$4 \times 10^4$	$5 \times 10^3$	<b>106.86</b>

Table 2: 3D rotation inference experiment using 1D and 2D spectra. Performance is measured by SO(3) geodesic distance error  $\Delta\theta$  (cf. eq. (4)). Space objects are positioned at GEO ( $r = 35768$  km).

Representing our objective task of orientation inference, we presents our results in Table 2. We measure validation performance through the SO(3) geodesic error, represented as the quantity  $\Delta\theta$ . To provide a standard by which to compare performance, we note that a random regressor of 3D rotation would yield a performance of  $\Delta\theta = 126.75^\circ$ . Across the 1D spectra, we observe that performance does improve an appreciable amount from random performance, though the geodesic errors are still high with mean and standard deviation  $102.18 \pm 4.72^\circ$ . As we move to the raw 2D spectra, we observe that we require a significantly larger model capacity in comparison to the 2D rotation task with raw spectra. Contrasted with the 2D rotation task, if we consider the complexity of the 3D rotation task as growing exponentially in the number of degrees of freedom, i.e. from one to three degrees of freedom, such an increase in network capacity is compulsory. We observe a systematic increase in error as we transition from the pristine spectra to the raw FPA spectra. In contrast the 2D rotation experiment, however, we see that the increase in this error is much

**DISTRIBUTION A. Approved for public release: distribution is unlimited.**  
**Public Affairs release approval #AFRL-2021-2863**

lower. We hypothesize that as the number of samples in the 2D rotation experiment is increased, the angular error associated with the raw spectra will begin to converge towards the pristine error, arriving a separation similar to what is observed in the 3D rotation experiment. If a valid hypothesis, this may suggest that this experiment represents the best lower bound on error given a dataset size of  $5 \times 10^4$  samples. Lastly, we note that even given the best performance of  $95.56^\circ$ , error must continue to be reduced by a sizable amount in order to provide sufficient localization of the space object in the space of 3D orientations.

## 7. SUMMARY

In order to build the necessary framework for establishing a baseline of performance in the task of orientation inference, one must first settle upon a representation of rotation. We establish  $SO(3)$  as the fundamental representation of rotation and analyze a set of candidate representations from the perspective of continuity, uniqueness, and associated measures of distance. Using continuity as our guiding principle, we are led to the  $SO(3)$  and Stiefel 6D representations. As we consider the implementation of these representations at the level of neural network architecture, we find that the  $SO(3)$  and Stiefel 6D instantiations are in fact equivalent up to a small number of extraneous neurons. With this equivalence in mind, we remove the redundancy and advance the Stiefel 6D representation as the optimal choice for deep learning applications.

With a representation in hand, we realize our implementation by using the mean-square-error loss function and  $SO(3)$  geodesic distance as our metric of performance. In light of the nonstandard pixel dimensions associated with our spectroscopic input images, we motivate the use of a modified ResNet architecture with an extended initial kernel and stride, which we refer to as ResNet-W. Concerning the output layer of our ResNet-W models, we explicitly detail the Gram-Schmidt-like orthogonalization process, a step paramount to ensuring our outputs represent valid 3D rotations. Having instantiated an implementation, we proceed to train our CNN's using the OneCycle [18] method of learning rate scheduling and regularization, where we discuss a number of practical techniques to reduce computational time and improve hyperparameter selection.

We then generate a set of datasets over three space objects categorized into 2D rotation and 3D rotation and further classified into pristine 1D spectra and raw 2D FPA spectra, yielding a total of 12 distinct datasets. Analyzing the results of our 2D rotation experiment, we observe that inference using pristine spectra yields a promising degree of localization of the space object, with the best model confining the angular range to an error of  $\Delta\theta = 15.48^\circ$ . Importantly, barring the effects of additional sources of noise and variation, this result establishes that spectra do in fact possess enough structure correlated to rotation to produce a sufficiently complete mapping from spectra to orientation. Upon using the realistic 2D FPA spectra as input, we observe an appreciable increase in systematic error, with a best localization error of  $\Delta\theta = 80.17^\circ$ . For our full task of 3D orientation estimation, we observe an improvement in performance relative to a random regressor for both pristine and FPA spectra. However, to reach localizations adequate to inform most targeted decisions of operators and analysts downstream, the average  $SO(3)$  geodesic error associated with the 3D rotation experiment will need appreciable improvement relative to its current standing. With the 2D rotation and 3D rotation datasets having been designed around scenario of minimum complexity, our experiments give an estimate on the upper bound of performance expected in mapping spectra to orientation. In addition, we perceive these experiments to serve as a practical baseline by which future applications of spectroscopy and rotation can build upon as more realistic sources of variation as added.

