

Unsupervised classification of fully kinetic simulations of plasmoid instability using self-organizing maps (SOMs)

Sophia Köhne ¹, Elisabetta Boella ^{2,3} and Maria Elena Innocenti ^{1,†}

¹Institut für Theoretische Physik, Ruhr-Universität Bochum, Universitätsstraße 150, 44801 Bochum, Germany

²Physics Department, Lancaster University, Bailrigg, Lancaster LA11NN, UK

³Cockcroft Institute, Sci-Tech Daresbury, Warrington WA44AD, UK

(Received 12 February 2023; revised 2 May 2023; accepted 3 May 2023)

The growing amount of data produced by simulations and observations of space physics processes encourages the use of methods rooted in machine learning for data analysis and physical discovery. We apply a clustering method based on self-organizing maps to fully kinetic simulations of plasmoid instability, with the aim of assessing their suitability as a reliable analysis tool for both simulated and observed data. We obtain clusters that map well, *a posteriori*, to our knowledge of the process; the clusters clearly identify the inflow region, the inner plasmoid region, the separatrixes and regions associated with plasmoid merging. Self-organizing map-specific analysis tools, such as feature maps and the unified distance matrix, provide us with valuable insights into both the physics at work and specific spatial regions of interest. The method appears as a promising option for the analysis of data, both from simulations and from observations, and could also potentially be used to trigger the switch to different simulation models or resolution in coupled codes for space simulations.

Key words: plasma nonlinear phenomena, space plasma physics, plasma simulation

1. Introduction

In recent years, space physics research based on observations and numerical experiments has converged under several points of view. First, both spacecraft and numerical simulations have nowadays easy access to kinetic-scale processes. Recent missions, such as the Magnetospheric MultiScale (MMS) (Burch *et al.* 2016) spacecraft, Parker Solar Probe PSP (Fox *et al.* 2016) and Solar Orbiter (Müller *et al.* 2020), routinely sample ion and electron scales. Similarly, fully kinetic codes are employed for the study of ion- and electron-scale processes embedded in increasingly large spatial and temporal domains. This is the case in particular for semi-implicit particle-in-cell (PIC) (Hockney & Eastwood 2021), codes, e.g. Markidis *et al.* (2010), Innocenti *et al.* (2017b) and Lapenta,

† Email address for correspondence: mariaelena.innocenti@rub.de

Gonzalez-Herrero & Boella (2017), where advanced numerical techniques may also be used to enable the simulation of large domains, either within the fully kinetic description, e.g. Innocenti *et al.* (2013, 2015b) and Innocenti, Tenerani & Velli (2019b), or by coupling the kinetic and fluid descriptions, e.g. Daldorff *et al.* (2014), Ashour-Abdalla *et al.* (2015) and Lautenbach & Grauer (2018). Second, both spacecraft observations and numerical simulations produce an increasing amount of data. As an example, MMS collects a combined volume of ~ 100 gigabits per day of particle and field data, of which only a fraction can be transmitted to the ground due to downlink limitations (Baker *et al.* 2016). At the same time, a single PIC simulation can produce \sim tens of terabytes of output, depending on the frequency at which field and particle information is saved. Such a huge amount of data leads naturally to the usage of machine learning (ML, Bishop 2006; Goodfellow, Bengio & Courville 2016)-based methods for data analysis and investigation. A review of recent applications of ML techniques to the space physics domain is found in Camporeale (2019). Particularly useful in this context are classification or clustering techniques that can be used for the identification of ‘similar’ regions, either sampled or simulated, and hence for the detection of boundary crossing. Recently, supervised ML techniques have been used for the classification of large-scale magnetospheric regions and detection of boundary crossing, chiefly bow-shock and magnetopause crossings, e.g. in Argall *et al.* (2020), Breuillard *et al.* (2020), da Silva *et al.* (2020), Olshevsky *et al.* (2021), Lalti *et al.* (2022) and Nguyen *et al.* (2022). ML techniques, either supervised, e.g. Camporeale, Carè & Borovsky (2017) and Li *et al.* (2020), or unsupervised, e.g. Amaya *et al.* (2020) and Roberts *et al.* (2020), have been used for the classification/clustering of solar wind states. Unsupervised techniques, namely Gaussian mixture models, have recently proven quite effective in PIC simulations, either for the identification of regions of interest (Dupuis *et al.* 2020), where the particle distribution functions deviate from Maxwellian, or to encode particle information for later resampling during simulation restarts (Chen, Chacón & Nguyen 2021). In this paper, we will use an unsupervised clustering technique based on self-organizing maps (SOMs, Kohonen 1982) to cluster simulated data points obtained from a PIC simulation. The aim is to verify whether this procedure can be used for two purposes, namely simulation pre-processing and scientific investigation. The same procedure (with minimal variations) has already given satisfactory results on data of rather different origin, obtained from both observations and simulations. In Amaya *et al.* (2020), it has been applied to 14 years of Advanced Composition Explorer (ACE, Stone *et al.* 1998) solar wind measurements. In Innocenti *et al.* (2021), it has been used to cluster simulated data, and specifically data from a global magnetospheric simulation. The code used there was the magneto-hydro-dynamic (MHD)-based code OpenGGCM-CTIM-RCM (Raeder 2003), which targets large-scale processes originating from the interaction of the solar wind with the magnetosphere–ionosphere–thermosphere system. Quite satisfactorily, the clustering procedure used there was able to cluster simulated points into regions associated, *a posteriori*, with the pristine solar wind, the magnetosheath (divided into three clusters, just downstream and further away from the bow shock), the lobes, the inner magnetosphere and boundary layers. Verification and validation activities were conducted to ascertain the dependence of the obtained clusters on several hyper-parameters, including the features used for the clustering, the number of *k*-means clusters used for the classification of the trained weights, and SOM-related hyper-parameters, such as the number of nodes in the SOM, the learning rate and the initial lattice neighbourhood width (see § 2). Robustness of the proposed clustering to temporal variation was also investigated.

One rather fundamental question left open in Innocenti *et al.* (2021) was whether a similar clustering procedure would produce equally meaningful results when applied to

smaller-scale processes, kinetic in nature. The MHD simulations intend to reproduce plasma behaviour at scales large enough that certain assumptions can be considered satisfied. Among these assumptions, we list quasi-neutrality, the presence of thermal (Maxwellian) and isotropic velocity distribution functions, the possibility of ignoring the non-ideal terms in Ohm's law and finite gyroradius effects (Ledvina, Ma & Kallio 2008). These assumptions are obviously not respected in most heliospheric plasmas, especially at the very small and fast scales sampled by recent magnetospheric and solar wind missions, such as the above-mentioned MMS, PSP and Solar Orbiter. Hence, the need arises to verify if the clustering procedure described in Innocenti *et al.* (2021) is robust to using simulation approaches, such as PIC methods, that deliver results directly comparable to observations (Innocenti *et al.* 2016). This work intends to address this question. The aim of this work is twofold: on one hand, we intend to verify if this procedure is useful in the post-processing of large-throughput numerical simulations. On the other hand, this analysis constitutes a necessary first step to validate the method before applying it to spacecraft observations. We apply the clustering procedure from Innocenti *et al.* (2021) to a fully kinetic PIC simulation of the plasmoid instability. The plasmoid instability is a fast instability that breaks down current sheets into multiple magnetic islands, which later undergo nonlinear evolution. We refer the reader to Loureiro & Uzdensky (2015) and Pucci *et al.* (2020) for a review of recent developments in plasmoid instability research both in the collisional and collisionless regimes. The plasmoid instability results in fast, spontaneous magnetic reconnection in plasmas and, as such, is deeply connected to the fundamental topic of particle heating and acceleration in space and astrophysical plasmas. A number of recent PIC simulations have focused specifically on the role of magnetic reconnection triggered by plasmoid instability in electron heating and acceleration. Processes observed in plasmoid instability simulations that result in particle heating and acceleration are acceleration by the reconnection electric field (Li *et al.* 2017), Fermi acceleration (Guo, Jokipii & Kota 2010) and plasmoid merging (Drake *et al.* 2006; Petropoulou *et al.* 2018). The efficiency of these processes has been observed to vary according to the plasma beta (Li *et al.* 2015) and magnetization (Guo *et al.* 2016). In simulations of plasmoid instability evolution, different regions are immediately distinguishable with the naked eye: an inflow region, separatrices, the plasmoid themselves and the plasmoid merging regions. We aim at understanding if our unsupervised clustering method is capable of comparable region identification.

This paper is organized as follows: in § 2 we describe SOMs and our clustering procedure, in § 3 the simulation used. In § 4 we describe our results: preliminary data inspection, scaling experiments, SOM training and the k -means clustering of trained SOM nodes (§ 4). An *a posteriori* analysis of clustering results and physical insights obtained from them are described in § 5. Discussions and conclusions follow. In the [Appendix](#) we comment on the robustness to hyper-parameter choice of our clustering procedure.

2. Self-organizing maps: a summary

Self-organizing maps (Kohonen 1982; Villmann & Claussen 2006) can be viewed as both a clustering and a dimensionality reduction procedure (Kohonen 2014). The aim is to represent a large set of possibly high-dimensional data as a (usually) two-dimensional (2-D) ordered lattice composed of $y \times x = q$ nodes/units/neurons, with y and x respectively the number of rows and columns, and q the total number of nodes. Each node i is characterized by a position in the 2-D lattice and by a weight $w_i \in \mathbb{R}^n$, where n is the number of features associated with each data point, and hence with each weight. Before the training, the weights are initialized randomly so as to span the parameter space covered by the first two principal components (Shlens 2014) of the input data, to make

training faster (Kohonen 2014). At the end of the training, the node weights will have been modified so that the nodes (*a*) represent local averages of the input data that maps to them, and (*b*) are topographically ordered according to their similarity relation. Nearby nodes should be more ‘similar’ to each other than far away nodes. To define similarity a distance metric is needed. Common choices are the Euclidean, sum of squares, Manhattan and Tanimoto distances (Xu & Tian 2015).

