Earth and Space Science

METHOD

10.1029/2023EA002965

Special Collection:
Machine Learning in Heliophysics

A Novel Method to Train Classification Models for Structure Detection in In Situ Spacecraft Data

K. Bergstedt[1,2] and H. Ji[1,2]

[1]Department of Astrophysical Sciences, Princeton University, Princeton, NJ, USA, [2]Princeton Plasma Physics Laboratory, Princeton, NJ, USA

Abstract We present a method for creating spacecraft-like data which can be used to train Machine Learning (ML) models to detect and classify structures in in situ spacecraft data. First, we use the Grad-Shafranov equation to numerically solve for several magnetohydrostatic equilibria which are variations on a known analytic equilibrium. These equilibria are then used as the initial conditions for Particle-In-Cell simulations in which the structures of interest are observed and labeled. We then take one-dimensional slices through the simulations to replicate what a spacecraft collecting data from the simulation would observe. This sliced data then can be used as training data for the initial training of ML models intended for use on spacecraft data. We demonstrate the method applied to the problem of detecting small-scale plasmoids in the magnetotail, which is important for understanding complex magnetotail reconnection dynamics. The simple 1D classifier we train is able to detect more than 70% of the plasmoid points in the data set but also produces a large number of false positives. Our further work on this example problem is detailed, and further potential uses of the method are discussed.

Key Points:

- We present a method to use analytic magnetohydrostatic equilibria to develop perturbed numerical equilibria to initialize simulations
- We present a method to use Particle-In-Cell simulation data to simulate in situ spacecraft data, for use in training Machine Learning models
- The above techniques are demonstrated by training a 1D Convolutional Neural Network to detect small-scale plasmoids in the magnetotail

Correspondence to:
K. Bergstedt,
kbergste@pppl.gov

Citation:
Bergstedt, K., & Ji, H. (2024). A novel method to train classification models for structure detection in in situ spacecraft data. *Earth and Space Science*, *11*, e2023EA002965. https://doi.org/10.1029/2023EA002965

Received 31 MAR 2023
Accepted 18 MAY 2024

Author Contributions:
Conceptualization: K. Bergstedt, H. Ji
Data curation: K. Bergstedt
Formal analysis: K. Bergstedt
Funding acquisition: H. Ji
Investigation: K. Bergstedt
Methodology: K. Bergstedt
Project administration: H. Ji
Resources: K. Bergstedt
Software: K. Bergstedt
Supervision: H. Ji
Validation: K. Bergstedt
Visualization: K. Bergstedt
Writing – original draft: K. Bergstedt
Writing – review & editing: K. Bergstedt, H. Ji

1. Introduction

In situ spacecraft data is an important component of space physics research. In situ instruments that are directly immersed in the local plasma environment are able to measure quantities which are difficult or impossible to obtain via remote measurement. Interpretation of those measured quantities, however, is not straightforward. A satellite instrument collects time series data, meaning it samples the data at discrete points in time. At the same time, the spacecraft may be moving significantly relative to the local plasma environment; for example, structures entrained in a magnetic reconnection outflow can move past the spacecraft at high speed. Additionally, while a spacecraft may be able to take data at different physical locations on the spacecraft, the spacecraft is generally very small compared to the plasma structures it is taking data from. Therefore, a single spacecraft's data is a 1D snapshot of the full 4D picture of a three dimensional plasma evolving with time. This makes interpretation of spacecraft data difficult despite the high quality of the local measurements. As a consequence, methodologies for analysis of in situ spacecraft data are sophisticated and varied. For example, there is a wide range of techniques for single and multiple spacecraft that aim to determine the geometry and/or orientation of observed plasma structures (e.g., Fu et al., 2015; Paschmann & Daly, 1998; Shi et al., 2006; Sonnerup & Cahill, 1967; Sonnerup et al., 2006; Torbert et al., 2020). Still, however, much existing spacecraft data has difficult or ambiguous interpretation, as existing methods cannot always conclusively determine whether an observed structure is spatial, temporal, or both.

As Machine Learning (ML) techniques have risen in prominence, there have been a number of successful attempts to apply them to the interpretation and analysis of in situ spacecraft data (e.g., Argall et al., 2020; Breuillard et al., 2020; dos Santos et al., 2020; Garton et al., 2021; Olshevsky et al., 2021). There have been successful attempts to use unsupervised learning for plasma structure identification (Innocenti et al., 2021), but the problem of identifying plasma structures within the data translates naturally to a classification problem, which can be tackled via supervised learning. Since supervised learning requires a training data set where the "correct" answer is known, attempts to utilize it in the past have either relied on training data sets comprising spacecraft data which has already been classified via a different method (e.g., Argall et al., 2020; Garton et al., 2021; Lenouvel et al., 2021) or comprising modeled or simulated data (e.g., dos Santos et al., 2020). Models of the former kind can be particularly useful in reducing workload on researchers in cases where plasma structure identification is done via meticulous visual inspection. In this work we have opted for the latter and developed a framework for training models based on simulated data. This choice allows the models to achieve potentially better performance than existing identification methods, including visual inspection.

There are a number of potential pitfalls to consider when training models on simulated data. The success of a model depends hugely on the quality of its input data. Simulated data has less fluctuations and variability than real space; for example, perfectly flat, uniformly dense current sheets are not found in nature. Care must be taken to ensure that the simulated input data set is sufficiently representative of real space, and even then domain adaptation methods such as those described in Singhal et al. (2023) should be considered to optimize performance on real data. In order to tackle this problem, we have developed a method of solving perturbed Grad-Shafranov (GS) equations to create "messy" magnetostatic equilibria that can be used as initial conditions in Particle-In-Cell (PIC) simulations, and have then taken 1D slices through these perturbed simulations to synthesize an input data set suitable for training a classification model. We demonstrate this method using the example case of training a 1D Convolutional Neural Network (CNN)-based classifier to identify plasmoids in reconnection regions. Note that this approach is one of many: there are other ways to increase the realism of simulation data, such as applying realistic instrument noise and using the same spatial resolution as an existing instrument. Such methods have been used with great success by for example, Tremblay et al. (2018) but they are beyond the scope of this report.

1.1. Motivation for Our Example Case: Detecting Plasmoids in the Magnetotail

Earth's magnetotail is an elongated plasma environment, spanning 20–60 Earth radii (Re) radially in the Y-Z Geocentric Solar Magnetospheric (GSM) plane but up to hundreds of Re in the -X GSM direction away from the Sun. The tail is often described in terms of near (∼<20 Re), mid (∼20–60 Re), and far (∼>60 Re) regions, though there are no precise definitions for these terms. Magnetic reconnection in the near-to-mid tail is often bursty, significantly influenced by internal magnetosphere dynamics in addition to direct driving, and allows finite guide fields (Petrukovich et al., 2016). Additionally, near-to-mid-tail reconnection is often associated with dipolarizations during substorms. The complex reconnection situation and the association with global magnetospheric changes makes the phenomenon important to study.

