Physics-Guided Machine Learning for Satellite Spin Property Estimation from Light Curves

Gregory P. Badura, Christopher R. Valenta
Georgia Tech Research Institute (GTRI), Electro-Optical Systems Laboratory (EOSL)

ABSTRACT

Knowledge of the spin state of space objects is critical for effectively planning operations such as collision avoidance and debris removal. One such passive method for assessing the spin rate and spin-axis of debris is through the use of passive brightness measurements known as "light curves." Astronomers have derived physics-based algorithms for retrieving spin state via light curve analysis. These algorithms convert the relative spin state into an inertial spin state by accounting for the motions of the observation telescope, the space object, and the sun. A major downside of these theories, however, is that the resulting cost functions for operational deployment are highly non-linear.

The intractable nature of the spin state estimation problem opens the door for solution via Machine Learning (ML) models. Typical "black box" ML algorithms do not rely on scientific theory, but rather are trained on large data-bases to learn how to solve a task in a manner that is obscured from the operator. While ML models can be effective for making predictions that out-perform human-derived algorithms, they also have the potential to derive solutions that either violate known physical constraints or are non-generalizable to new data instances. This is in particular a concern for many Space Domain Awareness (SDA) problems that are rooted in the physical theory of the motions of orbiting bodies.

To overcome the limitations of both physical theory and "black box" ML models for spin state retrieval, we leverage a hybrid approach: the physics-guided ML model. This concept uses a physics-based loss function in the learning objective of the ML model in order to guide the model towards making predictions that not only exhibit low prediction error with respect to training data but are also physically consistent with astronomer-derived theories.

Towards this end, we introduce a new physically derived equation for relating the inertial spin state to observations of relative spin rates. We then show that this equation can be used as a loss function for training ML models. We present a time-variant ML model for the retrieval of spin state that substantially outperforms both randomized numerical optimization approaches as well as temporally-invariant ML methods such as Convolutional Neural Networks (CNNs). Finally, we provide initial evidence that training of the time-variant ML model with our physics-based loss function is more stable and generalizes more effectively to unseen (i.e. "out-of-distribution") data instances. We believe that this paper provides promising avenues for merging big-data ML approaches with the robust physical theory of the SDA field.

1. INTRODUCTION

Due to the ever-increasing rate of satellite launches [1] and the potential for satellite failure [2], the population of space debris is continuously growing. Space debris in Low Earth Orbit (LEO) and Medium Earth Orbit (MEO), in particular, pose a major risk to functioning satellites due to the potential for collisions and fragmentation [3, 4]. Because of these dangers, it is critical to maintain attitude knowledge of debris at all times. Specifically, knowledge of the inertial spin-axis and spin rate of debris is necessary for orbital propagation [5], planning of collision avoidance maneuvers [6], performing remote satellite diagnostics [7], and preparing for debris removal missions [8, 9].

One method for assessing debris spin state is through the use of light curves. Astronomers have derived physical theories that explain how to extract spin-axis orientation and rotation rate from light curves [10–12]. These theories, however, result in optimization cost functions that are computationally expensive to solve and prone to local minima [5]. To elaborate, there are two important definitions to consider when explaining the rotation of debris: "sidereal," and "synodic" [5]. The synodic period is the amount of time that it takes for a spinning object to make a full revolution

relative to an Earth-based telescope. The sidereal period, on the other hand, is the amount of time that it takes the object to fully rotate about its spin-axis relative to the fixed orientation of the stars [10]. In short, the synodic spin rate is the rotation rate within a *relative* frame while the sidereal spin rate is the rotation rate within an *inertial* frame.

Astronomer-derived techniques for converting synodic into sidereal spin rate via light curve analysis require accounting for the relative motions of the observation telescope, the satellite, and the Sun [11, 12]. Although such algorithms were originally developed for asteroid characterization, they are applicable to debris observations because they are independent of the shape and reflectivity characteristics of the observed object [13]. While straightforward in theory, researchers have shown that this is a highly non-linear problem that is compounded by issues that will be discussed such as "spin-axis ambiguity" (see Figure 1 (b)). As we will show, the spin-axis retrieval problem's non-linearity makes it nearly impossible to solve via randomized numerical optimization. This leads to the conclusion that novel Machine Learning (ML) paradigms may lead to more accurate retrieval. Numerous studies have already shown the potential of ML algorithms to classify satellite properties such as attitude state, orientation, and optical cross section from light curve datasets [8, 14–20], making them a viable candidate to apply the spin retrieval problem.

The most common complaint about ML algorithms, however, is that they operate as "black boxes" that obscure their logic from end-users. These "black boxes" have been shown to lead to models that are (1) not generalizable to new data, (2) not consistent with known physical principles, and (3) require enormous amounts of data to train [21]. In the case of spin parameter retrieval for which known physical constraints exist and limited experimental data may be available, these are major problems for creating deployable ML tools. A new approach that has been suggested to overcome such issues for scientific problems is the use of physics regularization for ML models [21, 22]. The goal of these models is to use scientific knowledge as physics-based loss functions in the learning objective. As we will discuss, by incorporating a novel spin parameter retrieval equation in our ML model's learning objective, we obtain models that (1) potentially train faster, (2) lead to more generalizable predictions on unseen data, and (3) lead to more stable training across epochs.

This paper makes several contributions to advance both the characterization of debris spin properties and the use of deployable ML models in the Space Domain Awareness (SDA) community:

1. We present a new rigorously derived cost function for retrieving the sidereal spin rate and inertial spin-axis from observations of synodic spin rate. We provide evidence that this cost function is smoother than previous cost functions employed for spin parameter retrieval. We also provide methods for incorporating this loss function in physics-guided ML models.

2. We present a new sequential encoder model for the retrieval of both the linearly dependent sidereal spin rate and non-linearly dependent spin-axis from synodic period measurements. We provide evidence that this model outperforms both randomized numerical optimization approaches as well as temporally-invariant ML methods such as Convolutional Neural Networks (CNNs).

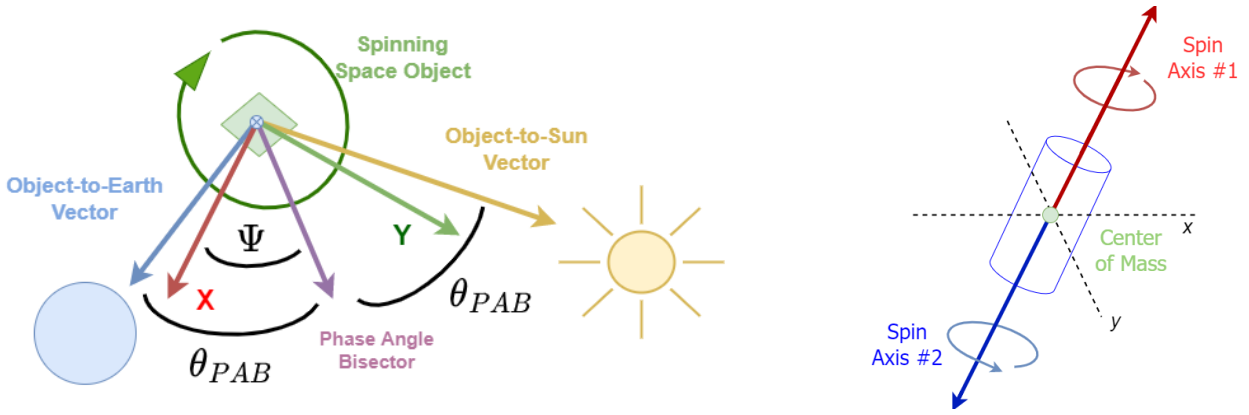
3. We provide evidence that physics-guided ML models can be both faster to train as well as more generalizable to unseen (i.e. "out-of-distribution") observations. This provides a potential pathway to fuse the numerous physical theories of the astronomy community with big-data ML approaches.

This paper proceeds as follows. In Section 2, we present the theory behind retrieval of inertial spin parameters and introduce a novel loss function that is smoother for numerical optimization approaches. In Section 3, we introduce the concept of physics-guided ML and give an overview of how we utilized our novel cost function during ML training. In Section 4, we introduce both the methods for simulation of datasets as well as the ML frameworks that were tested. Finally, in Section 5 we provide our results along with future directions of physics-guided ML in the SDA field.

2. THEORY: RETRIEVING SIDEREAL SPIN STATE FROM SYNODIC SPIN MEASUREMENTS

2.1 The Epoch Method

The "Epoch Method" approach that is used in this paper utilizes temporal changes in synodic period that have been extracted from a light curve to infer both inertial spin-axis orientation and sidereal rotation rate [11]. This method relies on the relationship between the synodic period and the Phase Angle Bisector's (PAB) motion to explain the time-evolving difference between synodic and sidereal spin rates. For reference, the PAB is the direction midway



(a) (b)

Fig. 1: (a) The concept of the Phase Angle Bisector (PAB) and its relationship to the spin-axis of an object: the spin-axis of the object points out of the page (blue $\otimes$) and Ψ denotes the azimuthal angle of the PAB along the plane perpendicular to the spin-axis (i.e. X/Y plane). (b) The spin-axis ambiguity problem that arises: a cylinder (i.e. rocket body) can spin clockwise about one of two mirrored axes (red & blue vectors) and generate the same perceived spin rate from the perspective of a ground-based observer.

between the body-to-observer and body-to-Sun unit vectors (purple vector in Fig. 1). The success of this method is directly tied to the fact that over short time scales, the passive brightness of a spinning body is controlled primarily by the position of the PAB with respect to the body's spin-axis [13]. The PAB vector's position and velocity is directly observable from Two-Line Element (TLE) data, allowing this information to be directly computed for use in machine learning and regression tasks.

The method directly relies on the orientation and position of the PAB with respect to the body's spin-axis, which can be expressed via the following equation:

$$\mathbf{b_I} = \frac{\mathbf{o_I} + \mathbf{s_I}}{||\mathbf{o_I} + \mathbf{s_I}||}, \tag{1}$$

where $\mathbf{o_I}$ and $\mathbf{s_I}$ denote the body-to-observer and body-to-sun unit vectors, respectively, and the subscript $\mathbf{I}$ denotes that the vectors are defined in the Earth-Centered Inertial (ECI) frame. Note that these vectors are implied to be time-dependent unit vectors, but that the time (t) notation (i.e. $\mathbf{b_I}(t)$) is excluded for brevity.

The algorithm assumes that the inertial-frame PAB vector can be rotationally aligned into a "spin-axis reference frame" that has it's z-axis oriented along the axis that the object is spinning, and x/y-plane that is fixed in the ECI frame per classical mechanics theorems [23]. The rotation alignment of the inertial-frame PAB into the spin-axis reference frame (denoted by sub-script $\mathbf{S}$) is then written as a function of two Euler angles [13]:

$$\mathbf{b_S} = \mathbf{R}(\theta, \phi)\mathbf{b_I} \tag{2}$$

$$\mathbf{R}(\theta, \phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

After performing the alignment into the spin-axis reference frame, the unit vector $\mathbf{b_S}$ is then decomposed into azimuthal (Ψ) and axial (Θ) components along the frame's axes:

$$\mathbf{b_S}^T = [\cos\Psi\sin\Theta, \sin\Psi\sin\Theta, \cos\Theta] \tag{4}$$

The azimuthal term can be written in the explicit form, $\Psi(t, \theta, \phi)$, that acknowledges the dependence on both time and the Euler angles required to align it to the spin-axis frame. We note this explicit form here in order to highlight that

the method assumes that azimuthal motion of the PAB (i.e. temporal changes in $\Psi(t,\theta,\phi)$) accounts for the perceived difference between synodic and sidereal frequencies over time [11]. This leads to the loss expression for the expected synodic frequency (ω') at a particular time given the temporally constant sidereal frequency (Ω) and the spin-axis Euler angles (θ,ϕ) [13]:

$$\omega'(t,\Omega,\theta,\phi) = \Omega - \frac{\partial\Psi(t,\theta,\phi)}{\partial t} \tag{5}$$

From this expression, we observe that there are two primary observed parameters (t, ω), and three parameters for which we must solve (Ω, θ, ϕ). This makes the problem under-determined for a single timestep. Because of this, the task of inferring an object's spin state from measured synodic frequencies is commonly formulated as a least-squares problem across a time-series of synodic frequency measurements. In other words, given a time-series of measured synodic frequencies (t_i, ω_i) the goodness-of-fit indicator that must be minimized via gradient descent approaches is written as [5, 8, 13]:

$$\chi^2(\Omega,\theta,\phi) = \left(\frac{1}{N}\right)\sum_i^N \left(\omega_i - \omega_i'\right)^2 = \left(\frac{1}{N}\right)\sum_i^N \left(\omega_i - \Omega + \left.\frac{\partial\Psi(t,\theta,\phi)}{\partial t}\right|_{t_i}\right)^2 \tag{6}$$

Many researchers who have deployed this algorithm [5, 11, 13] have approximated the term $(\partial\Psi(t,\theta,\phi)/\partial t)|_{t=t_i}$ using numerical differentiation approaches. In other words, they assume that the expression can be averaged over a short enough time window ($t_i - \frac{w}{2} \leq t \leq t_i + \frac{w}{2}$) such that the following holds [13]:

$$\left.\frac{\partial\Psi(t,\theta,\phi)}{\partial t}\right|_{t_i} \approx \frac{\Psi(t_i - \frac{w}{2},\theta,\phi) - \Psi(t_i + \frac{w}{2},\theta,\phi)}{2} \tag{7}$$

This approximation poses several such issues for both practical deployment and our physics-guided machine learning approach. One such issue is that the function $\partial\Psi(t,\theta,\phi)/\partial t$ is highly non-linear such that the error in this approximation can grow unexpectedly when traversing the parameter space of (Ω,θ,ϕ), as has been discussed by Zhao et al [5]. Another issue is that there are no clear guidelines on what time window size (w) is acceptable for the approximation. A final critical issue for our proposed method is that this approximation obscures the effect of known TLE information on the observed synodic frequencies. For these reasons, we derive a more exact form of the azimuthal velocity that can readily be ingested by our physics-guided machine learning models and leads to more stable convergence when solving for spin state parameters.