The training is unsupervised, meaning that training data are not labelled. Each data point \mathbf{x}_τ is presented to the map multiple times. On each occasion the following procedure, a mix of competition and collaboration, is repeated:

- (i) competition: the best matching unit (BMU) of the input data \mathbf{x}_τ is identified, by selecting the node whose weight \mathbf{w}_s , among the set \mathcal{W} of all the weights in the map, has minimum distance ($\|\cdot\|$) with respect to \mathbf{x}_τ

$$\mathbf{w}_s = \arg \min_{\mathbf{w}_i \in \mathcal{W}} (\|\mathbf{x}_\tau - \mathbf{w}_i\|); \quad (2.1)$$

- (ii) collaboration: to obtain an ordered map, not only the BMU but also some neighbouring nodes are tagged for update at each iteration. Such neighbours are selected through a lattice neighbourhood function $h_\sigma(\tau, i, s)$

$$h_\sigma(\tau, i, s) = \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_s\|^2}{2\sigma(\tau)^2}\right), \quad (2.2)$$

where s is the BMU index and $\mathbf{p}_i \in \mathbb{R}^2$ the position of node i in the 2-D map; $\sigma(\tau)$ is the iteration(τ)-dependent lattice neighbourhood width that determines the extent around the BMU of the update introduced by the new input;

- (iii) weight update: the weights of the selected nodes are updated, to make them become more similar to the input data. The magnitude of the correction $\Delta \mathbf{w}_i$ for the node i is calculated as follows:

$$\Delta \mathbf{w}_i = \epsilon(\tau) h_\sigma(\tau, i, s) (\mathbf{x}_\tau - \mathbf{w}_i). \quad (2.3)$$

The correction depends on the distance of the nodes from the BMU and on the learning rate $\epsilon(\tau)$, which is also iteration dependent.

Both the lattice neighbourhood width and the learning rate decrease with increasing iteration number, according to predetermined rules. The initial lattice neighbour width and learning rate, $\sigma(\tau = 0)$ and $\eta(\tau = 0)$, are labelled σ_0 and η_0 , respectively. The rationale for this iteration-dependent decrease is the following: at the beginning of the training far-reaching, large-magnitude updates of the node weights are needed, since the node weights are very different from the input data and the topology of the data has to be enforced on the map. After several iterations, when the map already resembles the data, the node updates become targeted in position and smaller in magnitude. The overall convergence of the map is, therefore, separable into two phases: first the topographic ordering of the nodes and then the convergence of the weight values according to the quantization error (Van Hulle 2012).

The quantization error, Q_E , defined as

$$Q_E = \frac{1}{m} \sum_{i=1}^m \|\mathbf{x}_i - \mathbf{w}_{s|\mathbf{x}_i}\|, \quad (2.4)$$

measures the average distance between each of the m entry data points \mathbf{x}_i and their BMU, $\mathbf{w}_{s|\mathbf{x}_i}$. As such, it can be used to measure how closely the map reflects the training data

distribution. The quantization error is expected to decrease and finally plateau during map training.

The SOMs are at the core of the clustering procedure we use in this work, which has already been used (with minimal variation) and described in Amaya *et al.* (2020) and Innocenti *et al.* (2021). We briefly describe the method here for the reader's convenience:

- (i) preliminary data inspection and data pre-processing: we describe in § 4 our experiments with different data scalers;
- (ii) SOM training, as described above;
- (iii) k -means clustering (Lloyd 1982) of the trained SOM nodes. After SOM training, the data points are clustered in q clusters, with q the number of SOM nodes; q is typically too high for meaningful results inspections. Hence, we further cluster the trained SOM nodes into a lower number of k -means clusters. The number of clusters is determined through an unsupervised procedure, the Kneedle cluster number determination (Satopaa *et al.* 2011);
- (iv) clustering of the simulated data points, based on the k -means cluster of their BMU. At this stage, we can inspect our clustering results and use the result of the clustering to obtain physical insights on our data, see §§ 4 and 5.

In Amaya *et al.* (2020) and Innocenti *et al.* (2021) the serial SOM implementation from Vettigli (2018) was used. Here, due to the large volume of data points to cluster, we move to the parallel SOM implementation using CUDA (NVIDIA, Vingelmann & Fitzek 2020) and C++ by Mistri (2018). The algorithm is the same as in the serial implementation, but the distance calculations between the data samples and the neuron weights are parallelized. The parallel implementation allows us to choose, for the training phase, between online and batch learning. In the online version of the algorithm, the data samples are each processed according to the scheme described above. In this case, the updates of the weights are determined by the impact of the individual samples. The batch algorithm, on the other hand, finds the BMU for each sample and then sums up all data samples mapping to one node at the beginning of a training cycle to form node sums. Then, during training, a neighbourhood set for each of the nodes is found and all the nodes sums are summed up to a neighbourhood sum, which is divided by the total number of samples contributing, to form a neighbourhood mean. In the end, the weights are updated in one operation over all nodes of the SOM, where the values of the weights are replaced by those neighbourhood means (Kohonen 2014). The batch algorithm does not need a learning rate. We chose here the online training algorithm. In table 1 we compare the average run time per epoch, in seconds, of the serial and parallel SOM implementations, as a function of the number of samples m . One epoch corresponds to presenting all the samples to the map once. In the last column we record the ratio of the execution times of the parallel and serial implementations as a function of the number of samples. We observe that using the parallel implementation becomes increasingly convenient with increasing sample number.

3. Simulation description

Self-organizing maps are used to cluster results of a 2-D kinetic simulation of plasmoid instability. The simulation was carried out with the semi-implicit PIC code ECsim (Lapenta 2017; Lapenta *et al.* 2017; Gonzalez-Herrero, Boella & Lapenta 2018). In the PIC algorithm, a statistical approach is adopted and plasmas are described using computational particles or macroparticles representative of several real plasma particles. These computational particles interact via the electromagnetic fields that they themselves

Implementation	m	average runtime/epoch in s	speedup factor
serial	2264	0.785	
parallel		0.314	2.4
serial	5660	2.476	
parallel		0.798	3.1
serial	11 320	2.986	
parallel		1.603	3.7
serial	22 641	14.625	
parallel		3.240	4.5
serial	226 419	356.094	
parallel		36.302	9.8
serial	1 132 096	3597.720	
parallel		176.865	20.3

TABLE 1. Comparison of the execution times of the parallel CUDA/C++ implementation CUDA-SOM (Mistri 2018) and the serial Python implementation miniSom (Vettigli 2018), both trained for 5 epochs with different sample sizes m . The data points are extracted from the upper current sheet of the simulation described in § 3. The SOM parameters used for the training are initial learning rate $\eta_0 = 0.5$ and initial neighbourhood radius $\sigma_0 = 0.2 \times \max(x, y)$.

produce. The fields are computed on a fixed grid by solving Maxwell's equations. ECsim implements a semi-implicit algorithm, which makes the code stable and accurate over different spatial and temporal resolutions. As a consequence, the code resolution can be tuned to the physics of interest, rather than to the smallest scales of the problem (Micera *et al.* 2020). In the simulation, two oppositely directed force-free current sheets are used as initial condition. The sheets sustain a magnetic field $\mathbf{B} = B_x(y)\hat{\mathbf{x}} + B_z(y)\hat{\mathbf{z}}$ with

$$B_x(y) = B_{0,x} \left[-1 + \tanh\left(\frac{y - 0.25L_y}{\delta}\right) + \tanh\left(-\frac{y - 0.75L_y}{\delta}\right) \right], \quad (3.1)$$

and

$$B_z(y) = B_{0,x} \sqrt{\operatorname{sech}^2\left(\frac{y - 0.25L_y}{\delta}\right) + \operatorname{sech}^2\left(\frac{y - 0.75L_y}{\delta}\right)} + B_g^2. \quad (3.2)$$

Here, L_y is the transverse size of the simulation box, $\delta = d_i$ is the current sheet half-thickness, $d_i = c/\omega_{pi}$ is the ion inertial length, $\omega_{pi} = \sqrt{4\pi e^2 n_{0i}/m_i}$ the plasma frequency for ions of density n_{0i} and mass m_i , e is the elementary charge and B_g the magnitude of the guide field. While the upper current sheet is unperturbed and the tearing instability is seeded from numerical noise, the magnetic field of the lower current sheet was perturbed with a long wave perturbation (Birn *et al.* 2001) to trigger the instability on faster time scales. The simulation box is filled with a plasma composed of electrons and ions with mass ratio $m_i/m_e = 25$, where m_e is the electron mass, having uniform density $n_{0i} = n_{0e} = n_0$ and temperature $T_{0i} = T_{0e} = T_0$. Electrons are initialized with a drift velocity $\mathbf{v}_e = v_{e,x}(y)\hat{\mathbf{x}} + v_{e,z}(y)\hat{\mathbf{z}}$, such that $\nabla \times \mathbf{B} = 4\pi \mathbf{J}_e/c$ is satisfied, with \mathbf{J}_e the electron current density. In our simulation, we set $v_A = 0.2c$, $\omega_{pe}/\Omega_{ce} = 1$, $\beta_e = \beta_i = 0.02$ and $B_g = 0.03B_{0,x}$. Here, $v_A = B_{0,x}/(4\pi n_0 m_i)$ is the Alfvén velocity, $\omega_{pe} = \sqrt{4\pi e^2 n_0/m_e}$ is the electron plasma frequency, $\Omega_{ce} = eB_{0,x}/(m_e c)$ is the electron cyclotron frequency, $\beta_{e,i} = 8\pi n_0 T_0/B_{0,x}^2$ is the ratio between electron/ion pressure and magnetic pressure and c