Another key feature of interest is the presence of plasmoids—regions with loop- or helix-shaped magnetic field lines—which may have been formed via the plasmoid instability. Small-scale plasmoids have been observed in the tail via in situ measurements many times (e.g., Chen et al., 2008, 2012). If near-to-mid tail reconnection is capable of producing many coexisting small-scale plasmoids, those plasmoids will have a significant impact on reconnection dynamics and energization efficiency in the region. Statistical properties of plasmoid dynamics of reconnecting current sheets are important to quantify multiscale reconnection in large plasmas in general (Ji et al., 2022).

Researchers have used a number of methods to detect plasmoids in in situ data in the past. Automated methods are repeatable, rigorous, and are less susceptible to human bias. For example, there are well-performing methods that automatically detect flux ropes in satellite data (Huang et al., 2018; Smith et al., 2017). These two methods can determine valuable information such as the flux rope's radius. However, both methods fit the spacecraft data to cylindrical flux rope models, force-free (Lundquist, 1950) and non-force-free (Elphic & Russell, 1983) respectively. These methods may not perform optimally in dynamic reconnection regions where large numbers of plasmoids are squashed or deformed from cylindrical shapes and the structure's magnetic field and the turbulent background field are difficult to distinguish from each other. Simpler geometry-based algorithms have also been developed (Bergstedt et al., 2020), but their generality invites the possibility of large numbers of false positives and they cannot distinguish whether they are detecting a plasmoid or the traveling compression region surrounding it.

Researchers will also often rely on their experience to identify plasmoids in data "by eye." This approach works well for case studies focused on one or two plasmoids and their nearby environment where all the nuances of the plasma's characteristics must be understood; in contrast, this approach is a poor choice for statistical studies of many plasmoids due to the method's inherent subjectivity and excessive consumption of researcher time. There have been numerous groundbreaking case studies which have advanced our knowledge of magnetosphere dynamics (such as Chen et al. (2008) among countless others), but a larger scale statistical study is needed in order to investigate broad multiscale phenomena such as the connection between reconnection dynamics and global magnetosphere dynamics. Therefore, we have significant motivation to develop a ML model to improve upon existing plasmoid detection techniques.

2. Constructing Perturbed Magnetostatic Equilibria

We wish to start our PIC simulations in hydrostatic equilibrium to avoid initially injecting large amounts of energy into the simulation. The GS partial differential equation (PDE) is a convenient description of a 2D magnetohydrodynamic (MHD) equilibrium:

$$\nabla^2 A_y = -\mu_0 \frac{d}{dA_y}\left(p + \frac{B_y^2}{2\mu_0}\right) \tag{1}$$

Here we are assuming a geometry in which $\hat{y}$ is the invariant direction. A_y is the $\hat{y}$ component of the magnetic vector potential, and is also known as the flux function in the case of 2D plasma. Additionally, p is the gas pressure, B_y is the $\hat{y}$ component of the magnetic field, and μ_0 is the vacuum permeability. The GS equation is notably used in GS reconstruction, which is a method used to extrapolate the geometry of quasi-magnetostatic, 2D structures from spacecraft data (Sonnerup et al., 2006). The solutions of partial differential equations are not generally well behaved and will often exhibit bifurcations where a continuous change of parameters causes a discontinuous change in the solution (Arnol'd, 1994). However, often a small perturbation of the right hand side (RHS) of the equation will result in a solution which is qualitatively close to the unperturbed analytic solution, and it is easy and quick to discard numeric solutions with differing behaviors via visual inspection.

The GS equation is a Poisson equation and therefore is solved as a boundary value problem. This can cause problems when trying to numerically solve the GS equation in space physics applications where the problem is unbounded or has boundary conditions at infinity (such as the magnetic field vanishing there). There exist methods to impose boundaries at infinity onto numerical solutions (Wang, 1999), though we do not use them here for simplicity. In the context of solving the GS equation to find an approximate hydrostatic equilibrium which is close to a known analytic solution, we can impose finite boundary conditions by requiring that the perturbed solution match the analytic one at the boundary of the region of interest. It would also be possible to add further variability to the solution by using boundary conditions which are slightly perturbed from the analytic ones, but we have not explored this.

2.1. Example Case: A Perturbed Harris Sheet

We next must choose an analytic equilibrium to perturb from. PIC simulations of magnetic reconnection often are started in a 1D Harris sheet configuration as it is an exact kinetic equilibrium. It is notable that an exact kinetic equilibrium with a more magnetotail-like shape has been found (Birn et al., 1975; Schindler, 1972), but we elect to use a Harris sheet for simplicity. Note also that if we were interested in the physics of reconnection onset it would be best to use the magnetotail-like configuration because having a nonzero magnetic field component normal to the current sheet affects reconnection onset (Liu et al., 2014). A Harris sheet in the y-z plane with its magnetic field in the $\hat{z}$ direction would be defined as

$$\vec{B}(x) = B_0 \tanh\left(\frac{x}{d_0}\right)\hat{z} + B_g\,\hat{y}, \quad p = \frac{B_0^2}{2\mu_0}\operatorname{sech}^2\left(\frac{x}{d_0}\right) \tag{2}$$

Here B_0 is the maximum magnetic field strength, B_g is an optional constant guide field and d_0 is the characteristic thickness of the current sheet.

We next need to calculate the RHS of Equation 1 for this equilibrium. Since we have $B_y = B_g =$ constant, the magnetic field term will drop out leaving just the derivative of the pressure with respect to A_y. By integrating the definition of the magnetic vector potential and assuming a constant of integration of zero, we find

$$A_y = d_0 B_0 \log\left[\cosh\left(\frac{x}{d_0}\right)\right] \Rightarrow \cosh\left(\frac{x}{d_0}\right) = \exp\left(\frac{A_y}{d_0 B_0}\right) \tag{3}$$

It then straightforwardly follows that

$$p = \frac{B_0^2}{2\mu_0}\exp\left(\frac{-2A_y}{d_0 B_0}\right) \Rightarrow -\mu_0 \frac{d}{dA_y} p = \frac{B_0}{d_0}\exp\left(\frac{-2A_y}{d_0 B_0}\right) = \frac{B_0}{d_0}\operatorname{sech}^2\left(\frac{x}{d_0}\right) \tag{4}$$