2.2 Exact Form of the Phase Angle Bisector Azimuthal Velocity

We begin by noting from Equation 4 and using a trigonometric identity of $\sin(\arccos x) = \sqrt{1-x^2}$, that the azimuthal component of the PAB vector in the spin-axis frame can be written as:

$$\Psi(t,\theta,\phi) = \arccos\left(\frac{b_x^S}{\sqrt{1-b_z^{S2}}}\right), \tag{8}$$

where the super-script S denotes that we are in the spin-axis reference frame and the sub-script denotes the axis.

We then obtain from Equations 2 and 3 that the unknown terms b_x^S, b_y^S, and b_z^S can be re-written in terms of the known TLE- and Ephemeris-derived vector $\mathbf{b_I}^T = [b_x^I, b_y^I, b_z^I]$:

$$\mathbf{b_S} = \begin{pmatrix} b_x^I\cos\phi + b_y^I\sin\phi \\ -b_x^I\sin\phi\cos\theta + b_y^I\cos\phi\cos\theta + b_z^I\sin\theta \\ b_x^I\sin\phi\sin\theta - b_y^I\cos\phi\sin\theta + b_z^I\cos\theta \end{pmatrix} \tag{9}$$

Plugging Equation 9 into Equation 8 then yields:

$$\Psi(t,\theta,\phi) = \arccos\left(\frac{b_x^I\cos\phi + b_y^I\sin\phi}{\sqrt{1-\left(b_x^I\sin\phi\sin\theta - b_y^I\cos\phi\sin\theta + b_z^I\cos\theta\right)^2}}\right) \tag{10}$$

We then observe that the derivative with respect to time can be found by repeated application of the "Chain Rule" [24]:

$$\frac{\partial\Psi(t,\theta,\phi)}{\partial t} = \frac{\partial\Psi}{\partial b_x^I}\frac{\partial b_x^I}{\partial t} + \frac{\partial\Psi}{\partial b_y^I}\frac{\partial b_y^I}{\partial t} + \frac{\partial\Psi}{\partial b_z^I}\frac{\partial b_z^I}{\partial t} \tag{11}$$

The vector $\dot{\mathbf{b}}^I = \partial\mathbf{b}^\mathbf{I}/\partial t$ can be directly computed by using known TLE and ephemeris information, as is shown in Appendix A. The other three scalar components must be derived by using a rigorous derivation. For the sake of focus, this derivation is fully outlined in Appendix B. As a result of this derivation, we obtain the following solution for the required terms by plugging Equations 35, 36, 53, 54 and 55 into Equation 11:

$$\frac{\partial\Psi(t,\theta,\phi)}{\partial t} = \left(\frac{-1}{ab}\right)\left(c\left(\frac{\partial b_x^I}{\partial t}\right) + d\left(\frac{\partial b_y^I}{\partial t}\right) + e\left(\frac{\partial b_z^I}{\partial t}\right)\right) \tag{12}$$

where for the sake of simplification, we have defined the following variables:

$$a = \left(1 - \left(\mathbf{b}^{\mathbf{I}^T}\mathbf{d}\right)^2 - \left(\mathbf{b}^{\mathbf{I}^T}\mathbf{c}\right)^2\right)^{1/2} \tag{13}$$

$$b = \left(1 - \left(\mathbf{b}^{\mathbf{I}^T}\mathbf{d}\right)^2\right) \tag{14}$$

$$c = \left(\hat{\mathbf{x}}^T\mathbf{c}\right)\left(1 - \left(\mathbf{b}^{\mathbf{I}^T}\mathbf{d}\right)^2\right) + \left(\mathbf{b}^{\mathbf{I}^T}\mathbf{c}\right)\left(\hat{\mathbf{x}}^T\mathbf{d}\right)\left(\mathbf{d}^T\mathbf{b}^\mathbf{I}\right) \tag{15}$$

$$d = \left(\hat{\mathbf{y}}^T\mathbf{c}\right)\left(1 - \left(\mathbf{b}^{\mathbf{I}^T}\mathbf{d}\right)^2\right) + \left(\mathbf{b}^{\mathbf{I}^T}\mathbf{c}\right)\left(\hat{\mathbf{y}}^T\mathbf{d}\right)\left(\mathbf{d}^T\mathbf{b}^\mathbf{I}\right) \tag{16}$$

$$e = \left(\mathbf{b}^{\mathbf{I}^T}\mathbf{c}\right)\left(\hat{\mathbf{z}}^T\mathbf{d}\right)\left(\mathbf{d}^T\mathbf{b}^\mathbf{I}\right) \tag{17}$$

We also clarify here that **c** and **d** are time-independent vectors that are functions of the spin-axis Euler angles. These variables are defined in Equations 35 and 36, respectively.

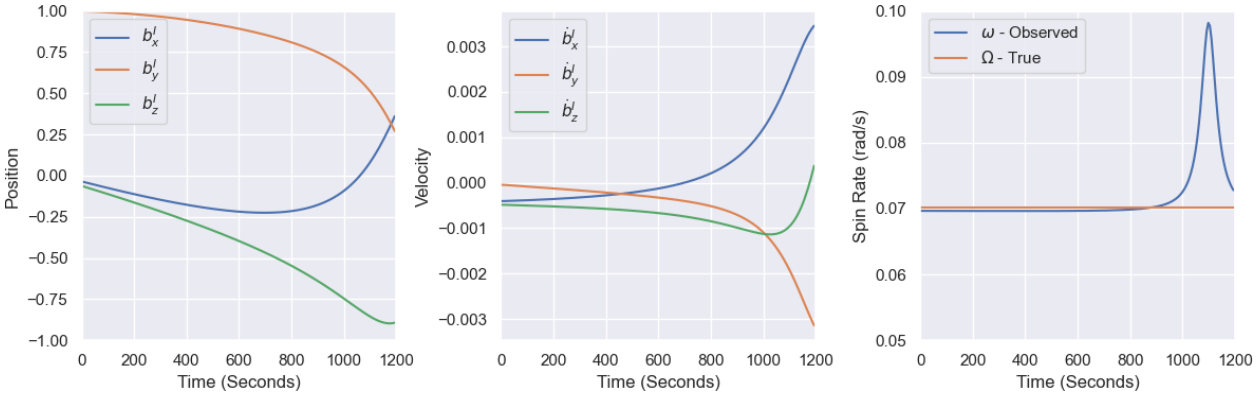
2.3 Adapting the Cost Function for Numerical Stability

Plugging the exact form of the PAB velocity from Equation 12 into the goodness-of-fit indicator of Equation 6 results in a highly non-linear function. The non-linearity of the goodnes-of-fit function has been documented for the approximated form of the PAB velocity equation by Zhao et al [5] in their study on spin-state estimation of rocket debris. As an example of this non-linearity, consider a scenario of a Medium Earth Orbit (MEO) satellite passing overhead, resulting in PAB position ($\mathbf{b}^I$) and velocity ($\dot{\mathbf{b}}^I$) vectors that are shown in Figure 2 (a). Assuming true spin-axis Euler angles of ($\theta \approx 26.3^\circ$, $\phi \approx 1.2^\circ$) and a sidereal (i.e. "true") spin rate of $\Omega = 0.07$ radians/second, Equation 5 results in an observed spin rate over time shown by ω (blue line) in the righthand plot of Figure 2 (a).

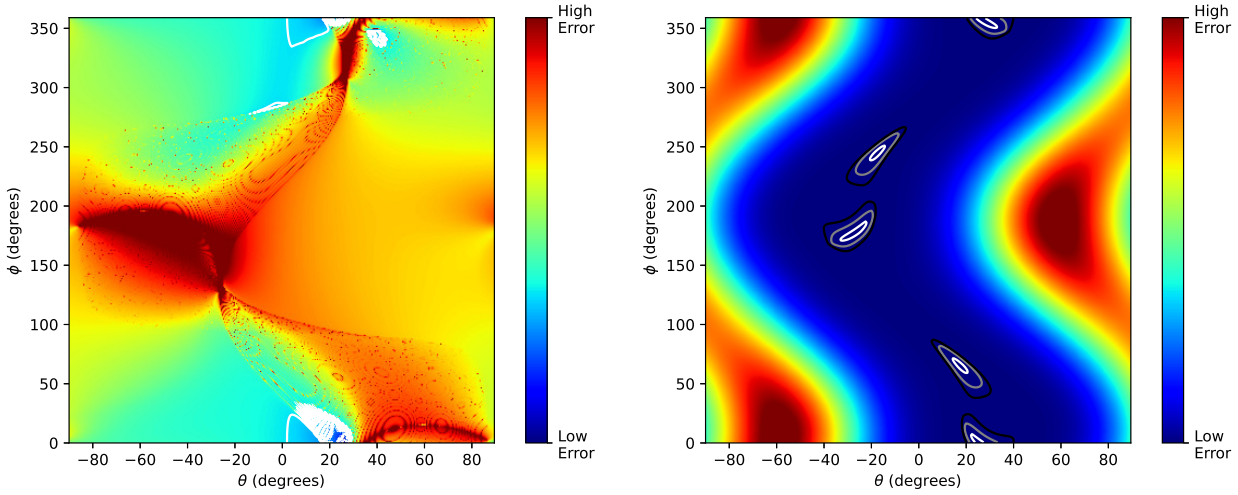
If the true value of Ω is known, plotting the cost function of Equation 6 for the data from Figure 2 (a) across the full span of (θ, ϕ) values at a resolution of 0.5° results in the loss field shown in Figure 2 (b). Examining the loss grid, it can be observed that there are scattered minima that are separated by contours of high loss errors. In particular, we have outlined the regions corresponding to the 1^{st} percentile of loss error using white contour lines. These regions of local minima are dispersed along a wide valley adjacent to the true spin-axis Euler angles, and are separated by striations of local maxima.

In reality, the exact value of Ω is not known *a priori*, leading to the necessity to perform an optimization over the full 3-dimensional parameter space of (Ω, θ, and ϕ). This can be represented by the "loss-cube" of Equation 6 that is shown on the left side of Figure 3. The z-dimension of this cube represents a grid that is centered on the true value

Fig. 2: An example simulated dataset for a MEO satellite passing over Atlanta (a), and associated Loss grids shown for the true value of Ω over the potential range of spin-axis Euler angles for the original Loss function of Equation 6 (b) and the proposed Loss function of Equation 23 (c). Note that in Figures (b) and (c), the contours of the 1^{st}, 2^{nd}, and 3^{rd} loss percentiles are shown as white, gray, and black lines, respectively.



(a)



(b) (c)

of $\Omega = 0.07$ radians/second and extends ± 0.025 radians/second. While the global minima of the cube (denoted by blue regions) occurs along the plane of the true Ω value, it is clear that there are many local minima where a gradient descent routine could become trapped due to an incorrect initialization of the Ω value.