is the speed of light in vacuum. These dimensionless parameters are similar to those used in Li *et al.* (2017) and correspond to plasma conditions typical of the solar corona and the accretion disk corona. We used a simulation box with longitudinal and transverse sizes $L_x = L_y = 200d_i$, discretized with 2128×2128 cells to resolve c/ω_{pe} twice. To model the plasma dynamics accurately, 64 particles per cell per species were employed. Particles were pushed for more than 7000 iterations with a temporal time step of $0.16\Omega_{ci}^{-1}$, with $\Omega_{c,i}$ ion cyclotron frequency. Periodic boundary conditions for fields and particles were adopted. We performed a detailed convergence study to ensure that the chosen numerical parameters do not affect the physics under investigation. Figure 1 displays the out-of-plane magnetic field component (B_z , *a*), one diagonal ($p_{xx,e}$, *b*) and one non-diagonal, ($p_{xz,e}$, *c*), electron pressure term, and the out-of-plane electron current ($J_{z,e}$, *d*) at $\Omega_{ci}t = 320$. Field lines are superimposed on each panel. At this time the tearing instability is in its nonlinear phase in both current sheets. In the upper current sheet, we observed the formation of small magnetic islands that grew with time and merged to produce the shown configuration. In the lower current sheet, where a perturbation was originally present, single X point reconnection developed initially. Later, plasmoids formed in the current sheet and plasmoid merging was also observed. Signatures associated with plasmoid merging are described e.g. in Cazzola *et al.* (2015, 2016). At the location of the X points ($x/d_i \simeq 70$ and 170 in the upper current sheet and $x/d_i \simeq 80$ in the lower current sheet), the out-of-plane magnetic field component B_z shows the typical quadrupolar structure associated with collisionless Hall reconnection. The xx component of the electron pressure tensor is low at the X points and high at the periphery of magnetic islands, as noted for the electron temperature in Lu *et al.* (2019). A similar behaviour is observed for the xz pressure tensor component, which is higher in absolute value at the border of magnetic islands with respect to other regions. It is interesting to notice that $p_{xz,e}$ changes polarity within a magnetic island. The role of the pressure tensor off-diagonal components in causing fast reconnection in collisionless plasmas has been studied in a number of previous works, e.g. Hesse & Winske (1998) and Ricci *et al.* (2004).

4. Clustering results

The data points we cluster are from the upper current sheet of the simulation described in § 3, at time $\Omega_{ci}t = 320$. As is standard practice in clustering procedures based on distance, the data are scaled to prevent the features with larger magnitude from dominating the clustering (Angelis & Stamelos 2000). Scaling the data is the transformation of the feature values according to a defined rule. This is necessary to assure that all features have the same (or rather a proportional) degree of influence in the evaluation (Huang, Li & Xie 2015). If the feature values range for one feature in the tens and for another in the thousands, a ML algorithm will always be influenced more by the feature ranging in the thousands, if scaling is not used. We tested three different scalars from the scikit-learn library (Pedregosa *et al.* 2011): the so-called MinMax, Standard and Robust scalars. The first scales each feature in the dataset to a fixed interval, here $[0, 1]$, the second to zero mean and unit variance. The third removes the median and scales the data according to a quantile range, here the interquartile range between the first and third quartiles. In figure 2 we depict the violin plots for the distributions of an outlier-poor feature, B_y , and an outlier-rich one, $J_{z,e}$, after scaling them with the three scalars; we can observe that the data range after scaling changes dramatically for the outlier-rich feature. Since $J_{z,e}$ is strongly associated with reconnection X-points, we can expect that the choice of scalar will have an effect on clustering results.

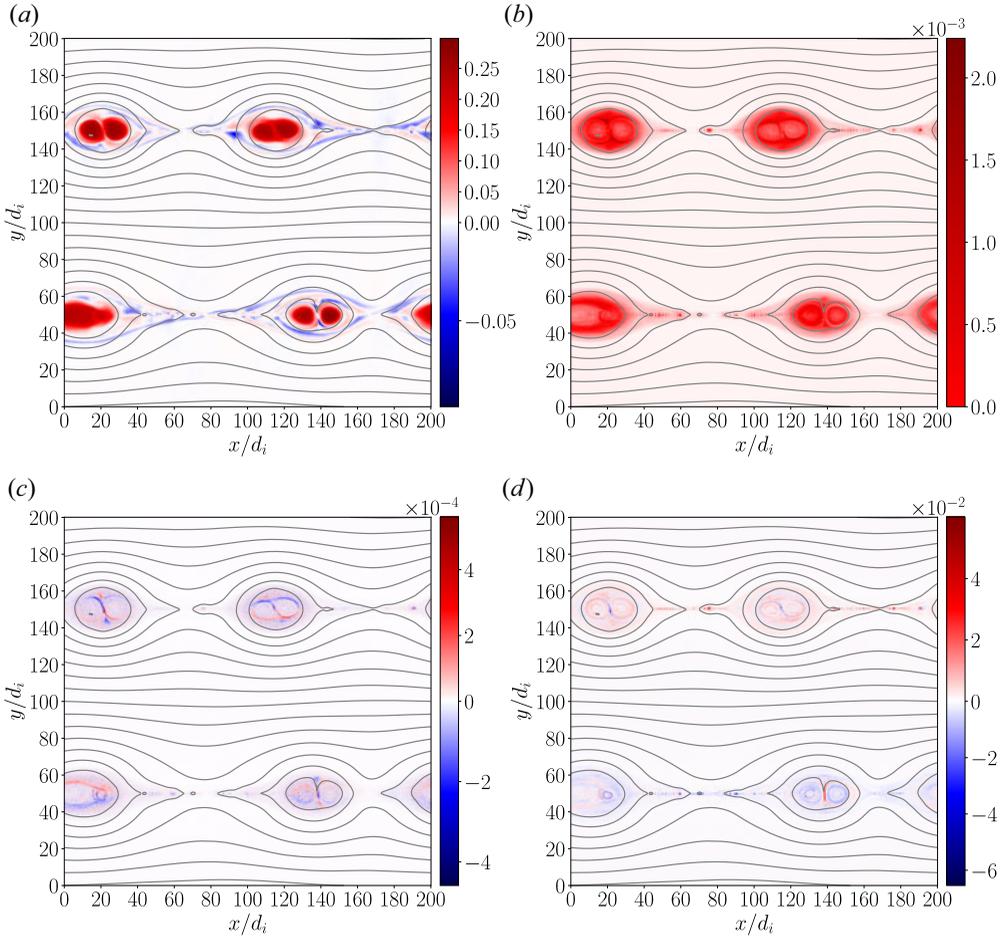


FIGURE 1. Out-of-plane magnetic field component (B_z , a), one diagonal ($p_{xx,e}$, b) and one non-diagonal, ($p_{xz,e}$, c), electron pressure term, and the out-of-plane electron current ($J_{z,e}$, d) at $\Omega_{ci}t = 320$. Magnetic field lines are superimposed in black. Electromagnetic fields, pressure components and currents are in units of $m_i c \omega_{pi} / e$, $n_0 m_i c^2$ and $en_0 c$, respectively.

For each set of scaled data, we proceed to train a SOM. The number q of nodes of each map has been determined using the following rule of thumb (Kohonen 2014):

$$q \approx 5 \times \sqrt{m}, \tag{4.1}$$

where q is approximated to an integer and m is the number of samples. The ratio of the side lengths x , y of the map is set to match the ratio of the largest two principal components (Shlens 2014) of the training data. We notice that, since the principal component analysis is performed on scaled data, the number of rows and columns in the map may differ in the different cases. The initial neighbourhood radius is then chosen to cover 20% of the larger side length of the map

$$\sigma_0 = 0.2 \times \max(x, y). \tag{4.2}$$

For the MinMax and the Standard scalers this resulted in $x = y = 82$ and $\sigma_0 = 16$ and for the Robust scaler in $x = 93$, $y = 71$ and $\sigma_0 = 19$.

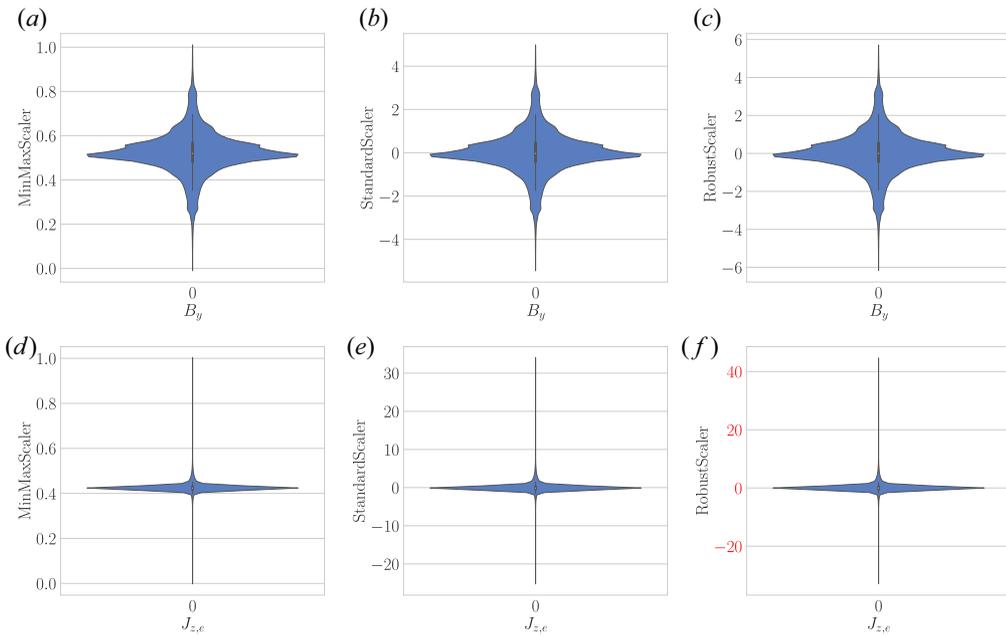


FIGURE 2. Violin plots of the distribution, after scaling, of an outlier-poor (B_y) and an outlier-rich (J_z, e) feature. The MinMax, Standard and Robust scalers are used.