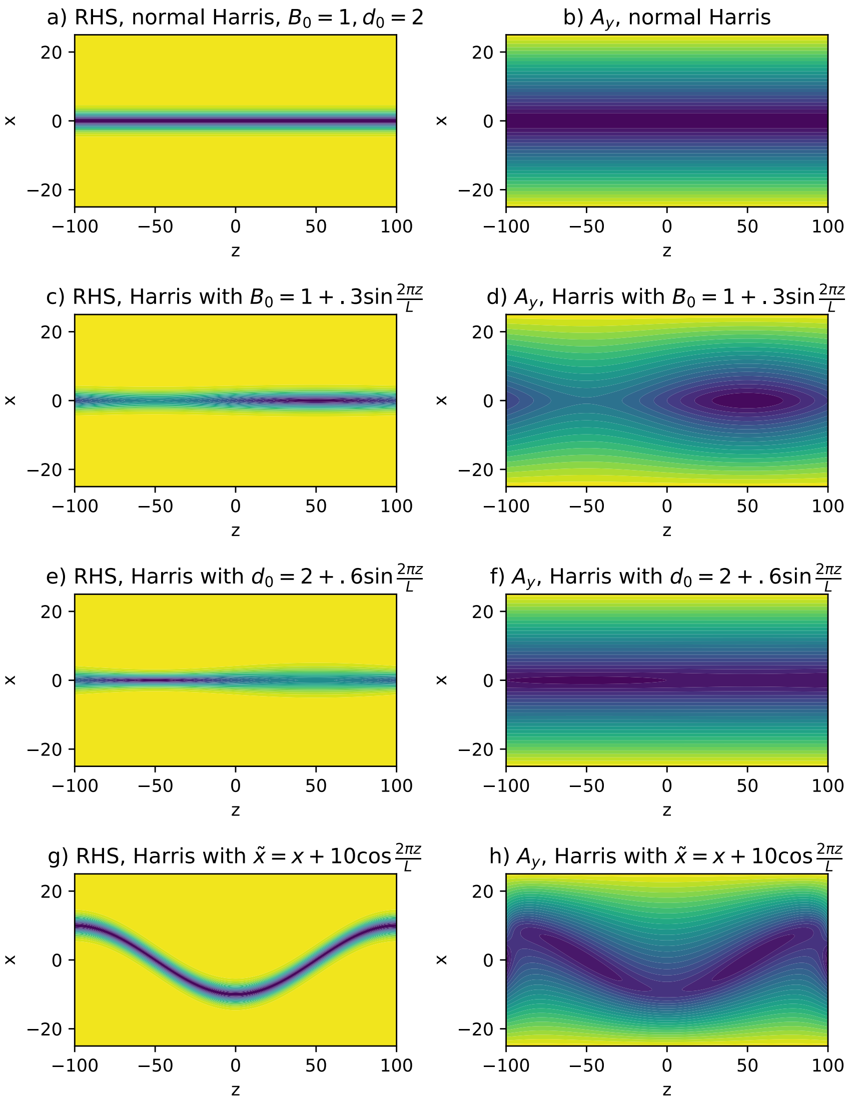

Figure 1. Examples of perturbations to the right hand side (RHS) of Equation 1 for a Harris sheet equilibrium and their resulting flux functions A_y. Panels (a and b) show the unperturbed Harris sheet and numerical solution. Panels (c and d) show a simple sinusoidal perturbation of B_0. The large plasmoid that appears in (d) from a seemingly small change to the RHS is of note. Panels (e and f) show a simple sinusoidal perturbation of d_0. Panels (g and h) showcase the substitution of a perturbed coordinate $\tilde{x}$ in the RHS equation. Note that the current sheet in (h) bends back to the x = 0 line at $z = \pm 100$ due to our enforcement of ideal Harris values at the boundaries.

Now we can modify this expression in order to perturb the GS solution away from an ideal Harris sheet. A natural place to start is changing the constants B_0 and d_0 to functions $B_0(x)$ and $d_0(x)$. If we were perturbing these functions in the expressions for $\overrightarrow{B}$ and p they would directly correspond to changing the magnetic field strength and width of the current sheet respectively; in contrast, when perturbing the RHS, these correspondences are often qualitatively present but are not exact due to the process of solving the PDE. Additionally, we can modify the expression by introducing a z-dependent deflection to the x coordinate, $x + f(z)$. This allows us to introduce bends and kinks into the current sheet. Again, due to the process of solving the PDE there isn't an exact correspondence between $x + f(z)$ and the location of the perturbed current sheet. Examples of perturbations to the RHS of Equation 1 and their effects on A_y are shown in Figure 1.

Now that we have our desired RHS we must solve the GS equation for A_y and then derive the remaining quantities of interest. In this paper we have elected to numerically solve the GS equation using a finite element method in the DOLFINX package (Alnaes et al., 2014, 2015; Logg & Wells, 2010; Scroggs et al., 2022), requiring matching with the ideal Harris sheet at the boundary. We use the same package to calculate B_z and B_x by differentiating A_y as per the definition of the magnetic vector potential, and calculate j_y similarly as per Ampere's law. Note that the other components of $\vec{j}$ are zero due to our assumptions of two-dimensionality and a constant guide field.

More subtlety and choice arises when determining pressure, density and temperature. We can calculate the pressure from the equilibrium force balance $\nabla p = \vec{j} \times \vec{B}$. Since our boundary conditions are at infinity, $p(x \to \pm\infty) = 0$, we again make do with matching with the ideal Harris sheet values at the finite solution boundaries. This process notably does not enforce $p \geq 0$, so we add a small constant to our numeric solution for p to preserve the nonnegativity of p everywhere. This is permitted by the equilibrium force balance which only involves the derivative of p. Density n and temperature T (in energy units) must obey the ideal gas law $p = nT$ and the equation of state $\frac{d}{dt}\left(\frac{p}{\rho^{\gamma}}\right) = 0$ where $\rho = \sum_{\alpha} m_{\alpha} n_{\alpha}$ is the total plasma mass density including all species α and γ is the adiabatic index. We make the isothermal assumption $\gamma = 1$ and assume a hydrogen-based plasma ($n_i = n_e = n/2$), and therefore can choose a fixed temperature T and use that and the pressure to determine the plasma density. With our field and particle values determined we are ready to use the equilibrium as the initial condition for a simulation.

3. Example Case: A Simulated Data Set for Plasmoid Detection

In this section, we continue our example of a potential use case for this methodology. We demonstrate the use of messy GS-constructed Harris-like equilibria as the starting point of simulations of plasmoids in current sheets, and the use of said simulations to train a simple plasmoid detection model. Our simulations are performed using the VPIC PIC code (Bowers, Albright, Bergen, et al., 2008; Bowers, Albright, Yin, et al., 2008; Bowers et al., 2009). Our 2D simulation space is a periodic $200d_i \times 50d_i$ box where d_i is the ion inertial length, and the simulation runs for 60 ion cyclotron times. We use an ion-to-electron mass ratio of 25, a cell size of 1.5 Debye lengths, and 600 plasma particles per cell. We use a timestep that is 98% of the Courant–Friedrichs–Lewy condition in order to ensure numerical convergence while minimizing the number of timesteps needed. The simulation uses a low plasma beta of 0.01, an electron velocity of 0.25c within the sheet, and a cooler background plasma population which has an electron velocity of 0.15c. Ion and electron temperatures are kept equal.

We keep all plasma parameters the same in all simulations for this demonstrative example, but models which will be applied to spacecraft data should be trained on simulations which have a broad spectrum of realistic plasma parameters. We elect to create variations in the initial equilibrium conditions solely by deflecting the x coordinate $x \to x + f(z)$ in the Harris sheet GS equation as described in Section 2.1. To choose a somewhat random but continuous $f(z)$ we use a Fourier series with 15 non-constant terms:

$$f(z) = \frac{a_0}{2} + \sum_{1}^{15} a_j \cos\left(2\pi j \frac{z}{L} + \phi_j\right) \tag{5}$$

The a_j are randomly chosen from a normal distribution with mean 0 and standard deviation of $2d_i$, the ϕ_j are randomly chosen from a uniform distribution with bounds $(0,2\pi)$ and $L = 200d_i$ is the extent of the simulation in the $\hat{z}$ direction. An example of one of the resulting equilibria is shown in Figure 2. The perturbation process introduces topological o-points to the current sheet which expand into distinct plasmoids as the simulation evolves.