Performing gradient descent on the cost function of Equation 6 is very unstable in practice due to this non-linearity. Our investigations revealed that the primary reason for this instability is the tendency for the denominator of the phase angle bisector velocity of Equation 12 to approach zero. An additional problem that will become clear in Section 2.4 is that there is an ambiguity in the sign associated with the PAB azimuthal velocity, such that only it's magnitude can be observed in the sidereal spin rate time series (i.e. Figure 1 (b)). To ameliorate these two problems, we algebraically re-arrange the presented equations of Equation 5 and Equations 13-17 to obtain a new cost function that only considers the magnitude of the PAB azimuthal velocity and arranges the denominator as a multiplicative term. For an individual timestep, this new cost function metric is obtained via the following steps:

$$\omega - \Omega + \frac{\partial \Psi(t, \theta, \phi)}{\partial t} = 0 \tag{18}$$

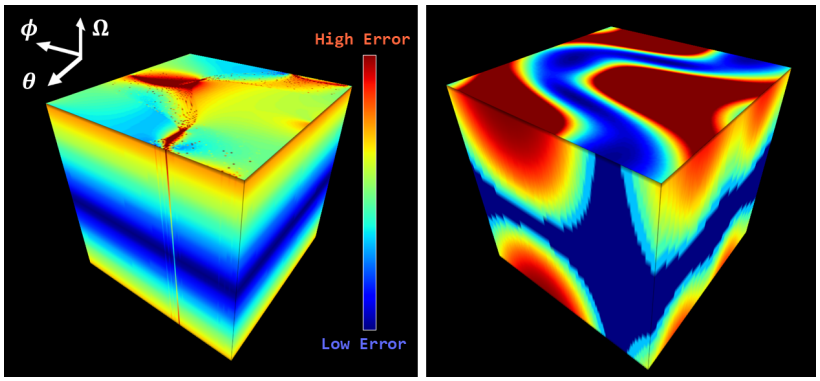


Fig. 3: Comparison of the loss cubes for the case of the loss function of Equation 6 (left) and proposed loss function of Equation 23 (right) over all three spin parameters that we sought to retrieve.

$$\omega - \Omega - \left(\frac{1}{ab}\right)\left(c\dot{\mathbf{b}}_x^I + d\dot{\mathbf{b}}_y^I + e\dot{\mathbf{b}}_z^I\right) = 0 \tag{19}$$

$$(ab)(\omega - \Omega) - \left(c\dot{\mathbf{b}}_x^I + d\dot{\mathbf{b}}_y^I + e\dot{\mathbf{b}}_z^I\right) = 0 \tag{20}$$

$$(ab)^2(\omega - \Omega)^2 = \left(c\dot{\mathbf{b}}_x^I + d\dot{\mathbf{b}}_y^I + e\dot{\mathbf{b}}_z^I\right)^2 \tag{21}$$

$$(ab)^2(\omega - \Omega)^2 - \left(c\dot{\mathbf{b}}_x^I + d\dot{\mathbf{b}}_y^I + e\dot{\mathbf{b}}_z^I\right)^2 = 0 \tag{22}$$

The equality presented in Equation 22 can then be utilized in a Mean-Squared Error (MSE) goodness-of-fit indicator that is based on the known geometrical constraints of the spin parameter estimation problem. For a series of N synodic measurements ω_i', the physics-loss function χ_{phy} is given by:

$$\chi_{phy}{}^2(\Omega,\theta,\phi) = \left(\frac{1}{N}\right)\sum_i^N\left((a_i b_i)^2(\omega_i - \Omega)^2 - \left(c_i\dot{\mathbf{b}}_{x,i}^I + d_i\dot{\mathbf{b}}_{y,i}^I + e_i\dot{\mathbf{b}}_{z,i}^I\right)^2\right)^2, \tag{23}$$

where the subscripts i denote the components that are dependent on time.

The loss plots of $\chi_{phy}{}^2$ for the data that was presented in Figures 2 (a) are shown in Figure 2 (c) and 3 (b). As can be observed, the transitions from regions of high to low error are much smoother for Equation 23 than for Equation 6. This makes the loss function more amenable to numerical derivative calculations that are deployed in both standard gradient descent approaches and backpropagation algorithms. However, there still persists the problem of spin-axis ambiguity of the solution even in this newly adapted cost function that must be addressed.

2.4 Spin-Axis Ambiguity of Cost Functions

A known limitation of the goodness-of-fit indicator in Equation 6 is the spin-axis ambiguity that can occur along the Euler angle (θ, ϕ) grid, even when the true sidereal spin rate (Ω) is known [5]. For an illustration of this problem, consider the case of a cylindrical piece of debris that has it's spin-axis oriented along either the top or bottom end of the cylinder as shown in Figure 1 (b).

Assume that right-hand conventions hold for the two potential spin-axis orientations in Figure 1 (b). Under this assumption the angular rotations due to the tumbling proceed in a clockwise manner about each of the spin-axes despite the absolute magnitude of the sidereal spin rates (i.e. $|\Omega|$) being the same. Under these conditions, the PAB velocities resulting from the two candidate spin-axes obey the relationship $(\partial\Psi_1/\partial t) = -(\partial\Psi_2/\partial t)$. Furthermore, we note that the magnitude of the PAB velocity on the perceived spin rate must be equal such that $|(\partial\Psi_1/\partial t)| = |(\partial\Psi_2/\partial t)| = |(\partial\Psi/\partial t)|$ [5]. By examination of Equation 5, we obtain the following equation to relate the perceived rotation rates of these two mirrored spin-axis candidates:

$$\Omega_2 = \Omega_1 - 2\frac{\partial\Psi}{\partial t} \tag{24}$$

Plugging these two terms into Equation 6 reveals that the two goodness-of-fit indicators provide the same value:

$$\sum_{i}^{N}\left(\omega_i-\Omega_2-\frac{\partial\Psi}{\partial t}\right)^2=\sum_{i}^{N}\left(\omega_i-\left(\Omega_1-2\frac{\partial\Psi}{\partial t}\right)-\frac{\partial\Psi}{\partial t}\right)^2=\sum_{i}^{N}\left(\omega_i-\Omega_1+\frac{\partial\Psi}{\partial t}\right)^2 \tag{25}$$

From this result, we can see that two possible spin orientations are produced due to geometrical symmetry, and the figure of the goodness-of-fit indicator should be symmetrical about these two potential candidates. This can be seen in our example scenario by the symmetry in Figures 2 (b) and (c). Recall that the true spin-axis value for this scenario occurs at a value of ($\theta \approx 26.3^\circ$, $\phi \approx 1.2^\circ$). The diagrams reveal that a second local minima occurs around the symmetric spin-axis region of ($\theta \approx -26.3^\circ$, $\phi \approx 181.2^\circ$). The angular resolution of the grid in Figure 2 (b) obscures this feature, but it can be clearly seen in 2 (c) due to the improved smoothness of our proposed loss function.

The spin-axis ambiguity problem presents a major challenge for utilizing standard gradient descent methods to retrieve spin-axis parameters due to the potential to drift in between valid solutions. Consequently, researchers have suggested that this spin-axis ambiguity can only be reduced by obtaining multiple observations in which the observer-target-sun geometry is significantly different [5, 13]. A major contribution of this paper is the presentation of preprocessing methods and sequential machine learning architectures which exhibit performance that significantly improves upon gradient-descent methods for spin parameter retrieval in the face of this spin-axis ambiguity.

3. PHYSICS-GUIDED MACHINE LEARNING

3.1 "Black Box" Machine Learning

The standard approach for training a machine learning model is to minimize the empirical loss of its model predictions ($\tilde{\mathbf{Y}}$) relative to the known truth values ($\mathbf{Y}$) across all j samples in the training dataset ($j \in [1, N_s]$). In the case of this paper, the target vector to retrieve is the 3-dimensional spin-state parameter vector ($\mathbf{Y} = [\theta, \phi, \Omega]$) generated using the simulation procedure outlined in Section 4.1. Because we are performing retrieval of continuous parameters, we seek to minimize the Mean-Squared Error (MSE) in between the target values and the predicted values by the ML architecture:

$$\mathscr{L}_{MSE}\left(\mathbf{Y}_j;\tilde{\mathbf{Y}}_j\right)=\left(\frac{1}{3}\right)\sum_{k=1}^{3}\left(\Delta(\mathrm{Y}_{j,k},\tilde{\mathrm{Y}}_{j,k})\right)^2\,, \tag{26}$$

where Δ is a difference operator that will be described in the next sub-section, and $[\mathbf{X}_j, \mathbf{Y}_j]_{j=1}^{N_s}$ denotes the set of N_s total time-series instances. Note that in this paper $\mathbf{X}_j = \left[\overrightarrow{\omega_j}, \mathbf{B}^I{}_j, \dot{\mathbf{B}}^I_j\right]$, which denotes that the time series of (1) the observed synodic spin rate, (2) the PAB position, and (3) the PAB velocity are ingested as input by the model to learn spin parameter retrieval. To clarify, the upper-case letter here signifies that this is a time-series of 3-dimensional PAB position/velocity vectors, and that $\mathbf{X}_j$ is therefore a $N \times 7$ matrix where N is the number of timesteps.

If Equation 26 is utilized as the loss function for training an ML model, then the model would operate as a physics-agnostic "black box" while learning to perform spin state retrieval. This is because the gradients of the MSE with respect to ML model parameters would be used to update the ML model parameters at each epoch [25]. These gradients are computed using automatic differentiation procedures for standard ML packages, and therefore would not leverage any known physical constraints. For this paper in particular, tensorflow was utilized which leverages automatic differentiation during the backpropagation stage of training [26].

While "black box" approaches have proven effective for solving a number of light curve inversion problems [15,16,18], there are many limitations of these approaches. One such issue is that "black-box" ML models require lots of data in order to train effectively. For many SDA problems of interest, it can be prohibitively difficult to collect and label thousands-to-millions of truth instances. Another issue is that a "black box" ML model may learn a solution that is suitable to the supplied training data (i.e. "in-distribution" data), but that does generalize well outside of said data's bounds (i.e. "out-of-distribution" data). Such issues can not only produce physically nonsensical results, but also lead to a model that produces excessive false alarms or true negatives in operational contexts. To fix these issues, we tested if injecting physically known loss functions to the training procedure could improve model stability.

3.2 Incorporation of Physics-Based Constraints on "Black-Box" Models

A major goal of this paper is to discover if using scientific knowledge in the form of a physics-based loss functions can aid in training generalizable ML models. Previous research in other domains has shown that physics-based loss functions can lead to more accurate and physically-valid predictions [21, 22, 27]. Given the wide body of theory that exists for many astronomy problems, leveraging physics-based cost functions is a natural extension for the SDA field.

Physics-based loss functions can take the form of either algebraic manipulations of physical variables or partial differentials of governing equations [21]. This paper utilizes the former as the guiding physics-based loss function. In particular, we utilize the adapted goodness-of-fit indicator from Equation 23 as our physics-based loss function. This physics-based loss function takes the form of:

$$\mathscr{L}_{PHY}\left(\mathbf{X}_j, \tilde{\mathbf{Y}}_j\right) = {\chi_{phy}}^2\left(\mathbf{X}_j, \tilde{\mathbf{Y}}_j\right) \ , \tag{27}$$

where we have explicitly written here that the physics-based loss function is being computed using the time series of observed synodic spin rates ($\overrightarrow{\omega}$), the known PAB position time series ($\mathbf{B}^I$), the known PAB velocity time series ($\dot{\mathbf{B}}^I$), and the predicted spin parameter values ($\tilde{\mathbf{Y}}$).

On a intuitive level, incorporating this loss function tells the model that it must account for the known physical relationship between the observed physical information ($\mathbf{X}$) and the predicted spin parameters ($\tilde{\mathbf{Y}}$). Our paper's hypothesis is that this will prevent the model from generating predictions that fit well to our training data, but that do not generalize well outside of the training dataset bounds.

On a systems level, incorporating this loss function in the ML automatic differentiation procedure ensures that the model parameters are updated not only with respect to the gradient of the MSE equation, but also with respect to an equation that is rooted in the physical nature of the problem [21]. The complete learning objective that is used in this study incorporates both loss functions and takes on the form of:

$$\mathscr{L} = \mathscr{L}_{MSE}\left(\mathbf{Y}_j; \tilde{\mathbf{Y}}_j\right) + \lambda\, \mathscr{L}_{PHY}\left(\mathbf{X}_j, \tilde{\mathbf{Y}}_j\right) \ , \tag{28}$$

where λ is a hyper-parameter that controls the relative importance of incorporating known physical information compared to the "black box" empirical loss function. Therefore, the loss function Equation 28 allows for trained models to capture the strengths of "black-box" ML approaches, while being regularized by known astronomy theory.

3.2.1 Difference Operators for Mean Squared Error Calculations

We return to the difference operator (Δ) that was utilized in the MSE formula of Equation 26. The calculation of this metric is unique to the three different spin-parameter components and requires some discussion. As was discussed in Section 2.4, a major challenge for the retrieval of the Euler angle components of the spin-state estimation problem is the ambiguity of two mirrored spin-axis solutions. An additional problem that has not yet been discussed in this paper, but that has been extensively covered in [19] is the difficulty that the periodic Euler angle representation presents for ML problems.

In order to address the spin-axis ambiguity effect on our ML loss functions, we considered both candidate spin-axis solutions in our evaluation metric calculation. Recall that we only simulated positive θ values in the range of $[0, \pi/2]$ radians. Therefore, in the event that the predicted θ value by an ML model under test was negative, we kept the mirrored (i.e. positive θ) form of the predicted spin-axis if it had a smaller total angular distance relative to the true spin-axis (i.e. known positive θ value).

A second problem that must be addressed is the distance function that is used in azimuthal angle distance calculations. Per the procedure outlined in [19], in order to handle the periodic nature of the azimuth Euler component, we first perform a modulo operation on the retrieved azimuth (ϕ) by a value of 2π to account for the potential of predicting an unwrapped phase angle. Then, when evaluating the proximity of a predicted ($\tilde{\phi}$) to a true (ϕ) azimuth value, we utilized a distance function [19, 28]:

$$\Delta_\phi(\tilde{\phi}, \phi) = \min\{|\tilde{\phi} - \phi|, 2\pi - |\tilde{\phi} - \phi|\} \tag{29}$$

This periodicity-accounting distance metric is leveraged in the MSE function of Equation 26 for *only* the azimuth component. A standard distance metric ($\Delta_\theta = \tilde{\theta} - \theta$) is utilized for the θ component because the predicted and true

Table 1: The TLE data that was utilized in this study to produce datasets of observed synodic spin rates. These parameters generated the training data that was used to train the Multi-Layer Perceptron and Sequential Encoder machine learning models and are therefore called the "in-distribution" orbital parameters.