The initial learning rate was kept constant at $\eta_0 = 0.5$ and the weights were initialized using random samples from the training data. The maps shown in figure 3 have all been trained for five epochs. We remark that, with this relatively low number of epochs, the map cannot be expected to have converged, as is often the case with SOMs. However, the results that we obtain are remarkably stable to all parameters, including the number of epochs, as shown in the Appendix. There, we study how varying the SOM hyper-parameters and the seed for the random initialization of the weights influences clustering results. In table 2 we report all hyper-parameters used for SOM training.

In figures 3 and 4 we show the clustering results obtained with the three scalers. The trained SOM nodes have been further clustered with k -means into larger-scale regions, as described in § 2. The optimal k -means cluster number, $k = 5$, has been selected with the Kneedle cluster number determination (Satopaa *et al.* 2011).

In figure 3, first column, we see the trained SOM nodes obtained with the three different scalers (MinMaxScaler, StandardScaler and RobustScaler, (a,c,e), respectively) and coloured according to the k -means cluster they are clustered into. In the second column (b,d,f), we depict the unified distance matrices (UDMs) associated with the three cases. In the UDM representation each neuron is coloured according to its normalized distance with respect to its nearest neighbour (Kohonen 2014); ‘darker’ neurons are less similar to their neighbour than ‘lighter’ neurons. We can observe a similar pattern in the three UDMs: a wide, light area composed of similar neurons is mapped to the same k -means cluster, the 0, blue cluster. A darker area is present in one of the map corners, not necessarily the same one in the three plots because, in SOMs, the information of interest is not the absolute position but the relative position with respect to the neighbouring nodes. This darker area maps to further four clusters. Even darker areas may be present within clusters or at the boundary between clusters, see e.g. the dark area in figure 3(f), at the intersection between cluster 3, red, 0, blue and 2, green.

Parameter	Value
learningmode	online 'o'
nRows x	82 (MinMax- and StandardScaler), 93 (RobustScaler)
nColumns y	82 (MinMax- and StandardScaler), 71 (RobustScaler)
initial learning rate η_0	0.5
final learning rate η_f	0
initial neighbourhood radius σ_0	$0.2 \times \max(x, y)$
distance function	Euclidean 'e'
neighbourhood function	Gaussian 'g'
initialization	random input vectors 'c'
lattice	hexagonal
toroidal	off
randomize	on
exponential decay of η and σ	both 'b'
normalizedistance	off

TABLE 2. The CUDA-SOM hyperparameters used in SOM training.

In [figure 4](#) we depict the simulated data points at time $\Omega_{cit} = 320$, see [figure 1](#) in [§ 3](#), coloured according to the k -means cluster they map to, for the three different scalers. We can now map the neurons from [figure 3](#) to the physical regions in [figure 4](#).

In all three cases, the blue cluster from [figure 3](#) maps to the plasma outside of the plasmoid region. We can call this region, slightly oversimplifying, the inflow region. The plasmoid region is clustered differently in the three cases; we have here a proof of the importance of pre-processing activities (in this case, the choice of scaler) in clustering results. With MinMaxScaler, [figure 4\(a\)](#), the outer plasmoid region neighbouring the inflow region is divided into two clusters, cluster 3, red and 4, purple. The inner plasmoid region, readily identifiable 'by eye' in the plots of [§ 3](#), is mapped to cluster 2, green. Intermediate plasmoid regions are mapped to cluster 1, orange. With StandardScaler, [\(b\) in figure 4](#), the outer plasmoid region is assigned to a single cluster, cluster 3, red. The inner plasmoid region is again mapped to cluster 2, green. The remaining plasmoid regions are clustered into two clusters, cluster 1, orange, and 4, purple. The blue and green clusters obtained with the RobustScaler are very similar to those obtained with MinMaxScaler and StandardScaler. 'Walking' in [figure 4\(c\)](#), from the inflow region towards the inner plasmoid region we encounter cluster 3, red, 1, orange and 4, purple. Incidentally, we notice that we can similarly walk from cluster 0 to 3, 1, 4 and finally 2 also in the map depicted in [figure 3\(e\)](#), a confirmation that neighbourhood in the SOM derives from feature similarity.

When confronted with different clustering results, we have to identify criteria that allow us to prefer one clustering method over another. This is usually quite a daunting task for unsupervised methods, where the 'ground truth' is not known. Luckily, here, as already in [Innocenti *et al.* \(2021\)](#), we are in a rather fortunate situation; we are clustering simulated data, hence we have complete information about all features as a function of space and time. Furthermore, we have previous knowledge of the process we are analysing. Hence, we can determine that the clustering obtained with the RobustScaler is the most useful for our purposes, because it separates regions where we can expect different physical processes to take place.

It is instructive to speculate on the reasons why regions that we may want to see clustered together, based on the physics that occur there, are assigned to different clusters.

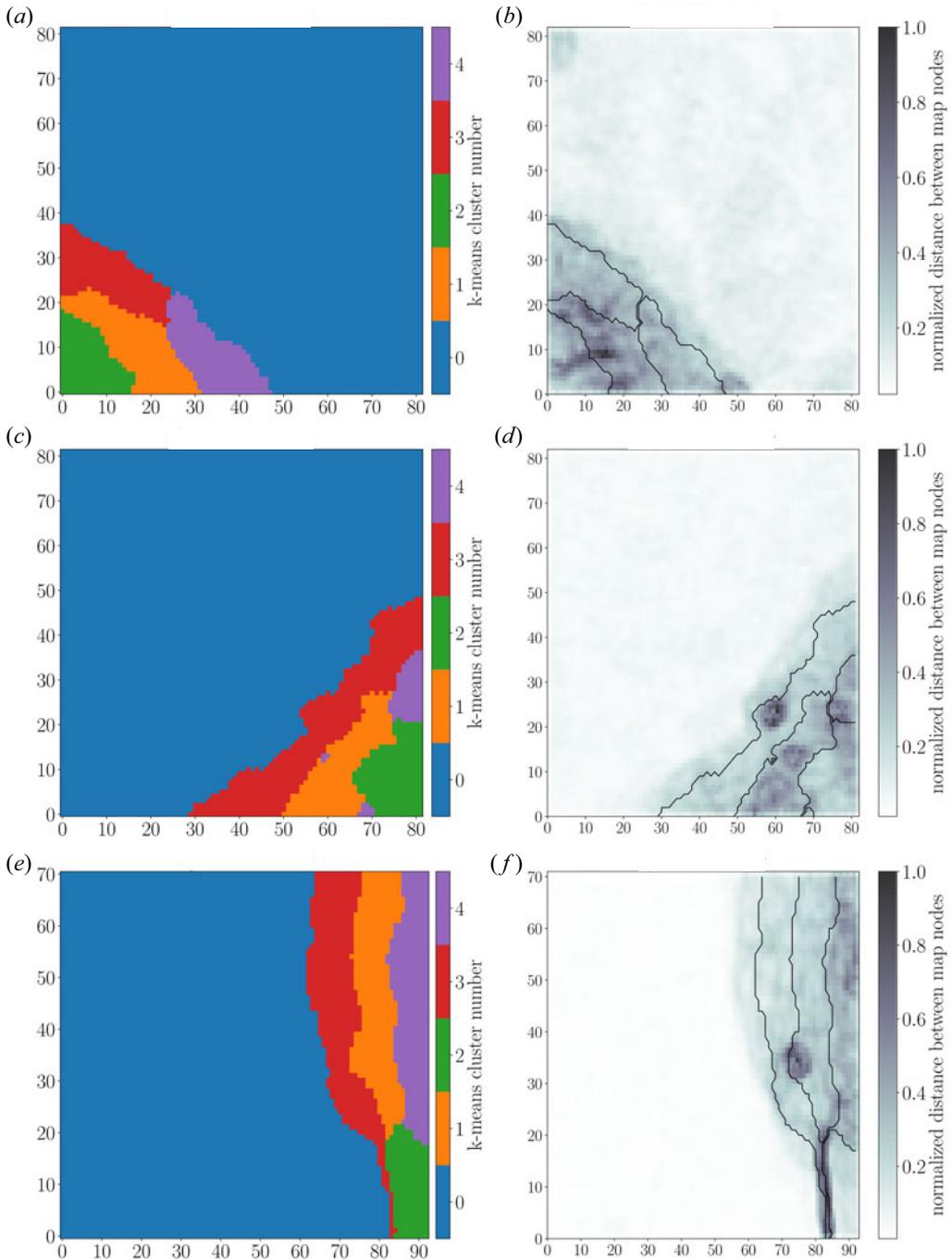


FIGURE 3. Trained SOM nodes coloured according to their k -means clusters (a,c,e) and UDM maps (b,d,f) for the data scaled with MinMaxScaler (a,b), StandardScaler (c,d), RobustScaler (e,f). Cluster boundaries are drawn in black in (b,d,f).