While it would be possible to use the generated equilibria as initial conditions for full 3D simulations, we elect to use simpler 2D simulations in order to simplify the construction of our simulated data set. In 2D, the magnetic topology contains unambiguous magnetic nulls and separatrices which partition the plasma and provide definite boundaries for plasmoids. It is both possible and common to have multiple plasmoids contained within another larger plasmoid, but the smaller-scale plasmoids are more of interest for our goal of detecting plasmoids formed via the plasmoid instability in the magnetotail, so we seek to label solely the innermost plasmoids for our data set. A process for detecting hierarchies of plasmoids is described in Banesh et al. (2020).

The first step toward finding plasmoids in the simulation is finding topological o-points. Ignoring the potential constant guide field, magnetic nulls are places where both in-plane components of the magnetic field B_z and B_x vanish. We find the locations of these nulls by using the image processing library scikit-image (van der Walt

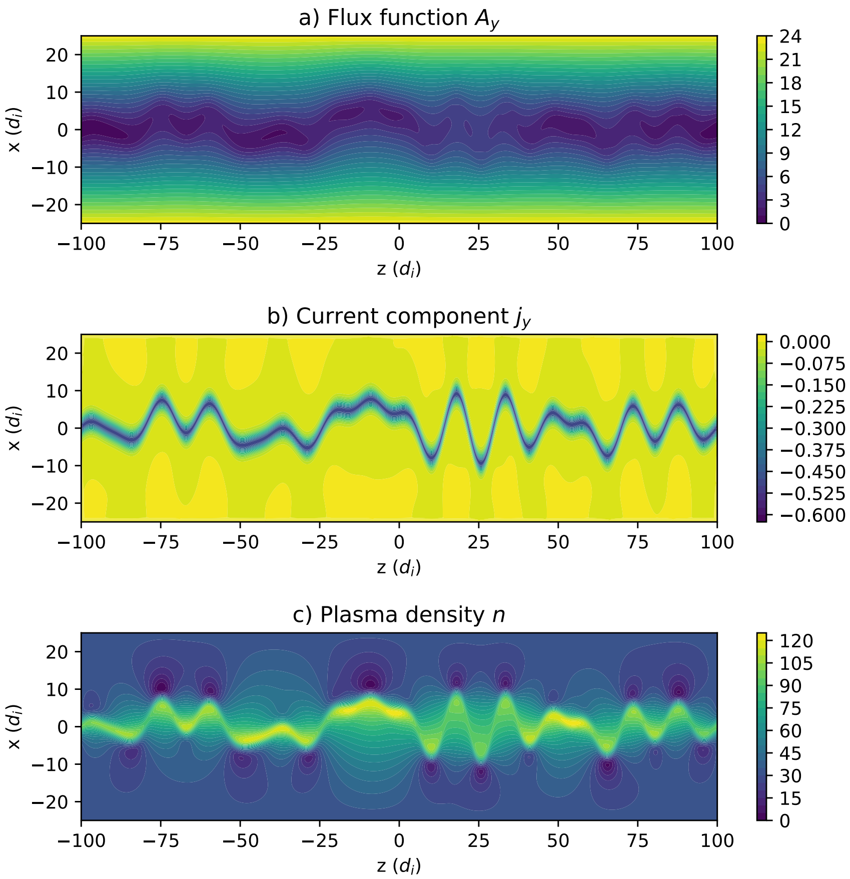

Figure 2. An example of the type of modified equilibrium constructed by adding Fourier components to the right hand side of Equation 1 and used as initial conditions to the particle-in-cell simulation. Panel (a) shows the value of the flux function A_y, where the boundaries of the colors are selected magnetic contours. The construction of the fluctuations in the current sheet has caused numerous topological o-points to form, seeding the current sheet with plasmoids. This is useful for increasing the sample size of plasmoids in the simulations and causes no issues because we are not studying reconnection onset specifically. Panel (b) depicts the current sheet itself. Panel (c) depicts the plasma density n. The regions of steep density gradients near the bends of the current sheet maintain force balance with the magnetic tension which is attempting to straighten the field lines.

et al., 2014) to find the contours where $B_z = 0$ as well as the contours where $B_x = 0$, and then locating their intersections. Gaussian filtering of the magnetic field data is first applied to minimize numerical noise. We do not round the location of contour intersections to the nearest grid point, instead using linear interpolation to calculate values away from the simulation mesh. We next must determine whether the observed nulls are topological o-points, which are defined as topological minima or maxima of the flux function A_y, or topological x-points, which are saddles. We can determine this by taking the determinant of the Hessian matrix **H** of A_y, which is defined as follows:

$$\mathbf{H} = \begin{pmatrix} \partial_{xx}A_y & \partial_{xz}A_y \\ \partial_{zx}A_y & \partial_{zz}A_y \end{pmatrix} = \begin{pmatrix} \partial_x B_z & -\partial_x B_x \\ \partial_z B_z & -\partial_z B_x \end{pmatrix} \Rightarrow \det\mathbf{H} = \partial_x B_x \cdot \partial_z B_z - \partial_x B_z \cdot \partial_z B_x \tag{6}$$

If det **H** is positive at a null point it is a topological o-point and if it is negative the null point is a topological x-point (Hubbard & Hubbard, 2009). We categorize the null points in our simulation in this fashion using numerical derivatives of the magnetic field components.

The separatrices which separate plasmoids from the rest of the simulation originate at x-points and have constant A_y, so we can find them by tracing level contours of A_y from each x-point. It is then straightforward to categorize all points surrounding an o-point and bounded by separatrices as plasmoids. Now that we have established the ground truth for our training, we assemble our spacecraft-like data by taking random 1D cuts through the simulation, linearly interpolating to find values off the mesh. We first crop the simulation to $\pm 20 d_e$ in the z direction to minimize the class imbalance between plasmoid and non-plasmoid points. We then determine the location of the cuts by taking as endpoints two random points within the cropped simulation space at a fixed time. The possible segment lengths are limited to $<200 d_i$ to shorten computation time. One hundred segments are created from each time, and the times we use are spaced five ion cyclotron times apart to ensure that the simulation has had time to evolve since the previous time slices were taken.

3.1. Training a 1D CNN Model

Now that we have our spacecraft-like data we are ready to train the model. The prototypical model structure used for classification problems is a 2D CNN (O'Shea & Nash, 2015). Models based on this structure have achieved excellent performance on classification tasks (e.g., Sharma et al., 2018). However, their usage of 2D inputs would necessitate significant modification of the input data. Two potential options directly suited to processing 1D data are Recurrent Neural Networks (RNNs) and 1D CNNs. RNNs are better documented and are well-suited to data with a temporal component (Medsker & Jain, 1999). 1D CNNs generally require less computations and have some recent evidence suggesting comparable performance to RNNs (Kiranyaz et al., 2021). For the purposes of this example we have chosen to use a 1D CNN architecture.