Name	ID	i (degrees)	Ω (degrees)	e (degrees)	ω (degrees)	Mean Anomaly (degrees)	Revolutions/Day
Ariane 40	21610	98.3778	295.5686	0.0004192	62.4766	297.6848	14.4109
Lageos 1	8820	109.8524	58.4654	0.45106	242.6425	273.6296	6.3866
NavStar-77	43873	55.0987	147.6125	0.19776	191.557	12.004	2.005
Molniya 1-52	13012	61.7398	270.5101	0.7240671	255.5462	20.8309	2.004

values are both positive based on the spin-axis ambiguity resolution method presented above. A standard difference metric ($\Delta_\Omega = \tilde{\Omega} - \Omega$) is also utilized for the the sidereal spin rate component because we only care about the difference in magnitude between predicted and true velocities.

4. METHODS

4.1 Synthetic Dataset for Spin Parameter Estimation

In order to generate synthetic datasets for training and evaluation, we adapted simulation software that was written for our previous studies on satellite attitude estimation from light curves [14]. This software relies on using TLE data on a selection of LEO and MEO satellites that are visible from the southeastern United States.

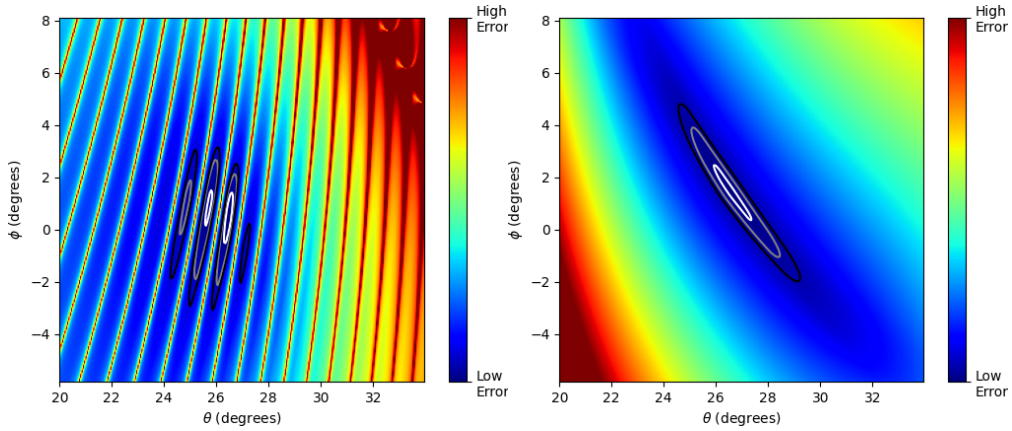
The four main satellite TLEs that were utilized in this study are shown in Table 1. We simulated the PAB vectors that would be observed from these satellite TLEs over a 4 year period from mid-town Atlanta, Georgia. These PAB vectors were sampled at 6 second intervals over a randomly initialized period of 1200 seconds for which the satellite was overhead. In short, each simulated synodic time series vector ($\vec{\omega}$) was of length 200 timesteps.

For each simulated PAB vector, we randomly selected values for the desired spin-axis parameters (Ω, θ, ϕ). The values for each simulated $\vec{\omega}$ vector were drawn from a uniform distribution over the ranges: $\Omega \in [1,5]$ (degrees/second), $\Phi \in [0, 2\pi)$ radians, and $\theta \in [0, \pi/2]$ radians. Note that we only drew the values of θ from the range of $[0, \pi/2]$ because we considered both ambiguous solutions for the Euler angle retrieval procedure to be valid in our study, as was just discussed for Equation 29. In other words, our distance metric calculations nullified the need to simulate negative θ datasets. The randomly drawn values of (Ω, θ, ϕ) and the accompanying PAB vector were then plugged into Equation 5 to produce the desired $\vec{\omega}$ vector as a function of time.

An additional "peak-finding" post-processing script was written after an investigation into the observability of spin-axis Euler angles from the simulated synodic time series. For many simulated instances, the synodic spin rate was fairly constant due to the PAB vector having relatively little motion over the course of the simulation. This can be observed in the first 800 seconds of the synodic time series in Figure 2 (a) where the synodic time series (ω, blue line) is not only constant, but it is relatively close to the of the true sidereal spin rate (Ω, orange line). From this time series, there would be almost no observability of the Euler angle components.

Only where there is a substantial change in the synodic time series in the form of a "peak" at 1100 seconds can we potentially discern the values of (θ, ϕ). Therefore, we ran a peak-finding algorithm on all our $\vec{\omega}$ vectors and only kept time series in which there was a peak with a prominence of at least 0.1 radians/second. We consider this a stopgap measure on this problem. The observability of the spin-axis Euler angles from an synodic time series is an open problem that we will look into in a future study.

This simulation and post-processing procedure yielded approximately 40,000 time series with accompanying truth data that were saved as XML files. The data was split into a 2/1 training-to-validation breakdown in order to train and evaluate the performance of each considered framework for spin-axis retrieval. This dataset was the "in-distribution" dataset because the training and validation data both came from the same set of TLE parameters. The "in-distribution" results are shown on the approximately 13,333 validation dataset samples in Section 5.1. As we will discuss in Section 5.2, we also generated an "out-of-distribution" dataset that came from a separate set of TLE parameters that was unique relative to the parameters shown in Table 1. This "out-of-distribution" dataset was meant to test the generalization of the physics-guided models.



(a) (b)

Fig. 4: Fine resolution loss grids shown for the true value of Ω over the potential range of spin-axis Euler angles for the original loss function of Equation 6 (a) and the proposed loss function of Equation 23 (b). Note that in sub-figures (a) and (b), the contours of the 1^{st}, 2^{nd}, and 3^{rd} loss percentiles are shown as white, gray, and black lines, respectively. Note that the true spin-axis Euler angle occurs at ($\theta \approx 26.3^{\circ}$, $\phi \approx 1.2^{\circ}$).

4.2 Architectures for Retrieving Spin State Parameters

In this section, we describe the architectures that were tested for spin parameter retrieval performance. The algorithms spanned a range of machine learning and standard regression techniques. We evaluate the effect of the physics loss function on each proposed model in the Results Section.

4.2.1 Regression Via Gradient Descent

Previous efforts at retrieving spin state parameters from the goodness-of-fit indicator in Equation 6 operate by first performing a grid search at a low angular resolution over the full (θ, ϕ) domain and then deploying a numerical minimization procedure to home-in on the best-fit Euler angles [5, 13]. After the Euler angles are solved for, the true spin rate can then be fine-tuned to provide the final spin parameter that is required [13].

A downside of this mode of operation is that the highly non-linear nature of the original goodness-of-fit indicator (Equation 6) does not guarantee a smooth descent towards the global minima even if the grid search places the initial spin-axis estimate of the gradient descent method adjacent to the global minima. For example, if we zoom in on the loss grids that were shown for the example in Figure 2 at a resolution of 0.05°, we obtain the loss grids of Figure 4. The original loss grid (Figure 4 (a)) has striations of high error that could potentially cause a gradient descent routine to drift away from the true spin-axis of ($\theta \approx 26.3^{\circ}$, $\phi \approx 1.2^{\circ}$). On the other hand, our proposed goodness-of-fit indicator (Equation 23) has a smooth descent towards the true spin-axis, as can be observed in Figure 4 (b). This result highlights that our proposed loss function not only has advantages for physics-guided machine learning, but also for traditional gradient descent approaches.

In order to compare our sequential machine learning models with blind gradient descent routines for spin parameter retrieval, we performed regression on the parameter space (θ, ϕ, Ω) by attempting to minimize either the original goodness-of-fit indicator of Equation 6 or the proposed goodness-of-fit indicator of Equation 23. In other words, we did not employ the joint cost function of Equation 28 as a function of varying λ values.

The numerical optimization was written in tensorflow [26]. We performed gradient descent on a sample-by-sample basis for each of the N_s samples using the Adam optimizer [29]. The descent ran for 35 epochs, with early stopping occurring if the goodness-of-fit indicator under test changed by less than 0.1% between epochs. Furthermore, the initial values of (θ, ϕ, Ω) were randomly seeded 10 different times. The run with the lowest goodness-of-fit value across these randomly seeded runs was kept as the best solution. This random seeding allowed for the potential to overcome a poor initialization of the search space. We acknowledge that this blind approach does not perfectly mimic the proposed grid-search then fine-tune approach that has been outlined in previous literature [5, 13]. However, we

believe that this approach provides a suitable method for assessing the effects of a poor initialization of the search space of the highly non-linear functions of Equations 6 and 23.

4.2.2 Multi-Layer Perceptron (MLP) Model

A Multi-Layer Perceptron (MLP) constitutes the most traditional architecture for deep learning. This form of architecture is also commonly referred to as a Fully-Connected (FC) network because each neuron in layer l is connected to every neuron in layer $(l-1)$ for the $l \in [1, L]$ layers of the network [30]. This model was chosen as a baseline to exhibit the performance of non-sequential architectures in spin parameter retrieval because it is a commonly chosen deep learning method for time series classification [31, 32] and time series forecasting [33]. The basics of MLPs are outlined widely in many publications, with [30] serving as an excellent overview of the model. For that reason, it will only briefly be discussed here.

In its most basic form, the connections between neurons at connected layers of a MLP are modeled by learned weights and biases. In other words, the response output of the k^{th} neuron in the l^{th} layer for an input time series $\mathbf{Z}$ can be represented by the following equation:

$$R_{l_k} = f_{A,k}\left(\mathbf{W}_{l_k}\mathbf{Z} + C_k\right), \tag{30}$$

where $\mathbf{W}_{l_k}$ are the learned weights for the neuron via backpropagation [25], C_k is the bias weight of the neuron, and $f_{A,k}$ is the chosen non-linear activation function [30]. Note that while Equation 30 is shown for a single neuron in the l^{th} layer for simplicity, there are nearly always multiple neurons in any single layer that are selected as hyper-parameters.

Equation 30 shows both the strengths and limitations of the MLP framework for time-series classification. The strength of the MLP is that it can learn highly non-linear decision boundaries by the use of an appropriate non-linear activation function and learned weights. The limitation is that MLP neurons do not exhibit any temporal variance because each time stamp has its own weight matrix ($\mathbf{W}_{l_k}$). The temporal dependence across observed points is therefore lost due to timesteps being treated independently from each other [30].

The MLP model that was utilized in this paper had $L = 4$ total layers with 32 neurons per layer. Each FC hidden layer employed a Rectified Linear Unit (ReLU) activation function [34]. Dropout layers with a probability of 25% were utilized for each of the hidden layers that preceeded the final FC layer. This final FC layer had a linear activation function because regression was being performed on the MSE cost function. The Adam optimizer was leveraged in order minimize the error function over the training dataset [29]. Early stopping with a patience level of 45 epochs was utilized in order to ensure that overfitting of the models did not occur.

Because the focus of this paper was on examining the benefits of physics loss functions and sequential models for spin parameter retrieval, the MLP model in this paper was only hypertuned over the learning rate of the Adam optimizer. The optimal learning rates for the various λ values were found by performing model training of the MLP model over the range of [1e-3, 1e-6] using a search grid with multiples of 5. The models were trained and evaluated for λ values of: $[0, 10, 50, 100, 500, 1000]$. This provided the ability to evaluate the change in model performance from the case of no physical insight being used, to maximal physical insight being used.

4.2.3 Sequential Encoder Model

A key limitation of Convolutional Neural Networks and MLP models is that they are temporally invariant. In other words, the relationship between a previous and current timestep is never factored into a prediction. Research into astronomy and SDA problems has revealed that this can be a major hindrance to performing time series classification on light curves [35, 36]. Because of these known issues, we propose an adapted form of our previously published squential encoder architecture for the spin parameter retrieval task [36].

The proposed architecture employs Recurrent Neural Networks (RNNs) rather than convolutional filters for feature extraction. RNNs are advantageous for handling time sequences because they are capable of treating input time-series in a sequential manner via the concept of a memory [37]. The RNN memory cell allows an individual unit to extract features from an input time series in a time-variant manner by combining both previous patterns and current input information [38]. In this paper, the Long Short Term Memory (LSTM) RNN archetype is employed in the classifier model in order to retrieve spin parameters [39]. The LSTM is covered exhaustively in countless publications and will

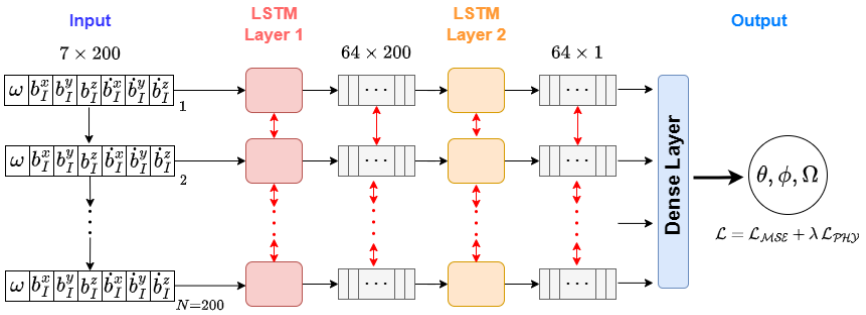


Fig. 5: A diagram of the sequential model used for spin parameter retrieval. The model accepts observed spin rate, Phase Angle Bisector (PAB) position, and PAB velocity as inputs across N timesteps. The time series is processed by hidden bi-directional LSTM layers with ReLU activation functions to produce a sequence of extracted features. A fully-connected layer with linear activation accepts the output of the final hidden LSTM layer and predicts spin parameters, $\tilde{\mathbf{Y}} = (\theta, \phi, \Omega)$.

therefore only be briefly mentioned here. Interested readers, in particular, can refer to publications by [35] and [38] for an overview of the LSTM algorithm.