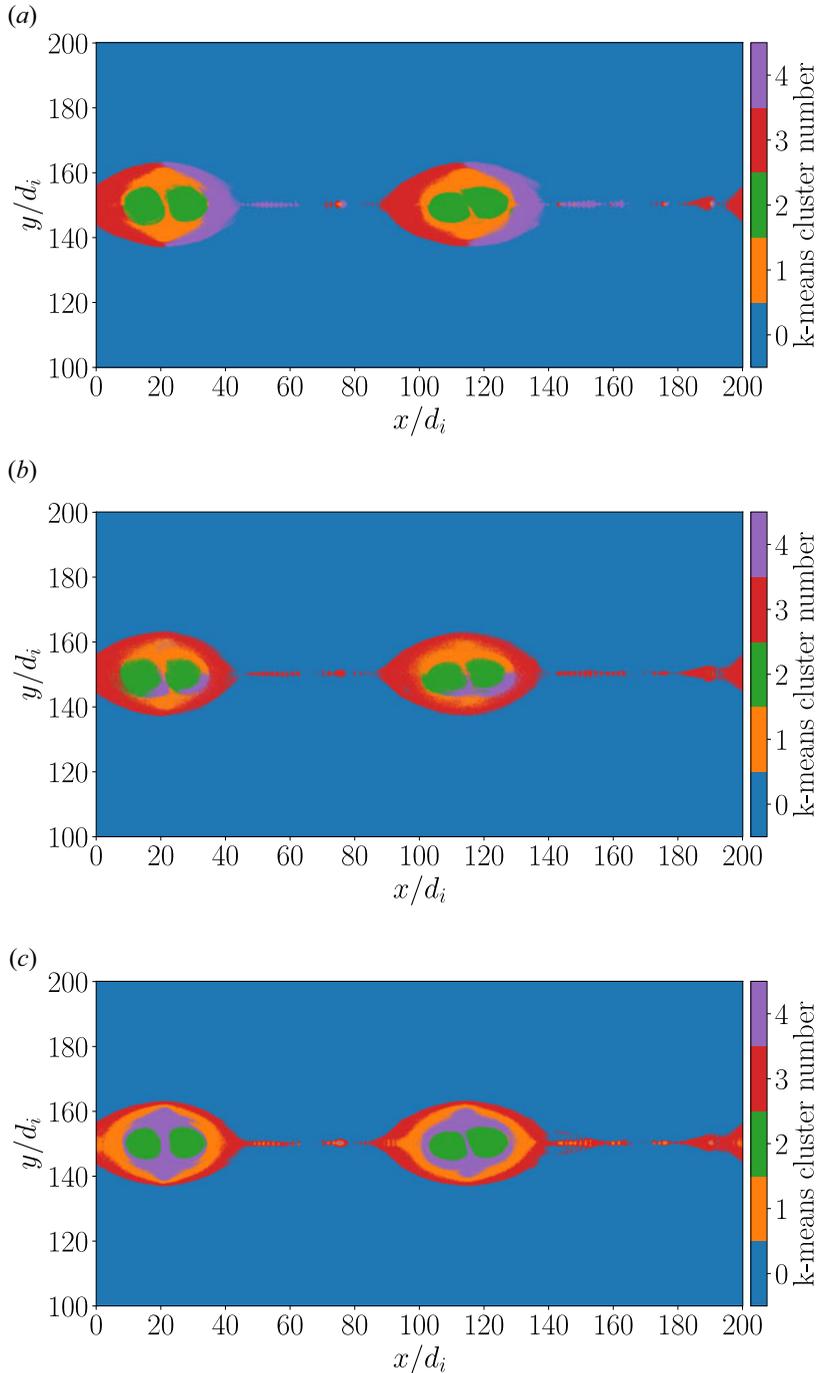


FIGURE 4. Upper current sheet simulated data at $\Omega_{cit} = 320$, coloured according to the k -means cluster the BMU of each point is clustered into. Data scaled with MinMaxScaler, StandardScaler and RobustScaler are depicted in (a), (b) and (c) respectively.

This is the case, for example, for cluster 3, red and 4, purple, in [figure 4\(a\)](#), depicting data scaled with MinMaxScaler. Analysing the feature values of the points in the two clusters, we realize that the main difference between the two sets of points is the sign of the y component of the magnetic field. Most other features, including features that we consider of particular relevance in plasmoid instability/magnetic reconnection simulations (non-diagonal pressure terms, out-of-plane electron current ...) present quite similar values in the two clusters. Looking at B_y and $J_{z,e}$ after scaling with MinMaxScaler in [figure 2\(a,d\)](#), we realize that the differences in the outlier-poor B_y for the two clusters are at the two extremes of the value range, here 0 and 1, while the similarities in the outlier-rich $J_{z,e}$ are compressed towards 0.5; the differences in the former ‘weigh’ more in the clustering than the similarities in the latter. With RobustScaler, instead, B_y spans $\sim[-6; 6]$, while $J_{z,e}$ varies in the range $\sim[-25; 42]$; the B_y values become significantly less relevant in the clustering than the $J_{z,e}$ values, and, according to the results of [figure 4](#), are not capable anymore of driving the formation of cluster 3 vs cluster 4.

We consider this quite a significant demonstration of the importance of accurate pre-processing before clustering activities; data should be inspected before and after scaling to assess if scaling has preserved important characteristics of the data we may want to rely upon during clustering (e.g. outliers). Furthermore, the results of clustering activities when using different scalers should be compared.

5. The *a posteriori* analysis of clustering results

In § 4 we identified the results obtained using the RobustScaler as the most useful for our purposes, since the points clustered together are the ones where we can expect similar processes to occur. This determination is made on the base of our previous knowledge of plasmoid instability evolution.

Now, we intend to examine these specific results in more detail.

First, we focus on the UDM depicted in [figure 3\(f\)](#). The area associated with the inflow region, cluster 0, blue in [figure 4\(c\)](#), is lighter in colour with respect to the other two cases, showing minimal differences between the nodes. We observe that a darker UDM region encloses a ‘walled in’ area in the lower right corner of the map associated with cluster 2, green. The nodes there are quite similar to each other, as seen from their light colour. They are, however, very different from the ones in the inflow region and also quite different from the other nodes mapping to plasmoids, as seen from the dark colour of the nodes at the boundary between clusters. Comparing [figures 3\(f\)](#) and [4\(c\)](#), we see that the ‘walled in’ region in the UDM maps to the inner plasmoid region. The features in this regions are, in fact, quite distinct with respect to the neighbouring points, as confirmed by violin plot analysis of the data clustered in each cluster and by [figure 9](#) below. Moving from cluster 2 in [figure 4](#) towards the inflow region, cluster 0, blue, we encounter cluster 4, purple, 1, orange, and finally cluster 3, red, at the boundary between the inflow region and the plasmoid. We can broadly identify cluster 3, red, as the ‘separatrix’ cluster.

In [figure 5](#) we depict the variation across the map of B_z , $p_{xx,e}$, $p_{xz,e}$, $J_{z,e}$. The boundaries between the k -means clusters are depicted as black lines. The feature values have been scaled back to the normalized values obtained as simulation results. Before going into a detailed analysis of [figure 5](#), we notice several sub-structures in the clusters depicted. This is especially the case for cluster 4, at the right edge of the map. It is therefore no surprise that cluster 4 breaks into 2 and 3 clusters mapping the larger-scale sub-structures with $k = 6$ and $k = 7$, respectively.

We compare [figure 5](#) with [figures 3](#) and [4](#) to highlight important characteristics of the different simulated regions. We observe that high-magnitude, positive values of B_z ([figure 5a](#)), are concentrated in the inner plasmoid region, cluster 2, green. Lower, positive

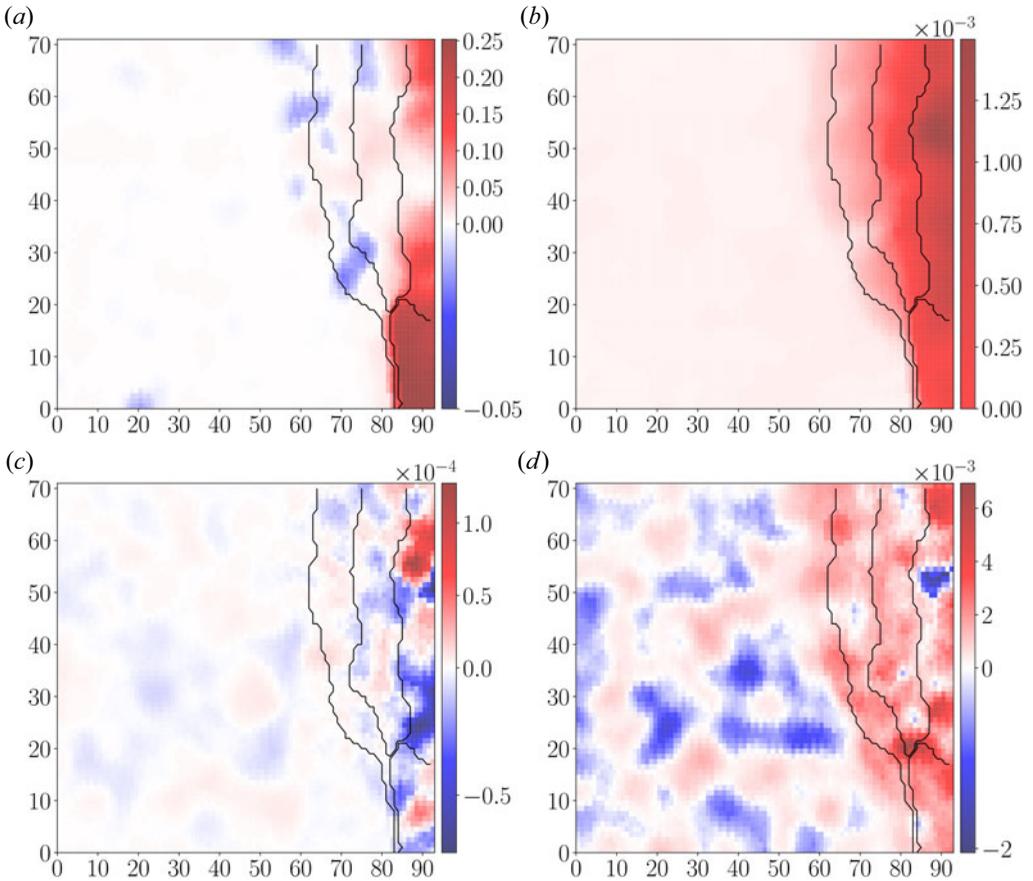


FIGURE 5. The B_z (a), $p_{xx,e}$ (b), $p_{xz,e}$ (c) and $J_{z,e}$ (d) values associated with the trained SOM nodes, for data points scaled with the RobustScaler. Cluster boundaries are drawn in black.

values are associated with cluster 4, purple. Most negative B_z values are found in clusters 1, orange and 3, red. In the latter, both positive and negative B_z values are found, most probably associated with the quadrupolar structure in the out-of-plane magnetic field associated with collisionless reconnection. In figure 5(b,c), we depict $p_{xx,e}$ and $p_{xz,e}$. We notice in (b) low $p_{xx,e}$ values in the inflow region, and higher pressure values in correspondence with the plasmoids. We further observe that $p_{xx,e}$ exhibits rather high values in the upper right corner of the map, associated with negative values of $p_{xz,e}$ (c) and rather high positive value of the out-of-plane electron current, $J_{z,e}$, depicted in (d); this interesting region deserves further consideration. We highlight in figure 6 some of these nodes. The nodes themselves are depicted in black in figure 6(a), and the associated points are depicted in the same colour in the 2-D simulated plane in (b). We observe that these nodes correspond to very specific regions in cluster 4 characterized by signatures associated with plasmoid merging.