Besides deciding on the model's structure, we must also preprocess the data to be in a suitable state for training. We did 10 simulations, each starting from a different perturbed equilibrium. To avoid cross contamination we use seven simulations exclusively for training data and three exclusively for testing, a 70–30 training-testing split. We additionally add to the training and testing data set spacecraft-like data from a simulation of a current sheet at the same parameters sans any plasmoids. Our goal is to be able to predict if any given point in a spacecraft trajectory is inside of a plasmoid. The surrounding points and how the plasma parameters change there give some information about the local plasma structure, so we include them in the model input. To increase the speed of training, we configure the model to output predictions for 10 points simultaneously. To standardize the size of the inputs to the ML model we use a sliding window to break each "spacecraft trajectory" into segments 30 datapoints in length. The model does the prediction on the interior 10 points, using the exterior 20 for additional spatial information. We next randomly shuffle all of the segments so that the model can't pick up on patterns based on the segments' locations in the data set. We then undersample the majority class of the training data set (non-plasmoid points) by 85% to combat class imbalance. After this process is done, we are left with 1,170,830 total points in the training data set, 475,718 non-plasmoid points and 695,112 plasmoid points, a slight reversal of the imbalance. The testing data set, which was not undersampled, has 1,684,620 total points, 1,320,125 non-plasmoid and 364,495 plasmoid, an imbalance of approximately 3.5:1.

There are several quantities commonly available from spacecraft which could be fed to a ML model—some newer missions like the Magnetospheric Multiscale mission (MMS) even provide electron and ion phase space distribution functions (Pollock et al., 2016), which could be analyzed with 2D or 3D techniques (e.g., Olshevsky et al., 2021). For the purposes of this simple model, we will use a handful which are directly correspond to our understanding of general traits of plasmoids: B_x, B_y, B_z, j_y (which is often but not always elevated within plasmoids), and the plasma bulk velocity v_z to quantify the direction the plasmoid is moving within the sheet. Each quantity is passed through a convolutional and pooling layer to identify spatial features of interest. These data are then averaged together, then convolved and pooled once more to select the most relevant features. A final fully connected layer outputs 10 logits which predict whether the 10 central points of the inputted data segment are part of a plasmoid. More details are shown and discussed in Figure 3.

As this is merely an example model rather than a polished production one, we did not aggressively optimize hyperparameters or otherwise maximize model performance. We used cross entropy loss and the Adam optimizer, and only trained for 10 epochs with a learning rate of 0.01. Our training and testing performance is summarized in panel (b) of Figure 3. The model achieved a recall of 0.79 and a precision of 0.4 on the testing data set, meaning that 79% of the true positives were detected but only 40% of the predicted positives were actually positive. This is good recall, but the precision should be increased to increase the model's usability for scientific research. Additionally, good performance on the synthesized data set does not guarantee good performance on real spacecraft data.

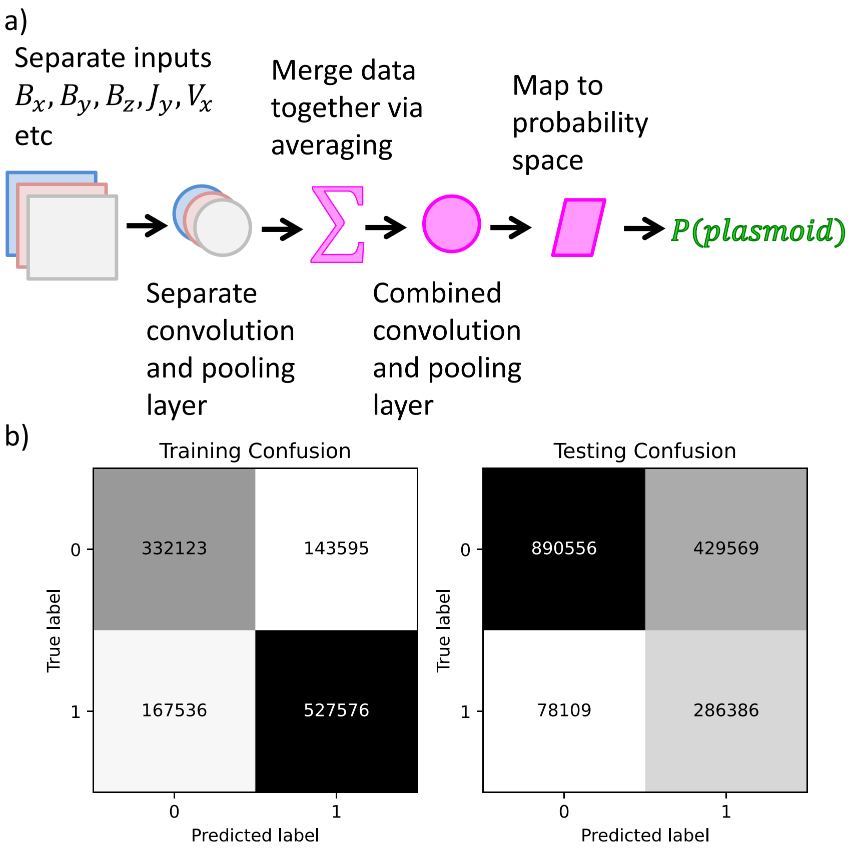

Figure 3. (a) A cartoon illustrating the structure of the simple 1D Convolutional Neural Network model we are training. All the convolutional layers use a size 3 kernel, are not padded, and use ReLU activation. The separate convolutional layers use 32 output channels while the combined convolutional layer uses 64. The pooling layers use max pooling with a pool size of 2. The averaging is done elementwise so the data maintains the same dimensions. The final steps in the classification process consist of a fully connected neural network layer and reshaping. The model outputs 10 logits which predict whether the center 10 points of the spacecraft trajectory segment are within plasmoids. (b) A confusion matrix summary of the results of training the model with 0 representing the negative (non-plasmoid) class and 1 representing the positive (plasmoid) class. Note that for the testing data set, which was not undersampled, the balance between false positives and true positives is nearly 2:1.

A visualization of the model's performance on some of our testing data is shown in Figure 4, and it depicts a notable phenomenon. When describing our plasmoid-detection process for the full 2D data, we noted that we would only find small-scale plasmoids using the method. Indeed, Figure 4 shows some structures which are clearly plasmoids by visual inspection but are not flagged as such by our separatrix-based algorithm. Our ML model often identifies the cores of these plasmoids as plasmoids despite the ground "truth" algorithm saying otherwise, which can be seen in the top panel of Figure 4. In a sense, for these plasmoids the ML model is "correct" and the separatrix-based algorithm is "incorrect." This is not a rigorous test of the model's capability, but it gives reason for optimism. We describe our next steps in the final section.

4. Final Remarks

In this paper we have described a novel method to train ML models to detect plasma structures in in situ spacecraft data using simulated training data. Simulations are done with "messy" initial conditions created by tweaking the RHS of the GS equation to better represent the variability of nature. Utilizing the complete information known about the simulated system, the desired structures are located and labeled in the simulation data. After this, 1D slices are taken from the simulations to replicate spacecraft data. These slices then are used to train a supervised ML model to detect the structures of interest.