The LSTM is an advancement over standard RNNs in that it utilizes "gates" that can both propagate and/or forget temporal trends [38]. The LSTM propagates trends via the concept of a memory cell that is passed across the time dimension. The LSTM controls the memory state using three different gate mechanisms: (1) an input gate that determines if input information at the current time is relevant to memorized temporal trends, (2) a forget gate that decides if the previous memory state is relevant to new trend information, and (3) an output gate that computes the next memory state based on consideration of previous output information and current memory state [36]. The combination of such gates with the memory cell enable an LSTM to extract features from a time series in a sequential and time-variant manner [35].

The LSTM is used as the basis of a sequential encoder model that was originally proposed by Naul et al for light curve analysis of stars [35], and recently adapted for spin-stability assessment of satellite light curves [36]. The model architecture used in this paper is shown in Figure 5. As can be seen, we adopt only the encoder portion of the original autoncoder for this study's spin parameter retrieval task [36].

The model diagram begins from the left and reads to the right in Figure 5. At the input stage, the observed synodic spin rates (ω_i), PAB position ($\mathbf{b}^I{}_i$), and PAB velocity ($\dot{\mathbf{b}}^I$) at each of the $i \in [0, N = 200]$ timesteps are input to the architecture as a $7 \times N$ matrix. These are the known variables that drive the discrepancy between synodic and sidereal spin rate, so they must be input in order to allow the ML model to learn how to predict spin parameters from these physical drivers. The bi-directional LSTM units that make up the hidden layers ingest the input time series and perform feature extraction. These LSTM layers maintain hidden states that are capable of extracting information on a sequential basis, as indicated by the double-sided red arrows that share information across the time domain. The final hidden LSTM layer passes the extracted feature embedding to a FC layer that has a linear activation function for performing regression of the desired spin parameters (θ, ϕ, Ω).

The sequential model that was utilized in this paper had 2 hidden layers with 64 total RNN units each. Each hidden layer had a ReLU activation function and was followed by a dropout layer with a probability of 25%. For the sake of focus on the benefits of physics-guided loss functions, sequential model was only hypertuned over the learning rate of the Adam optimizer. The optimal learning rates for the various λ values were found by performing model training of the MLP model over the range of [1e-3, 1e-6] using a search grid with multiples of 5. The models were trained and evaluated for λ values of: $[0, 10, 50, 100, 500, 1000]$.

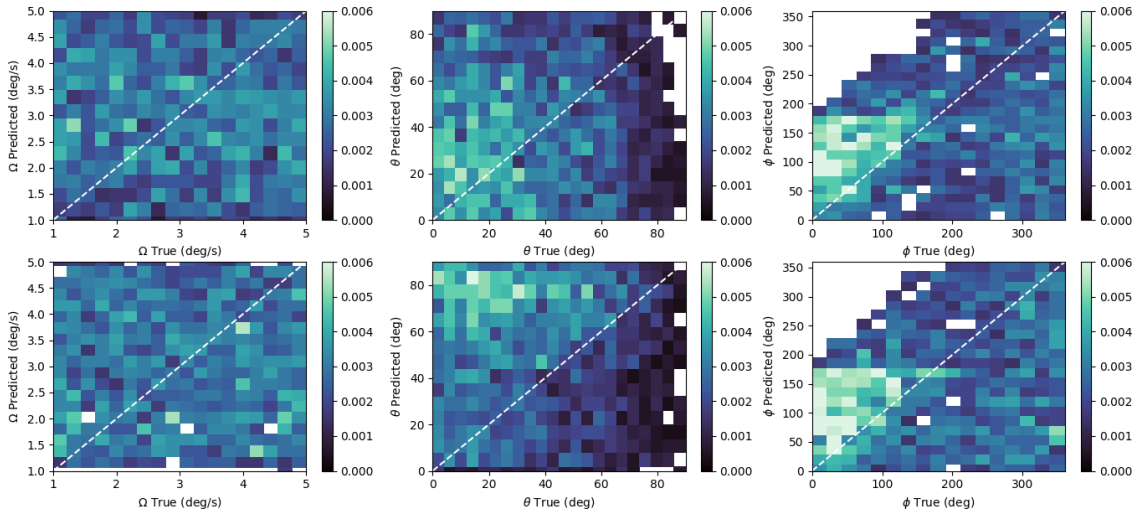


Fig. 6: Two-dimensional histograms of predicted vs. true values of Ω (left), θ (middle), and ϕ (right) when using a gradient descent approach with a the physics loss function (top row) and the original loss function (bottom row). This is shown for "in-distribution" validation data.

5. RESULTS

5.1 Retrieval of Spin State Parameters By Models on "In-Distribution" Validation Data

In this section we present the results of the models that were just presented on the validation dataset that was generated using the TLE data from Table 1. This validation data came from the same distribution as the training dataset. In other words, the model is trained on the same distribution of satellite orbital parameters on which it is evaluated. The goal of this section is to show the performance of the sequential encoder model relative to other spin parameter estimation techniques on the "in-distribution" dataset.

5.1.1 Regression Via Gradient Descent

The 2-dimensional histograms in Figure 6 show the results of the gradient-descent regression model from Section 4.2.3 to retrieve the three spin parameters. Recall that we performed ten different runs per dataset in which we randomly seeded the initial spin parameters and then performed gradient-descent across 35 epochs with early stopping. Therefore, there was no guarantee that any of the randomly seeded initial spin parameter values were adjacent to the global minimum solution. After all ten regressions had been performed, we kept the final spin parameters that had the lowest overall goodness-of-fit value across all of the runs. The resulting histograms for the proposed loss function (Equation 23) and the original loss function (Equation 6) are shown in the top and bottom rows, respectively.

From these results, we can observe that performing regression using both the proposed loss function and the original loss function are highly prone to getting trapped in local minima. The white dashed lines show the one-to-one regression line between the true and predicted parameter values. From these 2-dimensional histograms, there is no apparent higher density surrounding this desired region. Additionally, there is no apparent difference in gradient-descent model accuracy when using either the proposed or the original goodness-of-fit indicator as the loss function. This is to be expected given the highly non-linear nature of the 3-dimensional loss space that was illustrated in Figure 3.

The main takeaway from these histograms is that the highly non-linear natures of Equations 6 and 23 confounds the use of standard gradient descent approaches for retrieval of all three spin parameters. This is the reason that Hall et al originally suggested that there should be three steps to spin parameter retrieval: (1) retrieve the approximate sidereal spin rate via Fourier analysis, (2) perform a grid search over the potential (θ, ϕ) values, and finally (3) fine tune the grid search result via numerical optimization [13]. While Hall et al.'s approach would perform better than what is shown in Figure 6, it is very computationally expensive. It also does not handle issues mentioned such as the potential to get trapped in local minima (i.e Figure 4 (a)) or the spin-axis ambiguity problem.

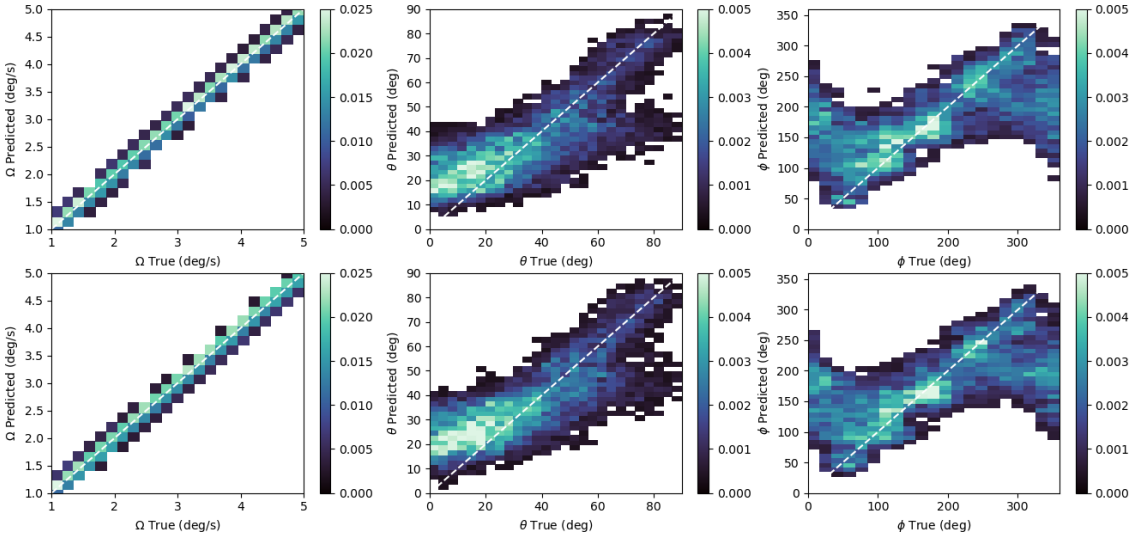


Fig. 7: Two-dimensional histograms of predicted vs. true values of Ω (left), θ (middle), and ϕ (right) when using a Multi-Layer Perceptron model with a physics loss value of $\lambda = 10$ (top row) and physics loss value of $\lambda = 0$ (bottom row). This is shown for "in-distribution" validation data.

We have already illustrated via Figures 2 and 4 that our proposed loss function is more favorable to gradient descent techniques via the reduction of loss function non-linearity. We will next show that our sequential machine learning model provides an advancement over gradient descent methods for predicting spin parameters.

5.1.2 Multi-Layer Perceptron (MLP) Model

We next evaluate the ability of the MLP model to retrieve the desired spin parameters. Recall that the MLP model is a temporally invariant architecture that is trained using 66% (≈26,666 time series) of the "in-distribution" data, which is classified as training data generated from the TLE data in Table 1. It is then evaluated on 33% of the data (≈13,333 time series) generated from the TLE data in Table 1, which is classified as validation data. The 2-dimensional histograms that show the ability to retrieve the desired spin parameters over this validation dataset are shown in Figure 7. The result from the best performing physics-guided training model with $\lambda = 10$ is shown in the top row. The result when using no physics-guided loss term (i.e. $\lambda = 0$) is shown in the bottom row.

An inference that can be immediately drawn from the left-hand plots in Figure 7 is that the MLP model performs much better than randomized gradient descent for retrieval of sidereal spin rate. We can see that when using either standard loss functions or physics-guided loss functions that the retrieved Ω distribution very closely hugs the one-to-one trend-line. Based on the fact that the Ω term is the only non-linear component of Equation 6, we can discern that the MLP model performs well when tasked with linear regression. The limitations of the MLP model, however, become apparent when examining how well it performs at retrieval of the Euler angle components. For the zenith parameter (middle histograms of Figure 7), there appear to be tendencies to under-predict θ values for the case of true $\theta > 40^\circ$, and to over-predict θ values for the case of true $\theta < 20^\circ$. For the azimuth parameter (right-hand histograms of Figure 7), there appears to be a sinusoidal distribution of prediction errors that is centered about the true $\phi = 180^\circ$ line.

In summary, for both Euler angle components there are non-linear distributions of errors in the predicted values relative to the true values. These error distributions are likely due to the highly non-linear nature of the PAB azimuthal velocity as a function of the Euler angle components. This is supported by the fact that the MLP model is capable of retrieving the sidereal spin rate upon which the synodic spin rate is linearly dependent, per Equation 5. An additional finding is that the retrieval accuracy of the MLP model is only marginally, if at all, boosted by the incorporation of a physics-based loss term when the training and validation data come from the same orbital parameter distribution. The distributions of errors in Figure 7 suggest that the physical constraints were not effective in aiding the training process for "in-distribution" validation data. As we will show in the next section, temporally variant architectures appear to be

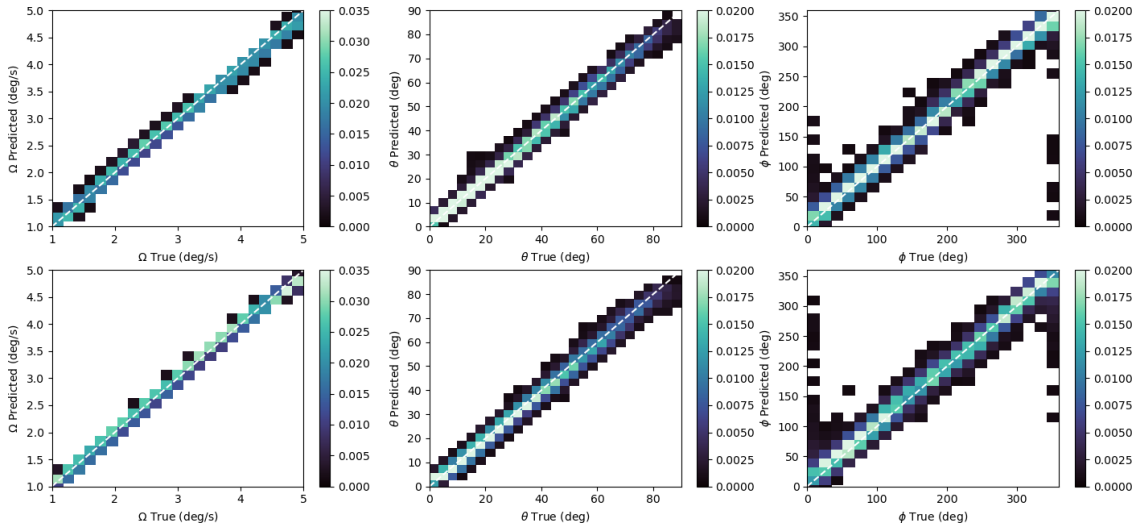


Fig. 8: Two-dimensional histograms of predicted vs. true values of Ω (left), θ (middle), and ϕ (right) when using a sequential model with a physics loss value of $\lambda = 500$ (top row) and physics loss value of $\lambda = 0$ (bottom row). This is shown for "in-distribution" validation data.

more capable of handling non-linear regression and to be aided by physics-based loss terms.