Going back to figure 5, we see that some of the nodes below the region just analysed are characterized both by large positive values of $p_{xx,e}$ and high, negative values of $J_{z,e}$. Again, in figure 7, we highlight these nodes in (a) and check which data points they correspond to in (b). Also in these cases, we obtain specific regions in cluster 4, purple, strongly associated with plasmoid merging. We can therefore reaffirm that cluster 4, purple, is the

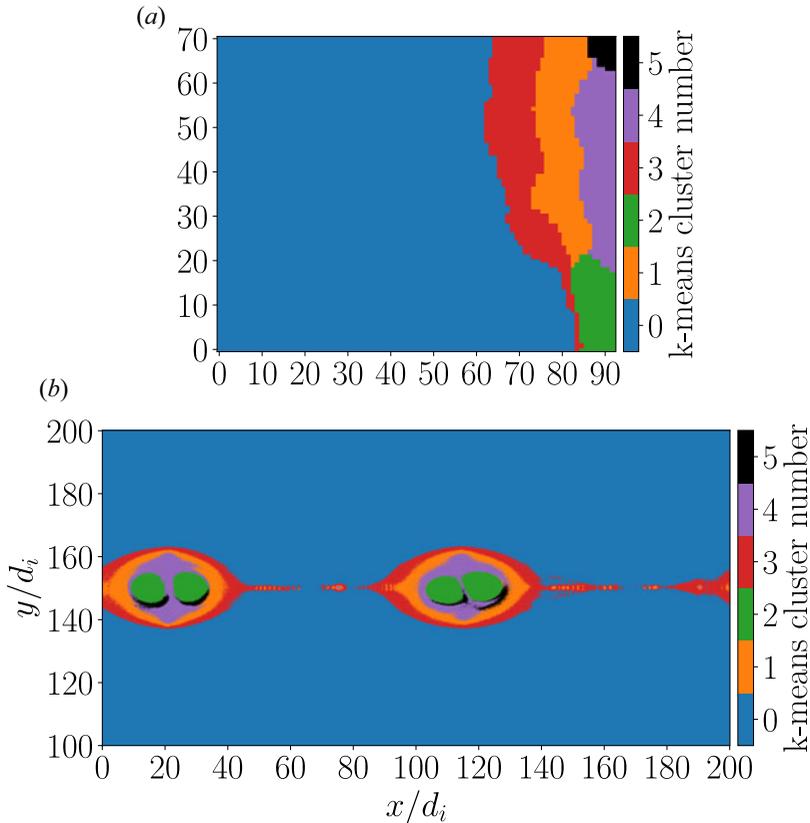


FIGURE 6. Identification in the simulated plane of regions of interest in the feature maps; the colour black is used to highlight nodes of interest in (a) and associated points in the simulation in (b).

one more directly associated with plasmoid merging, and that specific nodes in the SOMs map to rather localized regions in space where merging-specific signatures are particularly strong.

The variability of $J_{z,e}$ over the map (figure 5d) is extremely interesting to examine. First of all, we notice that $J_{z,e}$ in the feature map varies over values which seem quite different from those observed in figure 4. This may occur in SOM, and it has already been commented upon e.g. in Innocenti *et al.* (2021); the feature value associated with a node may be quite different from those of the data points associated with it, especially, as in this case, for outlier-rich features. We can identify in (d) specific patterns; for example, positive and negative values of $J_{z,e}$ are associated with the plasmoid merging regions examined in figures 6 and 7. A further pattern of interest in the $J_{z,e}$ feature map is the high-value region at the intersection between clusters 3, 2 and 1. Figure 8 shows that also these nodes map to a rather small region in space, namely smaller-scale plasmoids developing at the X-line, a feature commonly observed in PIC simulations e.g. in Innocenti *et al.* (2015a), Innocenti *et al.* (2017a) and Li *et al.* (2017).

In figure 9 we depict the position of data points as a function of the electron parallel plasma beta $\beta_{\parallel,e}$, x axis, and of the electron perpendicular to parallel temperature ratio, $T_{\perp,e}/T_{\parallel,e}$, y axis. In (a) we depict all points at initialization, in (b–f) the points associated with clusters 0, 1, 2, 3 and 4, respectively, at time $\Omega_{ci}t = 320$. The solid, dashed and

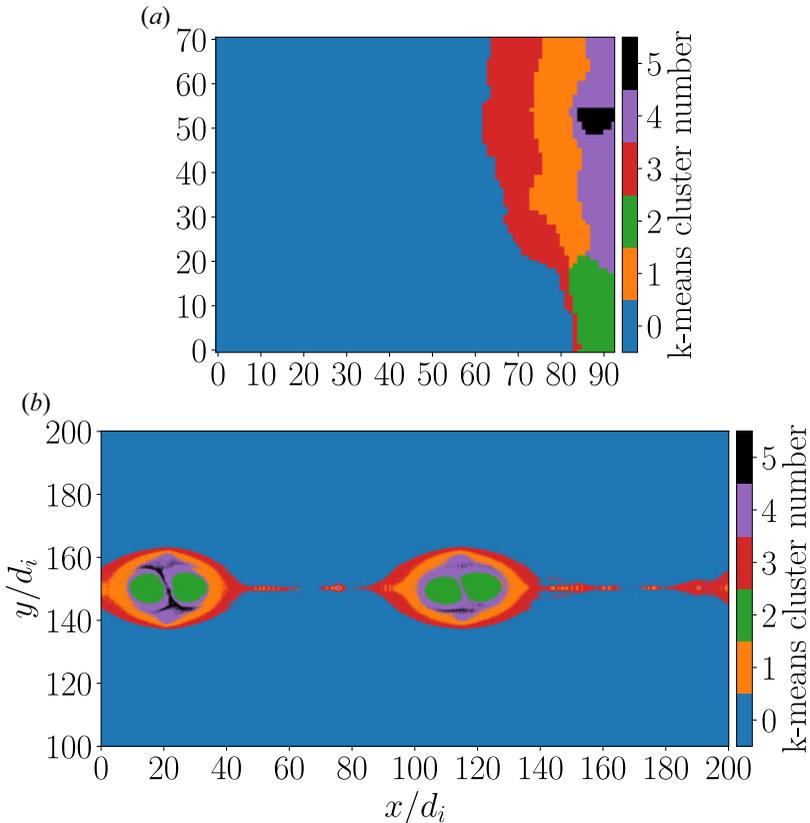


FIGURE 7. Identification in the simulated plane of regions of interest in the feature maps: the colour black is used to highlight nodes of interest in (a) and associated points in the simulation in (b).

dotted lines in the $T_{\perp,e}/T_{\parallel,e} < 1$ semi-quadrant are the isocontours of growth rates $\gamma/\Omega_{ce} = 0.001, 0.1, 0.2$ of the resonant electron firehose instability from Gary & Nishimura (2003). The solid and dotted lines in the $T_{\perp,e}/T_{\parallel,e} > 1$ semi-quadrant are the isocontours of growth rates $\gamma/\Omega_{ce} = 0.01, 0.1$ for the whistler temperature anisotropy instability (Gary & Wang 1996). The colours depict the number of points per pixel. This visualization is quite common for both electrons and ions in both the solar wind (e.g. Štverák *et al.* 2008; Innocenti *et al.* 2019a; Micera *et al.* 2021) and magnetospheric (e.g. Alexandrova *et al.* 2020) environment. When plotting electron quantities, we can quickly identify firehose- (bottom right corner) or whistler- (upper right corner) unstable plasma parcels. The area in between the two families of isocontour lines is stable to both instabilities. In this case, we can use this visualization to quickly appreciate the differences between the plots associated with the different clusters. In all panels the simulated data points sit in the stability region bounded by the stability thresholds. At initialization, (a), the data points are well into the stable region. In (b) we see that the majority of data points in cluster 0, inflow cluster, still sit quite close to initialization values at $\Omega_{ci}t = 320$, as expected from regions of space barely influenced by the development of the plasmoid instability. A small number of points (darker colour points) have moved to larger parallel beta, still within the stable region. In all the clusters associated with plasmoids the majority of points (lighter colour points) have moved to larger parallel electron beta with respect to initialization, due to the larger