We demonstrated the method by training a simple model to detect plasmoids in spacecraft-like data. There is a need for this kind of model because current methods of plasmoid detection are not well suited to detecting the

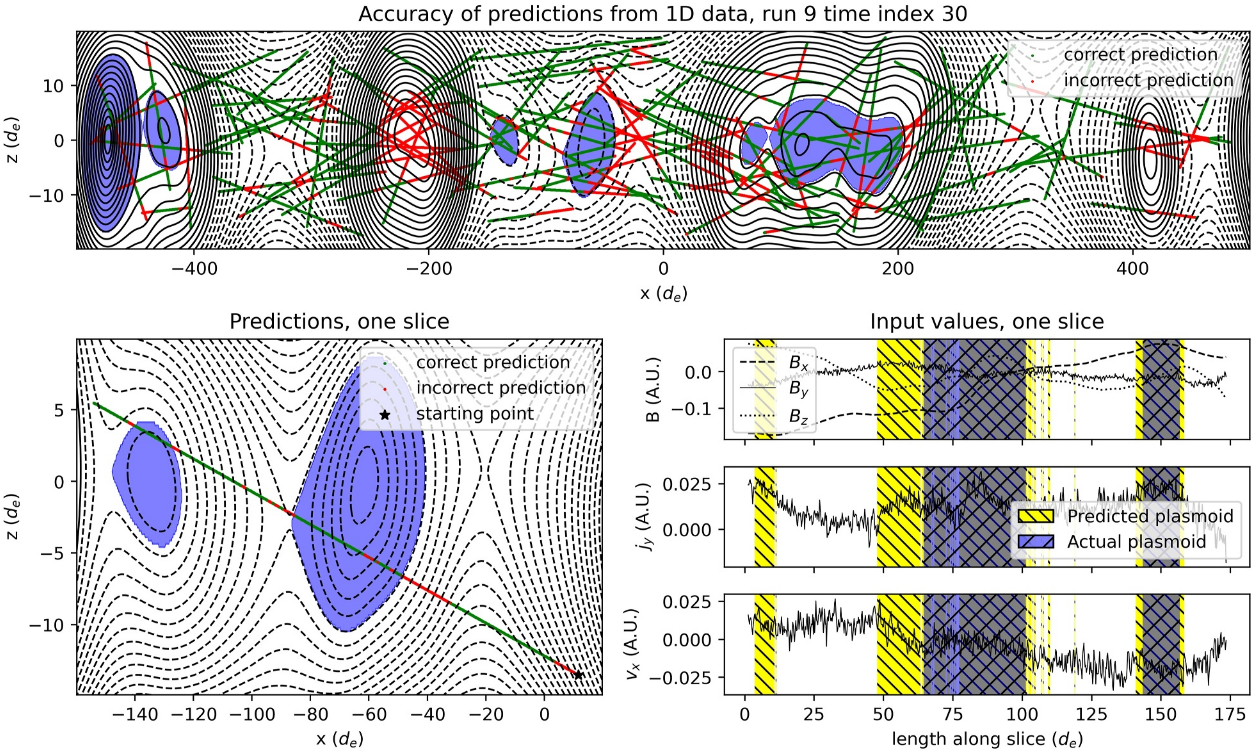

Figure 4. Top: Slices plotted over flux contours from the simulation they were taken from. The line is red where our Machine Learning model was incorrect (false positives and negatives) and green where it was correct (true positives and negatives). The plasmoids identified by our separatrix-based algorithm are shown in blue. Bottom left: A close-up on one single slice, plotted in the same fashion as the top figure. Bottom right: The model input quantities along the slice shown left. Regions the model predicted were plasmoids are shown in yellow and regions that were separatrix-determined plasmoids are shown in blue. The model correctly classified plasmoid points in the gray crosshatched area. Coordinate note: These plots have been transformed from their simulation coordinates to GSM-like coordinates with z as the direction across the current sheet.

large numbers of "messy" plasmoids necessary for a thorough statistical study of plasmoid impacts on turbulent magnetotail reconnection. These sorts of broad studies are necessary to develop an understanding of the overall characteristics of complex magnetotail reconnection and its interplay with larger-scale magnetosphere processes like substorms, as well as multiscale reconnection in large systems in general (Ji et al., 2022).

It will be necessary to develop a more sophisticated and better-performing model than the simple 1D CNN shown here. Our simulations had scant reconnection and mostly showed gradual evolution to a less perturbed state, so we will consider adding unsteady and/or inhomogeneous driving to the simulation to encourage more dynamic evolution. We could also attempt to fit the simulated plasmoids to cylindrical flux rope models such as Lundquist (1950) or Elphic and Russell (1983) to quantify how much the "messy" plasmoids deviate from the ideal. We intend to modify the separatrix-based plasmoid identification algorithm to detect interiors of large plasmoids, since not doing so is clearly confusing our model. Hyperparameter optimization and postprocessing must be done to combat the data set's imbalance. It also may be preferable to avoid training a model from scratch and instead adapt an existing well-performing image classifier such as ResNet50 (He et al., 2015) which can be found pretrained online. In any case, after the model achieves good performance on simulated data it will likely still perform poorly on real data due to the many differences between our training data and real data, such as dimensionality, sample rates, and sources of noise. To combat this we will utilize domain adaptation, which is the process of modifying a model trained on one data set so that it performs well on another data set. There are numerous existing methodologies which have been developed to facilitate successful domain adaptation (Singhal et al., 2023). We are currently using adversarial discriminative domain adaptation (ADDA) (Tzeng et al., 2017), to adapt simulation-trained models to MMS data. This method uses adversarial training to extract the features from the target data set which are most similar to those identified in the source data set, therefore excluding features which differ between the two domains such as numerical noise. The shared features from the target data set are then used as input to the classifier model. We expect that our results could be further improved by reducing the differences between the simulated and real data, using methods similar to those employed by for example, Tremblay et al. (2018), but we have no plans of doing so at this time. We plan to evaluate the finished model's performance on a database of known magnetotail plasmoids such as for example, Ieda et al. (1998). It would also be beneficial to compare the model's performance to potential alternate models, such as ones which use training data that have synthetic instrument noise and ones trained on unperturbed simulations or mathematical models. By comparing multiple approaches we can determine which factors are most important for training a successful plasmoid detection model.

There are numerous potential applications of this process beyond the magnetotail plasmoid detection model shown here. A logical extension would be to apply this process to detect the plasmoids known as Flux Transfer Events (FTEs) which occur on Earth's dayside at the magnetopause; reconnection there is highly asymmetric so it would be inappropriate to simply use the model trained in this work. Magnetic structures are particularly hard to identify in turbulent plasmas, so this method could be used to identify structures in reconnection outflows or in complex environments such as the turbulent solar wind. For example, researchers looking for magnetic structures in Parker Solar Probe data should consider this technique.