5.1.3 Sequential Encoder Model

The limitations of the randomized gradient descent approach and the MLP model suggest that a temporally variant machine learning model must be employed to handle the highly non-linear spin parameter retrieval task. For this reason, we derived the sequential encoder model that was presented in Figure 5.

The sequential encoder model was trained and then evaluated using the same training and validation datasets as the MLP model. The 2-dimensional histograms of predicted vs. true spin parameters of the validation dataset are shown in Figure 8. The result from the best performing physics-guided training model with $\lambda = 500$ is shown in the top row. The result when using no physics-guided loss term (i.e. $\lambda = 0$) is shown in the bottom row.

When qualitatively comparing the performance of the sequential encoder model to that of the other models in this study, it is apparent that the sequential encoder model performs best at the non-linear prediction of spin parameters. This suggests that the temporal relationship between neighboring points in the observed spin rate time series is important to account for when performing spin parameter retrieval. The sequential encoder model is able to retrieve both the Ω term upon which the PAB azimuth velocity is linearly dependent, as well as the Euler angles upon which the PAB azimuth velocity is non-linearly dependent. In particular, the sequential model does best at retrieving the zenith (θ) term, but appears to struggle when retrieving the azimuth (ϕ) term. This is shown on the right-hand side of Figure 8 by an increased spread in the distribution of prediction errors as the true ϕ value approaches 0° from the right and 360° from the left.

The reason for this inability to retrieve the true ϕ values becomes clear when the predictions are plotted in different bins of the true θ value. This breakdown is shown in a color-coded form in Figure 9. From this figure, it is immediately apparent that the largest errors occur when the true θ value approaches either 0° or 90°. Conceptually, this makes sense because when the true value of $\theta \approx 0^\circ$, the goodness-of-fit indicator equations are symmetric about the $\phi = 180^\circ$ line. This means that at least two different ϕ values map to the same loss value, explaining the poor retrieval of ϕ values for the $\theta \in [0^\circ, 15^\circ]$ bin of Figure 9. The same problem occurs if the true value of $\theta \approx 90^\circ$, which explains the poor retrieval of ϕ values for the $\theta \in [75^\circ, 90^\circ]$ bin of Figure 9. Mathematically, this can also be explained by considering that observability of ϕ in the PAB azimuth velocity equation is reduced when $\theta = 0^\circ$ or $\theta = 90^\circ$ in Equation 3.

An additional issue that is clear for Figure 9 is that there are certain bins not adjacent to $\theta = 0^\circ$ or $\theta = 90^\circ$ that produce "hockey-stick" distributions where the tail branches upward as ϕ approaches 0° from the right. This could

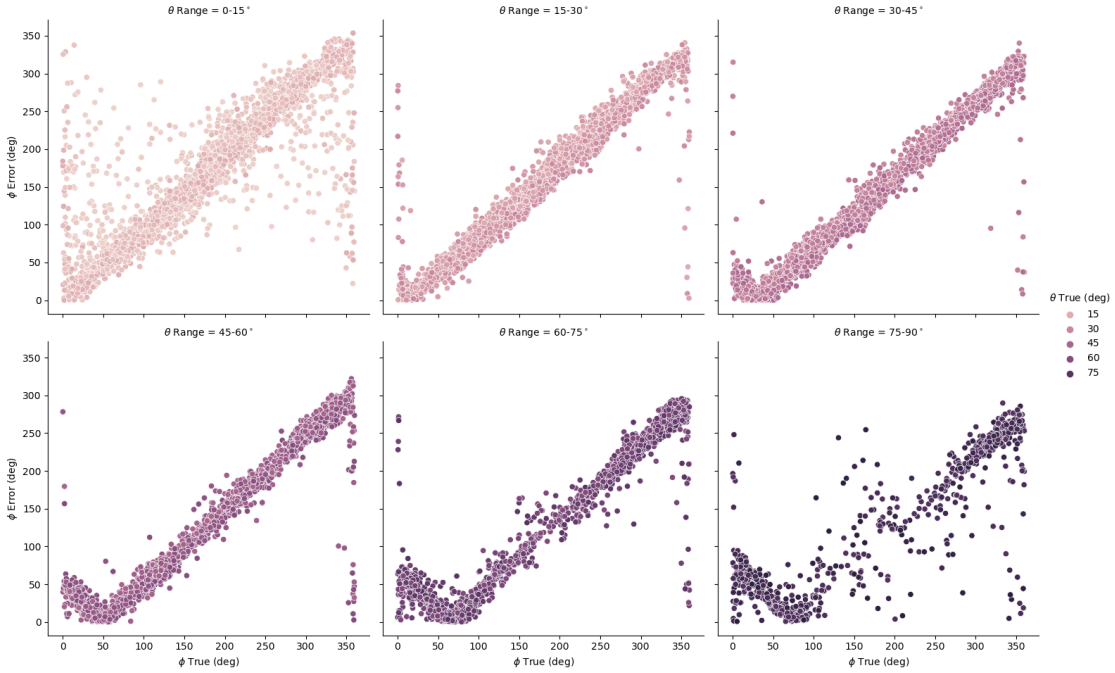


Fig. 9: Scatter plots showing distributions of the true (x-axis) and predicted (y-axis) values of ϕ by the physics-guided sequential encoder model. The individual plots and colors show the plots broken down by 15° increments of the true θ value. This is shown for "in-distribution" validation data. This is shown for "in-distribution" validation data.

be due fitting errors that are inherited by the model when attempting to retrieve the true ϕ value for spin axes with θ close to either 0° or 90°. For these reasons, a recommended improvement for future studies that we will look into is to apply a lower weight to the retrieval of true ϕ values when the θ value approaches either 0° or 90°.

An additional point to consider is whether the physics-guided loss term improves the sequential encoder model's spin retrieval performance on "in-distribution" validation data. When comparing the predictions of the physics-guided sequential encoder model those of the standard sequential encoder model, there are no apparent improvements in the ability to retrieve the true θ or Ω values. There does appear to be a marginally smaller spread in the distribution of ϕ retrieval errors.

5.1.4 Quantitative Retrieval Metrics

The results of the previous sub-section were presented as histograms to provide a qualitative description of each model's performance. The quantitative mean and standard deviation of prediction error for each model on the "in-distribution" validation data are shown in Table 2. The minimal mean error value is highlighted in bold text for the respective columns of θ, ϕ and Ω. From this table, it can be confirmed that the sequential encoder model is the best performing architecture among those considered in this study for the retrieval of the Euler spin-axis parameters. This suggests that the temporal dependence of neighboring synodic frequency observations is important to retrieve the non-linear Euler angle components. Surprisingly, the MLP model outperformed the sequential encoder model on the retrieval of the linear sidereal spin rate. This suggests that a combination of sequential and hierarchical models may be useful for optimal retrieval of all three spin parameters. This will be examined in a future study.

5.2 Retrieval of Spin State Parameters By Models on "Out-of-Distribution" Data

The previous discussion of Section 5.1 presented results for the "in-distribution" data. This term means that the model was trained on a dataset that was generated from the orbital parameters of Table 1, and then validated on a second dataset generated from these same orbital parameters. One of the hypotheses of this paper was that the use of a physics-guided loss function can lead to more generalizable ML models. In order to test this hypothesis, we generated a second "out-of-distribution" dataset from a set of orbital parameters that differed from those of the training data.

Table 2: The mean error and standard deviation of error for each of the considered models on "in-distribution" validation data that came from the same orbital parameters (see Table 1) as the training data.

Model / Approach	Loss Function	θ Mean (degrees)	θ Std. Dev. (degrees)	ϕ Mean (degrees)	ϕ Std. Dev. (degrees)	Ω Mean (degrees/second)	Ω Std. Dev. (degrees/second)
Sequential Encoder	Physics	**2.349**	3.083	**14.954**	32.888	0.025	0.103
Sequential Encoder	Normal	2.739	3.340	19.251	34.836	0.006	0.080
Multi-Layer Perceptron	Physics	12.089	9.855	64.802	58.384	**0.005**	0.115
Multi-Layer Perceptron	Normal	11.998	9.511	64.744	57.983	0.005	0.120
Gradient Descent	Physics	46.467	52.884	88.980	52.042	0.286	2.137
Gradient Descent	Normal	53.686	54.202	90.069	52.013	0.036	1.662

The "out-of-distribution" orbital parameters are shown in Table 3. For example, It can be seen that the inclination and eccentricity of these values substantially differ from the "in-distribution" values of Table 1. From these new orbital parameters, we generated approximately 5,000 additional time series and tested the trained sequential encoder model on this new dataset. Note that no retraining was performed on the sequential encoder model that would have prepared the model for time series generated using these differing orbital parameters.

The resulting 2-dimensional histogram for the sequential encoder model trained with an MSE loss function and a physics-guided loss function are shown in the bottom and top rows of Figure 10, respectively. Note that a physics hyper-parameter of $\lambda = 500$ was used for the physics-guided sequential encoder model, based on this model hyper-parameter showing the best performance on the "in-distribution" validation data.

While we acknowledge it is a limited test case, Figure 10 shows that using a physics-guided loss term leads to better generalizability of the sequential encoder model to unseen satellite instances. In particular, we note that the predictions of the physics-guided model much more closely track the one-to-one trendline for retrieval of sidereal spin rate (Ω) than those of the non-physics-guided (i.e. MSE loss trained) model. It can also be seen that the physics-guided model does a better job of retrieving the azimuth (ϕ) component of the spin-axis Euler angles based on the higher overall density surrounding the one-to-one trendline. Interestingly, both models struggle with the retrieval of the zenith (θ) component of the spin-axis Euler angles.

These results can be quantitatively validated by computing the mean and standard deviation of prediction errors for the three spin parameters. The results for the physics-guided and non-physics-guided sequential encoder models on the "out-of-distribution" dataset are shown in Table 4. From these results, it is clear that the physics-guided model has a lower overall mean error on the retrieval of all three spin parameters

A downside is that the standard deviations for retrieval of all three parameters are quite high relative to the "in-distribution" errors that were shown in Table 2. This suggests that the physics-guided loss function of Equation 27 does not guarantee models that will be fully generalizable to unseen satellite observations. Therefore, we conclude that a broad dataset as well as properly tuned loss functions must be leveraged in order to generate deployable models that will generalize well to unseen observations. Fortunately, a second advantage of our physics-guided loss function is that it has the potential to accelerate training on large datasets, as we will show in the next section.

5.3 Effect of Physics Loss on Training Speed and Stability

As was mentioned in the previous section, a key requirement of training a generalizable model for satellite spin parameter retrieval is the use of a large dataset. This will require large amounts of data with varying satellite orbital parameters, observation sites, and times of year. A downside of this is that a large dataset not only requires more training time, but can also be prone to unstable training across epochs. For these reasons, any method that can either stabilize the backpropagation or speed up training is desirable. Our results show that the incorporation of a physics-

Table 3: The TLE data that was utilized in this study to produce a second set of validation data that was not seen by the sequential encoder model during training. These parameters were not used to generate any training data, and are therefore called the "out-of-distribution" orbital parameters.