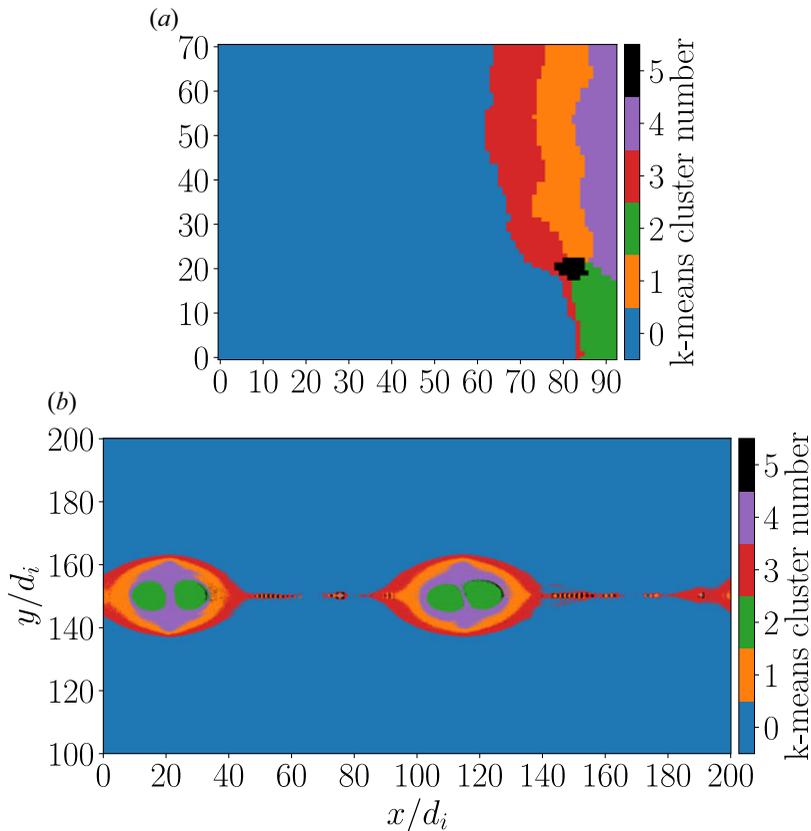


FIGURE 8. Identification in the simulated plane of regions of interest in the feature maps; the colour black is used to highlight nodes of interest in (a) and associated points in the simulation in (b).

electron pressure within the plasmoids with respect to the inflow region (see figure 3). Comparing (c–f), we notice immediately the difference between (d) (cluster 2, green, inner plasmoid region) and the others. In the inner plasmoid region the spread of the points around the core of the distribution is quite small with respect to the other clusters, and all points are quite far from instability thresholds. ‘Walking’ from the plasmoid interior towards the inflow region, and crossing from cluster 2, green, (d), to cluster 4, purple, (f), to cluster 1, orange, (c), and finally to cluster 3, red, (e), we notice that the data distribution progressively moves towards higher electron parallel beta, still within the stable region. At the separatrix cluster, (e), the data points have spread into the narrow stable region at high parallel beta.

6. Discussion and conclusion

In this paper, we have applied an unsupervised clustering method based on SOMs to data points obtained from a fully kinetic simulation of plasmoid instability. The clustering method and the simulation used for the clustering experiment are described in §§ 2 and 3. Pre-processing activities and clustering results obtained with three different types of data scaling are described in § 4. We remark on the fundamental role of scalars in determining clustering results. Our dataset includes outlier-poor and outlier-rich features, with the latter being particularly critical in the identification of magnetic reconnection

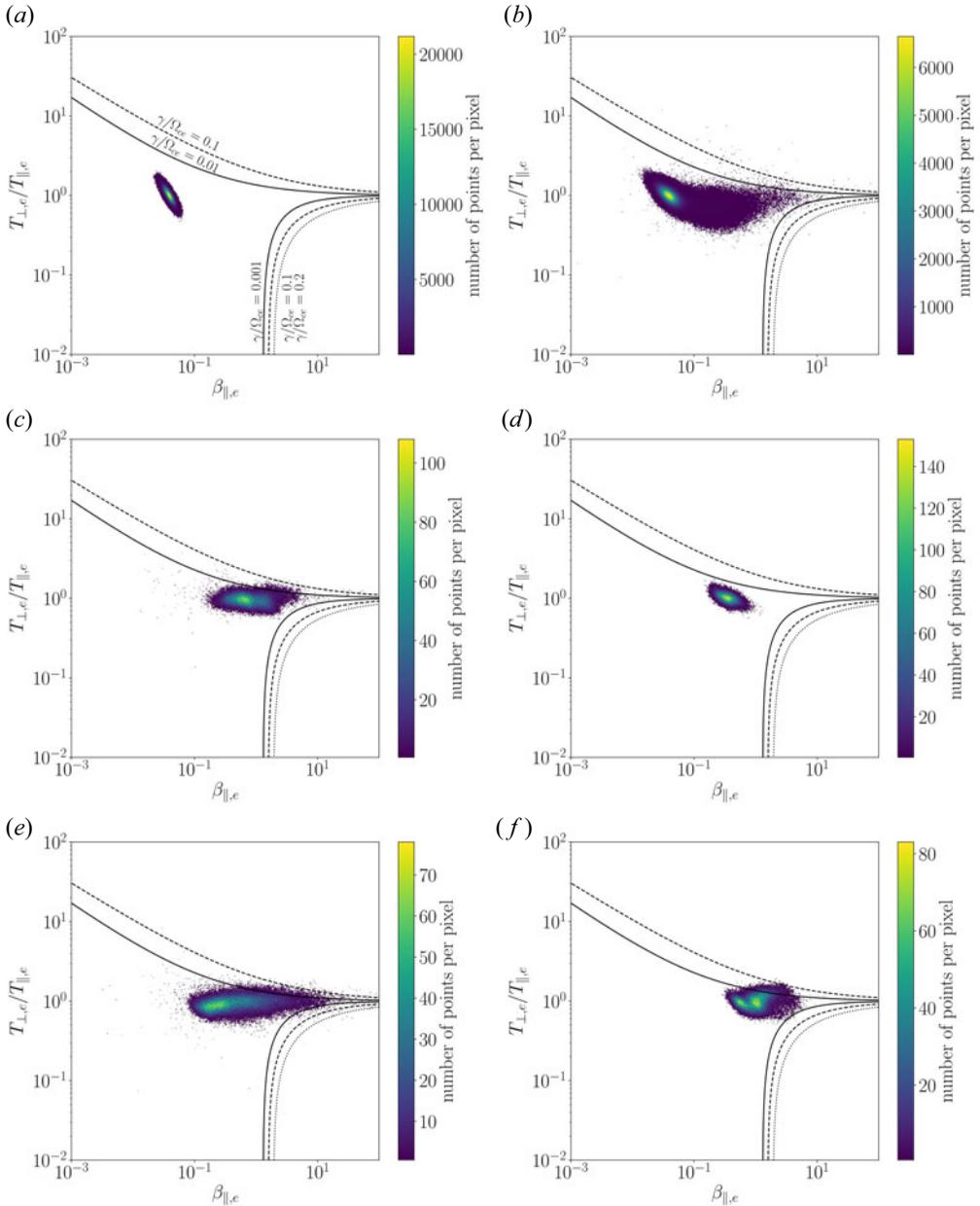


FIGURE 9. Data point distribution in the $\beta_{\parallel,e}$ vs $T_{\perp,e}/T_{\parallel,e}$ plane. (a) All upper current sheet simulated points at initialization. (b–f) Points associated with clusters 0, 1, 2, 3 and 4 at $\Omega_{cit} = 320$. The colours highlight the number of points per pixel. The isocontours of growth rates $\gamma/\Omega_{ce} = 0.001, 0.1, 0.2$ for the resonant electron firehose instability and of growth rates $\gamma/\Omega_{ce} = 0.01, 0.1$ for the whistler temperature anisotropy instability are depicted in the upper and lower semiquadrants.

regions. In this situation, a scaler designed to be robust to the presence of outliers, such as RobustScaler, delivers results that match better with our *a priori* knowledge of the process under investigation. The data scaled with RobustScaler are clustered in an unsupervised fashion into clusters that we identify, *a posteriori*, as an inflow cluster (cluster 0, blue in figure 4c), an inner plasmoid cluster (cluster 2, green), a separatrix cluster (cluster 3, red), a plasmoid merging cluster (cluster 4, purple) and a further cluster for regions intermediate between the separatrices and the area where plasmoid merging signatures are stronger (cluster 1, orange).

In § 5 we further examine our classification results in light of the analysis tools that SOMs provide us with, mainly feature maps, figure 5, and the UDM, figure 3. We recover from this analysis valuable information on the physical processes occurring in each cluster. In our case, where we are already acquainted with the dataset under investigation and with the physical results of interest, this *a posteriori* analysis results in an analysis procedure where information is made readily available in an easy-to-grasp 2-D representation, thus simplifying further investigation. Furthermore, we see the potential of using feature and UDM maps as a tool for scientific discovery when applied to datasets encoding unknown processes. We consider of particular interest the analysis of UDM patterns illustrated in figures 6 to 8; we show that darker patterns in the UDM (meaning, nodes ‘significantly’ different from their neighbours) map to small-scale regions of interest, such as regions in the 2-D simulated plane where plasmoid merging signatures are particularly strong, or where smaller-scale plasmoids develop in the current sheet in between larger-scale plasmoids. Also here, we see the double potential of the method, as a fast tool for identifying already known regions of interest in a potentially very large dataset, and as a tool of scientific discovery when applied to an unknown dataset.

This work is a follow up of previous activities, where a similar clustering method had been applied to MHD simulations. We verify here that this clustering technique delivers excellent and insightful results also with fully kinetic simulations, when it is fed data (moments separated by species, parallel and perpendicular pressure terms, electric field ‘including’ non-ideal processes, ...) comparable, in temporal and spatial resolution and in the nature of the processes of interest, to observed data from solar wind and magnetospheric missions.

Spacecraft observations are thus a natural field of future applicability for this method. Additionally, we envision a role of this and similar clustering methods in model development. An open issue in multi-physics and coupled methods (e.g. the above-mentioned Daldorff *et al.* 2014; Ashour-Abdalla *et al.* 2015; Innocenti *et al.* 2015b; Lautenbach & Grauer 2018) is how to decide when to switch between numerical methods (e.g. MHD to PIC, or 10-moment method to Vlasov) or between lower and higher resolution. This choice is often made somehow empirically, e.g. based on the expected location in space of target processes, or on thresholds of specific quantities. We envision the possibility of using SOMs trained on reference simulations to decide where to perform this switch; specific SOM nodes or specific clusters of SOM nodes could be associated with a certain model or resolution. We remark that, while training a SOM is (moderately) time consuming, BMU evaluation is in fact fast enough to be embedded into a run-time code without significant performance degradation. Preliminary and yet unpublished tests confirm that SOMs are robust to being deployed in simulations performed with a different physical model with respect to the one of training.