Data Availability Statement

The code used for this work as well as supplemental materials can be found in the Zenodo repository (Bergstedt, 2024). PIC simulations were performed using the VPIC code V1.2 (Bowers, Albright, Bergen, et al., 2008; Bowers, Albright, Yin, et al., 2008; Bowers et al., 2009), available at https://github.com/lanl/vpic under a BSD-3 license. Solution of the Grad-Shafranov equation was done using the DOLFINX package version 0.5.1 (Alnaes et al., 2014, 2015; Logg & Wells, 2010; Scroggs et al., 2022), available at https://fenicsproject.org/ under a GNU Lesser General Public License. Image processing work was done using scikit-image version 0.17.2 (van der Walt et al., 2014), available at https://scikit-image.org/ under a BSD-3-Clause license. Some of the Machine Learning model construction and training was done using Tensorfow version 2.3.0 (Abadi et al., 2015), available at https://www.tensorflow.org/ under an Apache License 2.0. The final model was constructed using Pytorch version 1.12.1 (Paszke et al., 2019), available at https://pytorch.org/ under a BSD-3 license. Supplementary work on the model was done using scikit-learn version 0.24.1 (Pedregosa et al., 2011), available at https://scikit-learn.org/ under a BSD-3-Clause license. Figures were constructed using Matplotlib version 3.6.1 (Hunter, 2007), available at https://matplotlib.org/ under the Matplotlib license. File input/output was done using h5py version 2.10.0 (Collette, 2013), available at https://www.h5py.org/ under a BSD-3-Clause license; the HDF5 file format and related software (The HDF Group, 1997–2023) is available at https://github.com/HDFGroup/hdf5 under the HDF5 license. Array manipulation throughout the work was done using numpy version 1.19.2 (Harris et al., 2020), available at https://numpy.org/ under a BSD-3-Clause license. All python work was done using jupyterlab version 3.4.8 (Kluyver et al., 2016), available at https://jupyter.org/ under a BSD-3-Clause license.

Acknowledgments

Thanks to Drs. Jonathan Jara-Almonte and William Daughton for their ideas and feedback on this work. The simulations and modeling shown in this presentation were performed on computational resources managed and supported by Princeton Research Computing, a consortium of groups including the Princeton Institute for Computational Science and Engineering (PICSciE) and the Office of Information Technology's High Performance Computing Center and Visualization Laboratory at Princeton University. Machine learning work was done using tools from Pytorch, Tensorflow, scikit-learn, and scikit-image. This work was supported by the U.S. Department of Energy's Office of Fusion Energy Sciences under Contract No. DE-AC0209CH11466, by NASA under Grants NNH15AB29I and 80HQTR21T0105, and by the National Science Foundation Graduate Research Fellowship under Grant DGE-2039656. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Retrieved from https://www.tensorflow.org/

Alnaes, M. S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., et al. (2015). The FEniCS project version 1.5. *Archive of Numerical Software*, *3*. https://doi.org/10.11588/ans.2015.100.20553

Alnaes, M. S., Logg, A., Ølgaard, K. B., Rognes, M. E., & Wells, G. N. (2014). Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, *40*(2), 1–37. https://doi.org/10.1145/2566630

Argall, M. R., Small, C. R., Piatt, S., Breen, L., Petrik, M., Kokkonen, K., et al. (2020). MMS SITL ground loop: Automating the burst data selection process. *Frontiers in Astronomy and Space Sciences*, *7*. https://doi.org/10.3389/fspas.2020.00054

Arnol'd, V. I. (1994). Dynamical systems V. Bifurcation theory and catastrophe theory.

Banesh, D., Lo, L.-T., Kilian, P., Guo, F., & Hamann, B. (2020). Topological analysis of magnetic reconnection in kinetic plasma simulations. In *2020 IEEE visualization conference (VIS)* (pp. 6–10). https://doi.org/10.1109/VIS47514.2020.00008

Bergstedt, K. (2024). KBergst/plasmoids_ml: plasmoid-ml initial code+slice data+perturbed equilibria (version v2-alpha) [Software]. *Zenodo*. https://doi.org/10.5281/zenodo.10894358

Bergstedt, K., Ji, H., Jara-Almonte, J., Yoo, J., Ergun, R. E., & Chen, L.-J. (2020). Statistical properties of magnetic structures and energy dissipation during turbulent reconnection in the Earth's magnetotail. *Geophysical Research Letters*, *47*(19), e2020GL088540. https://doi.org/10.1029/2020GL088540

Birn, J., Sommer, R., & Schindler, K. (1975). Open and closed magnetospheric tail configurations and their stability. *Astrophysics and Space Science*, *35*(2), 389–402. https://doi.org/10.1007/BF00637005

Bowers, K. J., Albright, B. J., Bergen, B., Yin, L., Barker, K. J., & Kerbyson, D. J. (2008). 0.374 pflop/s trillion-particle kinetic modeling of laser plasma interaction on roadrunner. In *Proceedings of the 2008 ACM/IEEE conference on supercomputing*. IEEE Press.

Bowers, K. J., Albright, B. J., Yin, L., Bergen, B., & Kwan, T. J. T. (2008). Ultrahigh performance three-dimensional electromagnetic relativistic kinetic plasma simulation. *Physics of Plasmas*, *15*(5), 055703. https://doi.org/10.1063/1.2840133

Bowers, K. J., Albright, B. J., Yin, L., Daughton, W., Roytershteyn, V., Bergen, B., & Kwan, T. J. T. (2009). Advances in petascale kinetic plasma simulation with VPIC and roadrunner. *Journal of Physics: Conference Series*, *180*(1), 012055. https://doi.org/10.1088/1742-6596/180/1/012055

Breuillard, H., Dupuis, R., Retino, A., Le Contel, O., Amaya, J., & Lapenta, G. (2020). Automatic classification of plasma regions in near-Earth space with supervised machine learning: Application to magnetospheric multi scale 2016–2019 observations. *Frontiers in Astronomy and Space Sciences*, *7*. https://doi.org/10.3389/fspas.2020.00055

Chen, L. J., Bhattacharjee, A., Puhl-Quinn, P. A., Yang, H., Bessho, N., Imada, S., et al. (2008). Observation of energetic electrons within magnetic islands. *Nature Physics*, *4*(1), 19–23. https://doi.org/10.1038/nphys777

Chen, L.-J., Daughton, W., Bhattacharjee, A., Torbert, R. B., Roytershteyn, V., & Bessho, N. (2012). In-plane electric fields in magnetic islands during collisionless magnetic reconnection. *Physics of Plasmas*, *19*(11), 112902. https://doi.org/10.1063/1.4767645

Collette, A. (2013). *Python and hdf5*. O'Reilly.

dos Santos, L. F. G., Narock, A., Nieves-Chinchilla, T., Nuñez, M., & Kirk, M. (2020). Identifying flux rope signatures using a deep neural network. *Solar Physics*, *295*(10), 131. https://doi.org/10.1007/s11207-020-01697-x

Elphic, R. C., & Russell, C. T. (1983). Magnetic flux ropes in the Venus ionosphere: Observations and models. *Journal of Geophysical Research*, *88*(A1), 58–72. https://doi.org/10.1029/JA088iA01p00058

Fu, H. S., Vaivads, A., Khotyaintsev, Y. V., Olshevsky, V., André, M., Cao, J. B., et al. (2015). How to find magnetic nulls and reconstruct field topology with MMS data? *Journal of Geophysical Research: Space Physics*, *120*(5), 3758–3782. https://doi.org/10.1002/2015JA021082

Garton, T. M., Jackman, C. M., Smith, A. W., Yeakel, K. L., Maloney, S. A., & Vandegriff, J. (2021). Machine learning applications to Kronian magnetospheric reconnection classification. *Frontiers in Astronomy and Space Sciences*, *7*. https://doi.org/10.3389/fspas.2020.600031

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition.