One potential extension of future work with high utility is pointing angle estimation. This task's performance measures are easily visualized, and, with the pointing angle being defined by two degrees of freedom (e.g. the two angles of a vector tracing the unit sphere  $S^2$ ), we anticipate the complexity of this problem to lie in between 2D orientation (one degree of freedom) and 3D orientation (three degrees of freedom). Similarly, we anticipate the application of spectra to space object maneuver classification to be a task with less complexity than 3D rotation and possibly 2D rotation. With respect to improving our current baseline, an exploration of 3D rotation datasets with larger sample sizes ( $n = 10^5 - 10^6$ ) may be also be a fruitful endeavor as the saturation of performance with respect to sample size has yet to be probed.

**DISTRIBUTION A. Approved for public release: distribution is unlimited.  
Public Affairs release approval #AFRL-2021-2863**

## 8. REFERENCES

- [1] André Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 4(11):e21, 2019.
- [2] G. Badura, C. Valenta, and B. Gunter. Convolutional Neural Networks for Inference of Space Object Attitude Status. *AMOS Conference Proceedings*, 2020.
- [3] Gregory Badura, Christopher Valenta, Brian Gunter, and Brian Shoffeitt. Multi-scale convolutional neural networks for inference of space object attitude status from detrended geostationary light curves. 02 2021.
- [4] Kevin Black, Shrivu Shankar, Daniel Fonseka, Jacob Deutsch, Abhimanyu Dhir, and Maruthi R Akella. Real-time, flight-ready, non-cooperative spacecraft pose estimation using monocular imagery. *arXiv preprint arXiv:2101.09553*, 2021.
- [5] Darshan Bryner. Endpoint geodesics on the stiefel manifold embedded in euclidean space. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1139–1159, 2017.
- [6] Justin Fletcher, Ian McQuaid, Peter Thomas, Jeremiah Sanders, and Greg Martin. Feature-based satellite detection using convolutional neural networks. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, 2019.
- [7] J. Zachary Gazak, Ian McQuaid, Justin Fletcher, Thomas Swindle, and Matthew Phelps. SpectraNet: Learned Recognition of Artificial Satellites from High Contrast Spectroscopic Imagery. *WACV*, Submitted, 2022.
- [8] Z. Gazak, J. Fletcher, T. Swindle, and I. McQuaid. Exploiting Spatial Information in Raw Spectroscopic Imagery using Convolutional Neural Networks. *AMOS Conference Proceedings*, 2020.
- [9] Kota Hara, Raviteja Vemulapalli, and Rama Chellappa. Designing deep convolutional neural networks for continuous object orientation estimation. *arXiv preprint arXiv:1702.01499*, 2017.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [11] Knut Hüper and Florian Ullrich. Real stiefel manifolds: An extrinsic point of view. In *2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)*, pages 13–18. IEEE, 2018.
- [12] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.
- [13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [15] J. Lucas, T. Kyono, M. Werth, N. Gagnier, Z. Endsley, J. Fletcher, and I. McQuaid. Estimating Satellite Orientation through Turbulence with Deep Learning. *AMOS Conference Proceedings*, 2020.
- [16] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):607–626, 2008.
- [17] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Learning 3-d object orientation from images. In *2009 IEEE international conference on robotics and automation*, pages 794–800. IEEE, 2009.
- [18] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.
- [19] Leslie N. Smith and Nicholay Topin. Super-convergence: very fast training of neural networks using large learning rates. In Tien Pham, editor, *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, pages 369 – 386. International Society for Optics and Photonics, SPIE, 2019.
- [20] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018.
- [21] Pavan Turaga, Ashok Veeraraghavan, and Rama Chellappa. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [22] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.
- [23] 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/CVF Conference on Computer Vision and Pattern Recognition*,

**DISTRIBUTION A. Approved for public release: distribution is unlimited.**  
**Public Affairs release approval #AFRL-2021-2863**

## 9. APPENDIX

### 9.1 SO(3) Geodesic Equivalence

By the orthogonality relation, any SO(3) matrix  $\mathbf{R}$  must obey  $\mathbf{R}\mathbf{R}^T = \mathbb{I}$ , in which it follows that  $\mathbf{R}^T = \mathbf{R}^{-1}$ . Thus  $\mathbf{R}^T$  itself is an element of SO(3) via the group invertability axiom. By extension, the product  $\mathbf{R}_1\mathbf{R}_2^T$  also lies in SO(3) and thereby may be expressed as an exponential of the underlying Lie algebra generators

$$\mathbf{R} = \mathbf{R}_1\mathbf{R}_2^T = \exp(\theta\mathbf{K}), \quad (30)$$

where  $\mathbf{K} \in so(3)$ , the generators of SO(3), are skew-symmetric 3x3 matrices (i.e.,  $\mathbf{K}^T = -\mathbf{K}$ ) and  $\theta$  represents the counter-clockwise angle of rotation about an axis  $\mathbf{k}$ . The unit vector  $\mathbf{k}$  is related to the rotation generators by