Name	ID	i (degrees)	Ω (degrees)	e (degrees)	ω (degrees)	Mean Anomaly (degrees)	Revolutions/Day
SL-16 R/B	22803	70.9913	12.8646	0.0019269	237.3761	122.5503	14.1688
H2A R/B	45166	97.0779	324.5919	0.0016035	151.1481	208.8517	15.3158

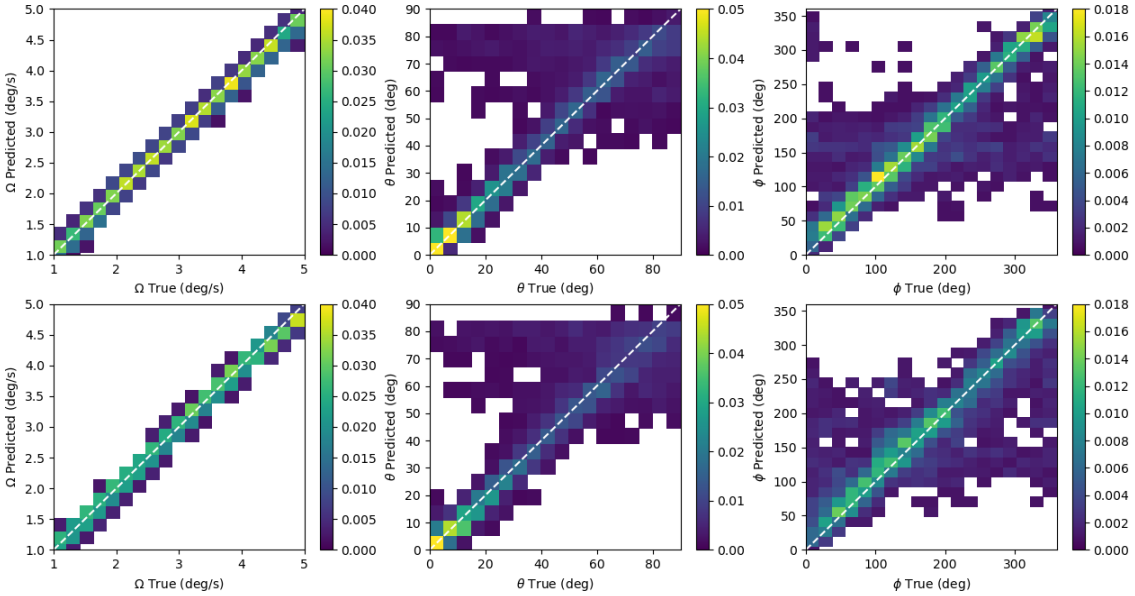


Fig. 10: Two-dimensional histograms of predicted vs. true values of Ω (left), θ (middle), and ϕ (right) when using a sequential model with a physics loss value of $\lambda = 500$ (top row) and physics loss value of $\lambda = 0$ (bottom row). This is shown for "out-of-distribution" validation data generated using orbital parameters from Table 3.

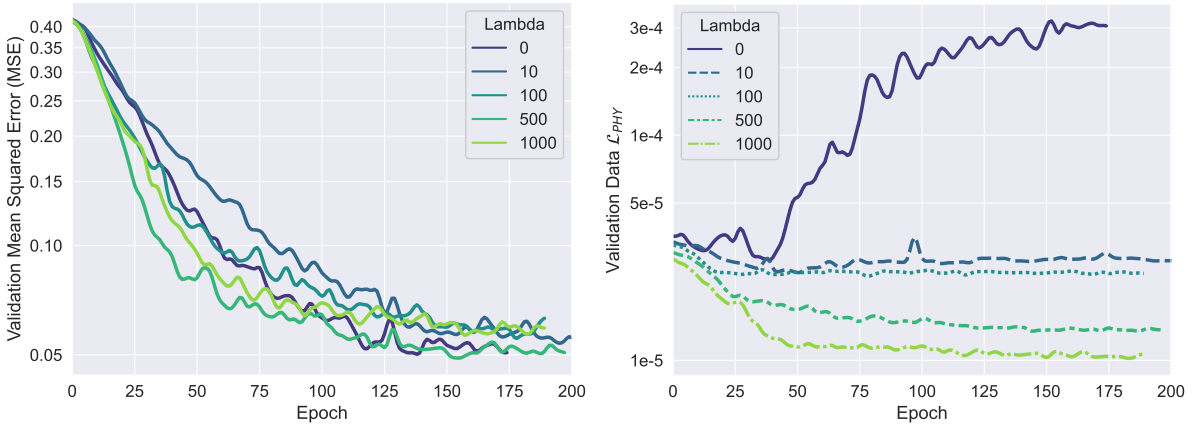
based loss function can potentially accomplish both of these goals.

The loss curves for the mean squared error (Equation 26) and physics-based loss function (Equation 27) are shown in Figures 11 (a) and (b), respectively. These curves are shown for the "in-distribution" validation dataset as a function of the physics hyperparameter λ for a constant learning rate of 9E-04 that was found to be the optimal rate via hyper-tuning. From the MSE curve in Figure 11 (a), it can be seen that large values of λ can speed up the training process even though the learning rate has not changed. This is particularly true in the early epochs for physics hyper-parameters of $\lambda \in [500, 1000]$. However, for the later epochs it can be seen that the use of an improperly tuned λ value can actually lead to a lower overall MSE in the retrieval of the spin parameters (θ, ϕ, Ω). It is possible that the physics-guided models are struggling primarily with the retrieval of the spin rate, as was shown by the lower overall mean error in Ω retrieval from Table 2. This is an issue that we will look into in our future work as we improve our physics-guided ML approaches.

While the final MSE values for physics-guided models are sometimes lower than those for non-physics-guided models, the value of $\mathscr{L}_{PHY}$ (Equation 27) is stabilized by increasing the magnitude of the λ hyper-parameter. This can be seen by the plots of Figure 11 (b) where increasing λ leads to a lower magnitude of $\mathscr{L}_{PHY}$. Interestingly, the use of a λ value equal to zero (i.e. a non-physics-guided model) results in a curve of $\mathscr{L}_{PHY}$ that diverges as the training proceeds. This suggests that the "black box" training of the sequential encoder model causes it to learn trends that are not consistent with the known physics-constraints of the spin parameter retrieval problem. This result potentially explains the reduced generalizability of the non-physics-guided model to "out-of-distribution" time series from Figure 10.

Table 4: The mean error and standard deviation of error for the sequential encoder model as a function of physics-guidance λ value. A λ value of zero means that no physics-guidance was employed in the loss function. Note that the results come from the "out-of-distribution" validation data that came from different orbital parameters (see Table 3) than the training data.

Model	**Lambda Value**	**θ Mean (degrees)**	**θ Std. Dev. (degrees)**	**ϕ Mean (degrees)**	**ϕ Std. Dev. (degrees)**	**Ω Mean (degrees/second)**	**Ω Std. Dev. (degrees/second)**
Sequential Encoder	500	**3.27**	14.52	**16.54**	58.24	**0.060**	0.097
Sequential Encoder	0	**4.23**	14.35	24.05	58.01	0.0635	0.112



(a) (b)

Fig. 11: The loss curves for the mean squared error (Equation 26) and physics-based loss function (Equation 27) are shown on the "in-distribution" validation dataset in (a) and (b), respectively. These curves are plotted as a function of the physics hyperparameter λ (Lambda) for a constant optimal learning rate of 9E-04.

6. CONCLUSION

We have presented a new time-variant machine learning model for retrieval of inertial spin rate and spin-axis from relative spin rate observations of debris and retired satellites. We have also introduced a novel physics-based loss function that is based on astronomy theory which can be used to train the model. We show initial results as evidence that incorporating a physics-based constraint in the model's learning objective can potentially train models faster, generate models that generalize more effectively to unseen data instances, and generate models that are consistent with physical theory. We believe that this paper demonstrates that the use of physics-guided models can improve models deployed for many different problems in the space domain awareness community that are hindered by either a lack of training data or by non-generalizability of model predictions. In particular, the improved generalizability of our physics-guided model's predictions is promising because it opens the door to generating models that can be continually trained in a physically-constrained manner as data streams onto open source data libraries such as the Unified Data Library (UDL).

Our future work will focus on extending our spin parameter retrieval tool towards an end-to-end system for non-cooperative satellite assessment. In particular, this study will link up well with our previous study on satellite stability assessment in which we found that determining accurate spin rate was critical for properly clasifying non-stable satellites [36]. Another future direction will be to investigate if the use of different angle representations such as quaternions can improve the ability for our time-variant machine learning framework to retrieve the spin rate of orbiting debris. This has been previously suggested as a direction for retrieval of satellite pose from spectroscopy datasets [19, 28]. Finally, we will work on creating both novel physics-guided loss functions for not only the spin retrieval problem but many other satellite assessment problems that can benefit from the merging of physical theory and machine learning theory.

REFERENCES

[1] Joseph R Kopacz, Roman Herschitz, and Jason Roney. Small satellites an overview and assessment. *Acta Astronautica*, 170:93–105, 2020.

[2] Stephen A Jacklin. Small-satellite mission failure rates. Technical report, 2019.

[3] Thomas Schildknecht. Optical surveys for space debris. *The Astronomy and Astrophysics Review*, 14(1):41–111, 2007.

[4] Michael D Abercrombie, Brandoch Calef, and Shadi Naderi. Light curve analysis of deep space objects in complex rotation states. 2021.

[5] Sisi Zhao, Michael Steindorfer, Georg Kirchner, Yongchao Zheng, Franz Koidl, Peiyuan Wang, Weidong Shang, Jinghao Zhang, and Tong Li. Attitude analysis of space debris using slr and light curve data measured with single-photon detector. *Advances in Space Research*, 65(5):1518–1527, 2020.

[6] James Cameron Bennett, Jizhang Sang, CH Smith, and Kefei Zhang. Accurate orbit predictions for debris orbit manoeuvre using ground-based lasers. *Advances in Space Research*, 52(11):1876–1887, 2013.

[7] Phillip M Cunio, Michael Bantel, Brien R Flewelling, William Therien, Mark W Jeffries Jr, Monica Montoya, Rhett Butler, and Douglas Hendrix. Photometric and other analyses of energetic events related to 2017 geo rso anomalies. In *Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, 2017.

[8] James C Jones and Michael Strong. Modeling small orbital debris remediation in low earth orbit. *The Advanced Maui Optical and Space Surveillance Technologies Conference*, 2022.

[9] Chen Song, Hou-Yuan Lin, and Chang-Yin Zhao. Analysis of envisat's rotation state using epoch method. *Advances in Space Research*, 66(11):2681–2688, 2020.

[10] Doyle Hall and Paul Kervin. Optical characterization of deep-space object rotation states. Technical report, AIR FORCE RESEARCH LAB KIHEI MAUI HI DETACHMENT 15, 2014.

[11] Per Magnusson, M Antonietta Barucci, Jack D Drummond, Kari Lumme, Steven J Ostro, Jean Surdej, RC Taylor, and V Zappala. Determination of pole orientations and shapes of asteroids. *Asteroids II*, pages 66–97, 1989.

[12] Per Magnusson. Distribution of spin axes and senses of rotation for 20 large asteroids. *Icarus*, 68(1):1–39, 1986.

[13] Doyle Hall, John Africano, David Archambeault, Brian Birge, David Witte, and Paul Kervin. Amos observations of nasa's image satellite. In *The 2006 AMOS Technical Conference Proceedings*, pages 10–14, 2006.

[14] Gregory P Badura, Christopher R Valenta, and Brian Gunter. Convolutional neural networks for inference of space object attitude status. *The Journal of the Astronautical Sciences*, 69(2):593–626, 2022.

[15] Roberto Furfaro, Richard Linares, and Vishnu Reddy. Shape identification of space objects via light curve inversion using deep learning models. In *AMOS Technologies Conference, Maui Economic Development Board, Kihei, Maui, HI*, 2019.

[16] Richard Linares and Roberto Furfaro. Space object classification using deep convolutional neural networks. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1140–1146. IEEE, 2016.

[17] Keiran McNally, Diego Ramirez, Alfredo M Anton, Duncan Smith, and James Dick. Artificial intelligence for space resident objects characterisation with lightcurves. In *8th European Conference on Space Debris*, 2021.

[18] Ian McQuaid, Laurence D Merkle, Brett Borghetti, Richard Cobb, and Justin Fletcher. Space object identification using deep neural networks. In *The Advanced Maui Optical and Space Surveillance Technologies Conference*, page 5, 2018.

[19] Matthew Phelps, J Zachary Gazak, Thomas Swindle, Justin Fletcher, and Ian Mcquaid. Inferring space object orientation with spectroscopy and convolutional networks. *AMOS (Sept. 2021)*, 2021.

[20] Xia Wang, YuRong Huo, YuQiang Fang, Feng Zhang, and Yifan Wu. Arsrnet: accurate space object recognition using optical cross section curves. *Applied Optics*, 60(28):8956–8968, 2021.

[21] Arka Daw, Anuj Karpatne, William D Watkins, Jordan S Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. In *Knowledge-guided machine learning*, pages 353–372. Chapman and Hall/CRC, 2017.

[22] Andrea Scorsoglio, Luca Ghilardi, and Roberto Furfaro. A physic-informed neural network approach to orbit determination. *The Journal of the Astronautical Sciences*, 70(4):1–30, 2023.

[23] Herbert Goldstein, Ch Poole, and J Safko. Classical mechanics addison-wesley. *Reading, MA*, 426, 1980.

[24] James Stewart. *Calculus*. Cengage Learning, 2015.

[25] Yann LeCun, Leon Bottou, Genevieve B Orr, Klaus-Robert Müller, et al. Neural networks: Tricks of the trade. *Springer Lecture Notes in Computer Sciences*, 1524(5-50):6, 1998.

[26] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[27] Jinlong Wu, Xiaolong Yin, and Heng Xiao. Seeing permeability from images: fast prediction with convolutional neural networks. *Science bulletin*, 63(18):1215–1222, 2018.