Huang, S., Zhao, P., He, J., Yuan, Z., Zhou, M., Fu, H., et al. (2018). A new method to identify flux ropes in space plasmas. *Annales Geophysicae*, *36*(5), 1275–1283. https://doi.org/10.5194/angeo-36-1275-2018

Hubbard, J. H., & Hubbard, B. B. (2009). *Vector calculus, linear algebra, and differential forms: A unified approach* (4th ed., pp. 346–349). Matrix Editions.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Ieda, A., Machida, S., Mukai, T., Saito, Y., Yamamoto, T., Nishida, A., et al. (1998). Statistical analysis of the plasmoid evolution with geotail observations. *Journal of Geophysical Research*, *103*(A3), 4453–4465. https://doi.org/10.1029/97JA03240

Innocenti, M. E., Amaya, J., Raeder, J., Dupuis, R., Ferdousi, B., & Lapenta, G. (2021). Unsupervised classification of simulated magnetospheric regions. *Annales Geophysicae*, *39*(5), 861–881. https://doi.org/10.5194/angeo-39-861-2021

Ji, H., Daughton, W., Jara-Almonte, J., Le, A., Stanier, A., & Yoo, J. (2022). Magnetic reconnection in the era of exascale computing and multiscale experiments. *Nature Reviews Physics*, *4*(4), 263–282. https://doi.org/10.1038/s42254-021-00419-x

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, *151*, 107398. https://doi.org/10.1016/j.ymssp.2020.107398

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., et al. (2016). Jupyter notebooks—A publishing format for reproducible computational workflows. In F. Loizides, & B. Scmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). IOS Press. Retrieved from https://eprints.soton.ac.uk/403913/

Lenouvel, Q., Génot, V., Garnier, P., Toledo-Redondo, S., Lavraud, B., Aunai, N., et al. (2021). Identification of electron diffusion regions with a machine learning approach on mms data at the Earth's magnetopause. *Earth and Space Science*, *8*(5), e2020EA001530. https://doi.org/10.1029/2020EA001530

Liu, Y.-H., Birn, J., Daughton, W., Hesse, M., & Schindler, K. (2014). Onset of reconnection in the near magnetotail: PIC simulations. *Journal of Geophysical Research: Space Physics*, *119*(12), 9773–9789. https://doi.org/10.1002/2014JA020492

Logg, A., & Wells, G. N. (2010). DOLFIN: Automated finite element computing. *ACM Transactions on Mathematical Software*, *37*(2), 1–28. https://doi.org/10.1145/1731022.1731030

Lundquist. (1950). Magneto-hydrostatic fields. *Arkiv för fysik*, *2*, 361–365.

L. Medsker, & L. C. Jain (Eds.) (1999). *Recurrent neural networks: Design and applications* (1st ed.). CRC Press. https://doi.org/10.1201/9781003040620

Olshevsky, V., Khotyaintsev, Y. V., Lalti, A., Divin, A., Delzanno, G. L., Anderzén, S., et al. (2021). Automated classification of plasma regions using 3D particle energy distributions. *Journal of Geophysical Research: Space Physics*, *126*(10), e2021JA029620. https://doi.org/10.1029/2021JA029620

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv*. https://doi.org/10.48550/ARXIV.1511.08458

Paschmann, G., & Daly, P. W. (1998). *Analysis methods for multi-spacecraft data*. ISSI Scientific Reports Series, 1 (Vol. 1). ISSI Scientific Reports Series SR-001, ESA/ISSI.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Petrukovich, A., Artemyev, A., & Nakamura, R. (2016). Magnetotail reconnection. In W. Gonzalez, & E. Parker (Eds.), *Magnetic reconnection: Concepts and applications* (pp. 277–313). Springer International Publishing. https://doi.org/10.1007/978-3-319-26432-5_7

Pollock, C., Moore, T., Jacques, A., Burch, J., Gliese, U., Saito, Y., et al. (2016). Fast plasma investigation for magnetospheric multiscale. *Space Science Reviews*, *199*(1–4), 331–406. https://doi.org/10.1007/s11214-016-0245-4

Schindler, K. (1972). A self-consistent theory of the tail of the magnetosphere. In B. M. McCormac (Ed.), *Earth's magnetospheric processes* (pp. 200–209). Springer.

Scroggs, M. W., Baratta, I. A., Richardson, C. N., & Wells, G. N. (2022). Basix: A runtime finite element basis evaluation library. *Journal of Open Source Software*, *7*(73), 3982. https://doi.org/10.21105/joss.03982

Sharma, N., Jain, V., & Mishra, A. (2018). An analysis of convolutional neural networks for image classification. *Procedia Computer Science*, *132*, 377–384. https://doi.org/10.1016/j.procs.2018.05.198

Shi, Q. Q., Shen, C., Dunlop, M. W., Pu, Z. Y., Zong, Q.-G., Liu, Z. X., et al. (2006). Motion of observed structures calculated from multi-point magnetic field measurements: Application to cluster. *Geophysical Research Letters*, *33*(8), L08109. https://doi.org/10.1029/2005GL025073

Singhal, P., Walambe, R., Ramanna, S., & Kotecha, K. (2023). Domain adaptation: Challenges, methods, datasets, and applications. *IEEE Access*, *11*, 6973–7020. https://doi.org/10.1109/ACCESS.2023.3237025

Smith, A. W., Slavin, J. A., Jackman, C. M., Fear, R. C., Poh, G.-K., DiBraccio, G. A., et al. (2017). Automated force-free flux rope identification. *Journal of Geophysical Research: Space Physics*, *122*(1), 780–791. https://doi.org/10.1002/2016JA022994

Sonnerup, B. U. Ö., & Cahill, L. J., Jr. (1967). Magnetopause structure and attitude from explorer 12 observations. *Journal of Geophysical Research*, *72*(1), 171–183. https://doi.org/10.1029/JZ072i001p00171

Sonnerup, B. U. Ö., Hasegawa, H., Teh, W.-L., & Hau, L.-N. (2006). Grad-shafranov reconstruction: An overview. *Journal of Geophysical Research*, *111*(A9), A09204. https://doi.org/10.1029/2006JA011717

The HDF Group. (1997–2023). Hierarchical data format, version 5. Retrieved from https://www.hdfgroup.org/HDF5/

Torbert, R. B., Dors, I., Argall, M. R., Genestreti, K. J., Burch, J. L., Farrugia, C. J., et al. (2020). A new method of 3-D magnetic field reconstruction. *Geophysical Research Letters*, *47*(3), e2019GL085542. https://doi.org/10.1029/2019GL085542

Tremblay, B., Roudier, T., Rieutord, M., & Vincent, A. (2018). Reconstruction of horizontal plasma motions at the photosphere from intensitygrams: A comparison between DeepVel, LCT, FLCT, and CST. *Solar Physics*, *293*(4), 57. https://doi.org/10.1007/s11207-018-1276-7

Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. *arXiv e-prints, arXiv:1702.05464*. https://doi.org/10.48550/arXiv.1702.05464

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., et al. (2014). scikit-image: Image processing in Python. *PeerJ*, *2*, e453. https://doi.org/10.7717/peerj.453

Wang, Z. (1999). Efficient implementation of the exact numerical far field boundary condition for Poisson equation on an infinite domain. *Journal of Computational Physics*, *153*(2), 666–670. https://doi.org/10.1006/jcph.1999.6289