$$\mathbf{K} = \begin{pmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{pmatrix}, \quad (31)$$

whereby the Frobenius norm  $\|\mathbf{K}\|_F$  may be readily calculated as

$$\|\mathbf{K}\|_F = (2\mathbf{k}^2)^{1/2} = \sqrt{2}. \quad (32)$$

Now, defining  $\mathbf{R} = \mathbf{R}_1\mathbf{R}_2^T$ , we can express (3) as

$$\Phi_0 = \|\log(\mathbf{R})\|_F = \|\log(\exp(\theta\mathbf{K}))\|_F = |\theta|\|\mathbf{K}\|_F = \sqrt{2}|\theta|. \quad (33)$$

Finally, via the angle axis representation one may take the trace of the rotation matrix to obtain the angle of rotation

$$|\theta| = \cos^{-1} \left[ \frac{1}{2}(\text{Tr}(\mathbf{R}) - 1) \right] = \Phi_1, \quad (34)$$

and thus we may establish the relation of  $\Phi_0$  to  $\Phi_1$  as

$$\Phi_0 = \sqrt{2}\Phi_1. \quad (35)$$

### 9.2 2D Rotation

To formulate the problem of 2D rotation, we swap the 6D output layer defined by the architecture in Sect. 4.3 with a single linear activation unit, to thus cast the 2D rotation task into learning a map from input spectra to the continuous valued  $\theta_{pred}$ . We then define our loss function over a dataset of size  $m$  as the L2 angular distance

$$\mathcal{L}_{L2,ang} = \frac{1}{m} \sum_i \frac{1}{2} \left[ \left( \cos \theta_{target}^{(i)} - \cos \theta_{pred}^{(i)} \right)^2 + \left( \sin \theta_{target}^{(i)} - \sin \theta_{pred}^{(i)} \right)^2 \right]. \quad (36)$$

To measure the performance, we simply use the angular difference defined by

$$\Delta\theta = \min\{|\theta_{target} - \theta_{pred}|, 2\pi - |\theta_{target} - \theta_{pred}|\}. \quad (37)$$

While other 2D representations and associated loss functions may have more desirable properties in terms of continuity (as defined in (1)) and uniqueness, counter logical to the 3D methodology the authors found (36) to empirically yield the best training stability and performance. Nonetheless, we state other candidate 2D rotation loss functions below for reference.

**DISTRIBUTION A. Approved for public release: distribution is unlimited.  
Public Affairs release approval #AFRL-2021-2863**

Let us define a regularization constant  $\alpha$  and choose the principle branch of  $\text{atan2}$  such that  $\text{atan2} : \mathbb{R}^2 \rightarrow [-\pi, \pi)$ . First, we provide loss functions appropriate for a network output layer comprising a single output linear unit  $\theta_{pred}$ :

$$\mathcal{L} = \frac{1}{2} [1 - \cos(\text{atan2}(\sin \theta_{pred}, \cos \theta_{pred}) - \theta_{target})]$$

$$\mathcal{L} = \frac{1}{2} (\text{atan2}(\sin \theta_{pred}, \cos \theta_{pred}) - \theta_{target})^2$$

$$\mathcal{L} = \frac{1}{2} (\theta_{pred} - \theta_{target})^2$$

$$\mathcal{L} = \min\{|\theta_{target} - \theta_{pred}|, 2\pi - |\theta_{target} - \theta_{pred}|\}.$$

For a network output layer consisting of two linear units  $(x_{pred}, y_{pred})$ , potential loss functions include:

$$\mathcal{L} = \frac{1}{2} [(x_{pred} - \cos \theta_{target})^2 + (y_{pred} - \sin \theta_{target})^2] + \alpha [1 - (x_{pred}^2 + y_{pred}^2)]$$

$$\mathcal{L} = \frac{1}{2} \left[ \left( \frac{x_{pred}}{\sqrt{x_{pred}^2 + y_{pred}^2}} - \cos \theta_{target} \right)^2 + \left( \frac{y_{pred}}{\sqrt{x_{pred}^2 + y_{pred}^2}} - \sin \theta_{target} \right)^2 \right]$$

$$\mathcal{L} = \frac{1}{2} \min\{|\text{atan2}(y_{pred}, x_{pred}) - \theta_{target}|, 2\pi - |\text{atan2}(y_{pred}, x_{pred}) - \theta_{target}|\}$$

$$\mathcal{L} = \frac{1}{2} [1 - \cos(\theta_{target} - \arctan2(y_{pred}, x_{pred}))].$$

**DISTRIBUTION A. Approved for public release: distribution is unlimited.**  
**Public Affairs release approval #AFRL-2021-2863**