[28] Matthew Phelps, Thomas Swindle, J Zachary Gazak, Justin Fletcher, and Andrew Vandenberg. Improving spectral-based estimation of space object orientation. *Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*, 2022.
[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[30] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.
[31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
[32] Marija Agatonovic, Zoran Stankovic, Ivan Milovanovic, Nebojsa Doncov, Leen Sit, Thomas Zwick, and Bratislav Milovanovic. Efficient neural network approach for 2d doa estimation based on antenna array measurements. *Progress In Electromagnetics Research*, 137:741–758, 2013.
[33] G Peter Zhang, B Eddy Patuwo, and Michael Y Hu. A simulation study of artificial neural networks for nonlinear time-series forecasting. *Computers & Operations Research*, 28(4):381–396, 2001.
[34] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.
[35] Brett Naul, Joshua S Bloom, Fernando Pérez, and Stéfan van der Walt. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2(2):151–155, 2018.
[36] Gregory P Badura, Christopher R Valenta, Layne Churchill, and Douglas A Hope. Recurrent neural network autoencoders for spin stability classification of irregularly sampled light curves. *The Advanced Maui Optical and Space Surveillance Technologies Conference*, 2022.
[37] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
[38] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.
[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Appendices

A. DERIVATION OF THE PHASE ANGLE BISECTOR VELOCITY

In this section, we derive the vector $\dot{\mathbf{b}}^{\mathbf{I}} = \partial \mathbf{b}^{\mathbf{I}}/\partial t$ that is necessary to compute the PAB azimuthal velocity in Equation 11. We begin by considering that the terms in Equation 1 can be written as $\mathbf{b_I} = \mathbf{h}/k$ where $\mathbf{h} = \mathbf{o_I} + \mathbf{s_I}$ and $k = ||\mathbf{o_I} + \mathbf{s_I}||$.

We then note that the "Quotient Rule" of differentiation can be applied in order to compute the partial derivative with respect to t [24]. This is broken down via the following set of equations:

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial t} &= \dot{\mathbf{o}}_\mathbf{I} + \dot{\mathbf{s}}_\mathbf{I} \\ \frac{\partial k}{\partial t} &= \frac{(\mathbf{o_I} + \mathbf{s_I})^T (\dot{\mathbf{o}}_\mathbf{I} + \dot{\mathbf{s}}_\mathbf{I})}{||\mathbf{o_I} + \mathbf{s_I}||} \end{aligned} \tag{31}$$

These components are then combined to calculate the desired partial derivative with respect to time [24]:

$$\frac{\partial \mathbf{b}^\mathbf{I}}{\partial t} = \frac{\left(\frac{\partial \mathbf{h}}{\partial t}\right) k - \left(\frac{\partial k}{\partial t}\right) \mathbf{h}}{k^2} \tag{32}$$

Plugging in terms and simplifying, we obtain the final result for the PAB velocity $\dot{\mathbf{b}}^\mathbf{I}$:

$$\frac{\partial \mathbf{b}^\mathbf{I}}{\partial t} = \left(\frac{1}{||\mathbf{o_I} + \mathbf{s_I}||^3}\right) \Big((\mathbf{o_I} + \mathbf{s_I})^T (\mathbf{o_I} + \mathbf{s_I}) (\dot{\mathbf{o}}_\mathbf{I} + \dot{\mathbf{s}}_\mathbf{I}) - (\mathbf{o_I} + \mathbf{s_I})^T (\dot{\mathbf{o}}_\mathbf{I} + \dot{\mathbf{s}}_\mathbf{I}) (\mathbf{o_I} + \mathbf{s_I}) \Big) \tag{33}$$

The PAB velocity can therefore be directly computed via position and velocity information that is directly observable from Two-Line Element (TLE) data and solar ephemeris information.

B. DERIVATION OF EXACT FORM OF THE PHASE ANGLE BISECTOR AZIMUTHAL VELOCITY

We begin by defining a variable A using components from Equation 12:

$$A = \frac{b_x^I \cos\phi + b_y^I \sin\phi}{\sqrt{1 - \left(b_x^I \sin\phi \sin\theta - b_y^I \cos\phi \sin\theta + b_z^I \cos\theta\right)^2}} \tag{34}$$

In order to simplify the calculation of partial derivatives, we also introduce two vectors and re-write A using vector notation:

$$\mathbf{c}^T = [\cos\phi, \sin\phi, 0] \tag{35}$$

$$\mathbf{d}^T = [\sin\phi \sin\theta, -\cos\phi \sin\theta, \cos\theta] \tag{36}$$

$$A = \frac{\mathbf{b^I}^T \mathbf{c}}{\sqrt{1 - (\mathbf{b^I}^T \mathbf{d})(\mathbf{d}^T \mathbf{b^I})}} = \frac{\mathbf{b^I}^T \mathbf{c}}{\sqrt{1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2}} \tag{37}$$

By substituting the value A into Equation 12, the equation becomes $\Psi(t,\theta,\phi) = \arccos(A)$. The partial derivative with respect to the vector $\mathbf{b^I}$ can then be derived by applying the "Chain Rule" of calculus and using the theorem that the partial derivative of $\arccos x = (-1/\sqrt{1-x^2})$ [24]:

$$\frac{\partial \Psi(t,\theta,\phi)}{\partial b_x^I} = \left(\frac{-1}{\sqrt{1-A^2}}\right) \frac{\partial A}{\partial b_x^I} \tag{38}$$

$$\frac{\partial \Psi(t,\theta,\phi)}{\partial b_y^I} = \left(\frac{-1}{\sqrt{1-A^2}}\right) \frac{\partial A}{\partial b_y^I} \tag{39}$$

$$\frac{\partial \Psi(t,\theta,\phi)}{\partial b_z^I} = \left(\frac{-1}{\sqrt{1-A^2}}\right) \frac{\partial A}{\partial b_z^I} \tag{40}$$

We can observe that the leading multiplicative term on the right side of these equations are all equal to $\left(-1/\sqrt{1-A^2}\right)$. Therefore, the only components that remain to be evaluated equations are the partial derivatives of A with respect to the vector $\mathbf{b^I}$. In the following sub-sections, we perform these calculations and finally combine all components in order to compute an exact form of Equation 11.

B.1 Partial Derivative of "A" with Respect to b_x^I

We begin by noting that the partial-derivative of the vector $\mathbf{b^I}$ with respect to the x-axis of the inertial frame simplifies to the x unit-vector:

$$\frac{\partial \mathbf{b^I}}{\partial b_x^I} = [1, 0, 0]^T = \hat{\mathbf{x}} \tag{41}$$

We then note that the "Quotient Rule" of differentiation can be applied to Equation 37 to compute the partial derivative with respect to b_x^I [24]. This is broken down via the following set of equations:

$$\begin{aligned} f &= \mathbf{b^I}^T \mathbf{c} \\ g &= \sqrt{1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2} \\ \frac{\partial f}{\partial b_x^I} &= \hat{\mathbf{x}}^T \mathbf{c} \\ \frac{\partial g}{\partial b_x^I} &= -\left(\hat{x}^T \mathbf{d}\right)\left(\mathbf{d}^T \mathbf{b^I}\right)\left(1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2\right)^{-\frac{1}{2}} \end{aligned} \tag{42}$$

The "quotient rule" combines these components to calculate the desired partial derivative [24]:

$$\frac{\partial A}{\partial b_x^I} = \frac{\left(\frac{\partial f}{\partial b_x^I}\right) g - \left(\frac{\partial g}{\partial b_x^I}\right) f}{g^2} = \frac{\left(\hat{\mathbf{x}}^T \mathbf{c}\right)\left(1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2\right) + \left(\mathbf{b^I}^T \mathbf{c}\right)\left(\hat{\mathbf{x}}^T \mathbf{d}\right)\left(\mathbf{d}^T \mathbf{b^I}\right)}{\left(1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2\right)^{3/2}} \tag{43}$$

2.2 Partial Derivative of "A" with Respect to b_y^I

The partial derivative with respect to the other axes proceeds in a similar manner. The partial-derivative of the vector $\mathbf{b^I}$ with respect to the y-axis of the inertial frame simplifies to the y unit-vector:

$$\frac{\partial \mathbf{b^I}}{\partial b_y^I} = [0, 1, 0]^T = \hat{\mathbf{y}} \tag{44}$$

The partial derivatives of the scalar terms f and g from Equation 42 can then be written via the following equations:

$$\begin{aligned} \frac{\partial f}{\partial b_y^I} &= \hat{\mathbf{y}}^T \mathbf{c} \\ \frac{\partial g}{\partial b_y^I} &= -\left(\hat{\mathbf{y}}^T \mathbf{d}\right)\left(\mathbf{d}^T \mathbf{b^I}\right)\left(1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2\right)^{-\frac{1}{2}} \end{aligned} \tag{45}$$

The "Quotient Rule" combines these components to approximate the desired partial derivative as:

$$\frac{\partial A}{\partial b_y^I} = \frac{\left(\frac{\partial f}{\partial b_y^I}\right) g - \left(\frac{\partial g}{\partial b_y^I}\right) f}{g^2} = \frac{\left(\hat{\mathbf{y}}^T \mathbf{c}\right)\left(1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2\right) + \left(\mathbf{b^I}^T \mathbf{c}\right)\left(\hat{\mathbf{y}}^T \mathbf{d}\right)\left(\mathbf{d}^T \mathbf{b^I}\right)}{\left(1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2\right)^{3/2}} \tag{46}$$

2.3 Partial Derivative of "A" with Respect to b_z^I

Once again, we note that the partial derivative of the vector $\mathbf{b^I}$ with respect to the z-axis of the inertial frame simplifies to the z unit-vector:

$$\frac{\partial \mathbf{b^I}}{\partial b_z^I} = [0, 0, 1]^T = \hat{\mathbf{z}} \tag{47}$$

When deriving the necessary components to apply the chain rule, it is found that one of the terms drops to 0:

$$\begin{aligned} \frac{\partial f}{\partial b_z^I} &= \hat{\mathbf{z}}^T \mathbf{c} = 0 \\ \frac{\partial g}{\partial b_z^I} &= -\left(\left(\hat{\mathbf{z}}^T \mathbf{d}\right)\left(\mathbf{d}^T \mathbf{b^I}\right)\right)\left(1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2\right)^{-\frac{1}{2}} \end{aligned} \tag{48}$$

The "Quotient Rule" once again combines these components to compute the partial derivative as:

$$\frac{\partial A}{\partial b_z^I} = \frac{\left(\frac{\partial f}{\partial b_z^I}\right) g - \left(\frac{\partial g}{\partial b_z^I}\right) f}{g^2} = \frac{-\left(\frac{\partial g}{\partial b_z^I}\right) f}{g^2} = \frac{\left(\mathbf{b^I}^T \mathbf{c}\right)\left(\hat{\mathbf{z}}^T \mathbf{d}\right)\left(\mathbf{d}^T \mathbf{b^I}\right)}{\left(1 - \left(\mathbf{b^I}^T \mathbf{d}\right)^2\right)^{3/2}} \tag{49}$$

2.4 Combining All Equations

It can be seen that the denominator of Equations 43, 46, and 49 are equal after applying simplifications. It is therefore desired to simplify the term $\left(-1/\sqrt{1-A^2}\right)$ from Equations 38, 39, and 40 in terms of this denominator. It can be shown that this term can be simplified in the following manner:

$$\frac{-1}{\sqrt{1-A^2}} = \frac{-1}{\sqrt{1-\left(\frac{\mathbf{b^I}^T\mathbf{c}}{\sqrt{1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2}}\right)^2}} \tag{50}$$

$$\frac{-1}{\sqrt{1-A^2}} = \frac{-1}{\sqrt{\left(\frac{1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2}{1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2}\right)-\left(\frac{\left(\mathbf{b^I}^T\mathbf{c}\right)^2}{1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2}\right)}} \tag{51}$$

$$\frac{-1}{\sqrt{1-A^2}} = \frac{-\left(1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2\right)^{1/2}}{\sqrt{1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2-\left(\mathbf{b^I}^T\mathbf{c}\right)^2}}, \tag{52}$$

where it can be seen from the last equation that we have successfully isolated a term that contains the denominator of Equations 43, 46, and 49. Finally, by combining Equation 52 with Equations 43, 46, and 49, we are able to obtain the desired form of the required Equations 38, 39, and 40:

$$\frac{\partial\Psi(t,\theta,\phi)}{\partial b_x^I} = -\frac{\left(\hat{\mathbf{x}}^T\mathbf{c}\right)\left(1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2\right)+\left(\mathbf{b^I}^T\mathbf{c}\right)\left(\hat{\mathbf{x}}^T\mathbf{d}\right)\left(\mathbf{d}^T\mathbf{b^I}\right)}{\left(1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2-\left(\mathbf{b^I}^T\mathbf{c}\right)^2\right)^{1/2}\left(1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2\right)} \tag{53}$$

$$\frac{\partial\Psi(t,\theta,\phi)}{\partial b_y^I} = -\frac{\left(\hat{\mathbf{y}}^T\mathbf{c}\right)\left(1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2\right)+\left(\mathbf{b^I}^T\mathbf{c}\right)\left(\hat{\mathbf{y}}^T\mathbf{d}\right)\left(\mathbf{d}^T\mathbf{b^I}\right)}{\left(1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2-\left(\mathbf{b^I}^T\mathbf{c}\right)^2\right)^{1/2}\left(1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2\right)} \tag{54}$$

$$\frac{\partial\Psi(t,\theta,\phi)}{\partial b_z^I} = \frac{-\left(\mathbf{b^I}^T\mathbf{c}\right)\left(\hat{\mathbf{z}}^T\mathbf{d}\right)\left(\mathbf{d}^T\mathbf{b^I}\right)}{\left(1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2-\left(\mathbf{b^I}^T\mathbf{c}\right)^2\right)^{1/2}\left(1-\left(\mathbf{b^I}^T\mathbf{d}\right)^2\right)} \tag{55}$$