Acknowledgements

We thank G. Lapenta and the ECsim development team at KULeuven for early access to the code, the AIDA WP7 team (J. Amaya, J. Reader, B. Ferdousi and J. ‘Andy’ Edmond)

for long discussions on the methodology (<https://www.aida-space.eu>) and R. Sydora for useful feedback on the draft.

Editor Antoine C. Bret thanks the referees for their advice in evaluating this article.

Funding

This work was funded by the German Science Foundation DFG within the Collaborative Research Center SFB1491. We gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for providing computing time on the GCS Supercomputer SuperMUC-NG at Leibniz Supercomputing Centre (www.lrz.de). This work also used the DiRAC Extreme Scaling service at the University of Edinburgh, operated by the Edinburgh Parallel Computing Centre on behalf of the STFC DiRAC HPC Facility (www.dirac.ac.uk). This equipment was funded by BEIS capital funding via STFC capital grant ST/R00238X/1 and STFC DiRAC Operations grant ST/R001006/1. DiRAC is part of the UK National e-Infrastructure.

Declaration of interests

The authors report no conflict of interest.

Data availability statement

The data that support the findings of this study are openly available in Zenodo at <https://zenodo.org/record/7463339#.Y6R2GhWZPEb>, reference number 7463339 (Köhne, Boella & Innocenti 2022a). The scripts are available on GitHub at <https://github.com/SophiaKoehe/unsupervised-classification> (Köhne, Boella & Innocenti 2022b).

Author contribution

M.E.I. performed the simulations. S.K. performed the clustering. All authors contributed to analysing data and reaching conclusions, and in writing the paper.

Appendix. Robustness tests

In this Section we illustrate several tests run with the objective of assessing the robustness of our clustering procedure. First, we want to understand how dependent our clustering results are on the SOM hyper-parameters, see § 2. In table 3 we list several SOM training experiments, run using as input the upper current sheet data scaled with RobustScaler. We change the number of epochs, the initial learning rate η_0 , the initial neighbourhood width σ_0 (calculated as a difference percentage of the largest map side), the random seed used for random weight initialization and the number of SOM nodes q . We then list, in the last column, the percentage of points classified in the same k -means cluster as with a 5 epoch-long training, $\eta_0 = 0.5$, $\sigma_0 = 0.2 \times \max(x, y)$, random initialization seed. The k -means clustering is done afresh on the newly trained map. Since the cluster number in k -means is assigned arbitrarily, clusters which are assigned the same number in different clustering experiments do not necessarily correspond to similar regions of simulated space. For this reason, before calculating R , we reassign cluster numbers, so that the clusters compared during the R calculation indeed match to similar regions of physical space in the simulations.

In calculating the matching factor R we exclude the cluster mapping to the inflow region (cluster 0, blue), which stays essentially constant notwithstanding the SOM hyper-parameters, since we are primarily interested in clusters mapping to the plasmoid region. Without excluding the inflow cluster, the matching factors increase significantly in all cases, simply due to the amount of data points falling into the inflow region.

epochs	η_0	$\sigma_0/\% \max(x, y)$	random init seed	q	Matching factor $R/\%$
5	0.3	20	42	71×93	93.55
5	0.5	20	42	71×93	100
5	0.7	20	42	71×93	92.67
5	1	20	42	71×93	91.99
5	0.5	10	42	71×93	85.78
5	0.5	20	42	71×93	100
5	0.5	40	42	71×93	91.75
5	0.5	60	42	71×93	87.78
5	0.5	20	42	71×93	100
5	0.5	20	0	71×93	93.6
5	0.5	20	1	71×93	94.47
5	0.5	20	42	64×83	91.48
5	0.5	20	42	71×93	100
5	0.5	20	42	78×102	93.23
5	0.5	20	42	85×110	93.58
5	0.5	20	42	91×118	93.99
5	0.5	20	42	101×132	94.47
3	0.5	20	42	71×93	92.13
4	0.5	20	42	71×93	91.16
5	0.5	20	42	71×93	100
6	0.5	20	42	71×93	89.77
7	0.5	20	42	71×93	88.4

TABLE 3. Percentage R of samples classified in the same k -means cluster as with a map trained for five epochs, initial learning rate $\eta_0 = 0.5$, initial neighbourhood radius $\sigma_0 = 0.2 \times \max(x, y)$, random initialization seed 42 and number of nodes $q = 71 \times 93$ when the number of epochs, η_0 , σ_0 , random initialization seed and q are varied. In all cases, the scaler used is RobustScaler. The inflow cluster, cluster 0, is excluded in the calculation of the matching factor, since we are primarily interested in clusters mapping to the plasmoid region.

We notice that the matching factors are very high for all cases in table 3; our classification procedure is significantly stable to the choice of SOM hyper-parameters. The high matching factors can be explained noticing that map nodes may individually change with different hyper-parameters, but not so much as to give rise to significantly different k -means clustering results. The parameter that gives rise to the most significant differences is the initial neighbourhood radius. In deciding which initial neighbourhood radius to use in the body of the manuscript, we followed the rule of thumb prescription from Kohonen (2014). We see here that, with smaller neighbourhood radius and training on the same number of epochs, the map may have more difficulties reflecting the true structure and topology of the data.

It is interesting to identify which points in the simulation change clusters ‘more easily’. In figure 10 we plot the clustered points obtained from the SOM trained with the hyper-parameters listed in red and blue in table 3 compared with the reference SOM in (a). We notice that the regions of space which are more affected by the choice of hyper-parameters are those associated with cluster 1, orange and cluster 4, purple, i.e. the intermediate cluster region. This is not a surprise; the two clusters are the most similar and in fact tend to merge when the cluster number is reduced from $k = 5$ to $k = 4$. Cluster 2, inner plasmoid region, and cluster 3, separatrix cluster, are instead significantly different

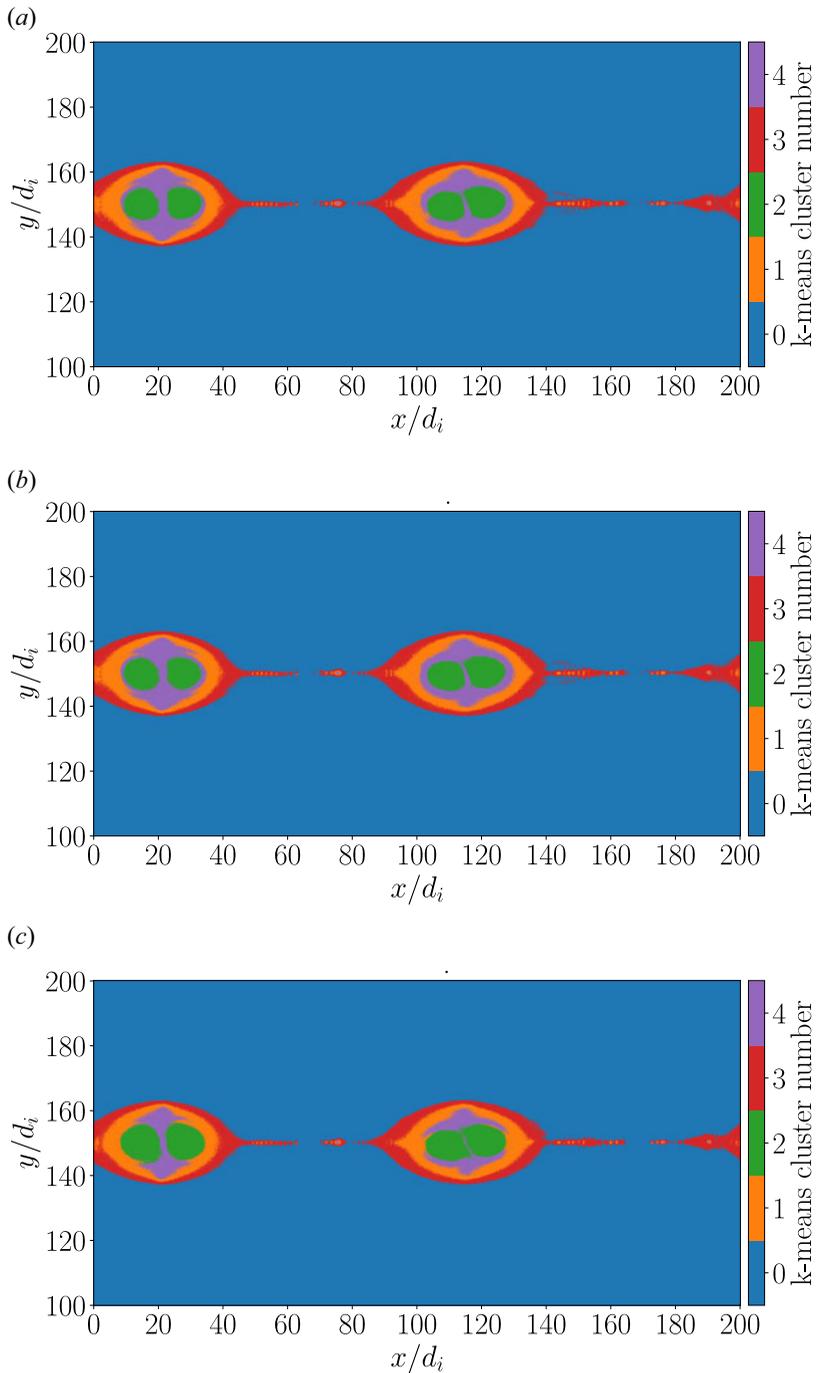


FIGURE 10. Clustering of the upper current sheet simulated data, with the SOM used as a reference in table 3 in (a) and the SOMs trained with the hyper-parameters listed in the lines coloured in blue and red in (b,c).

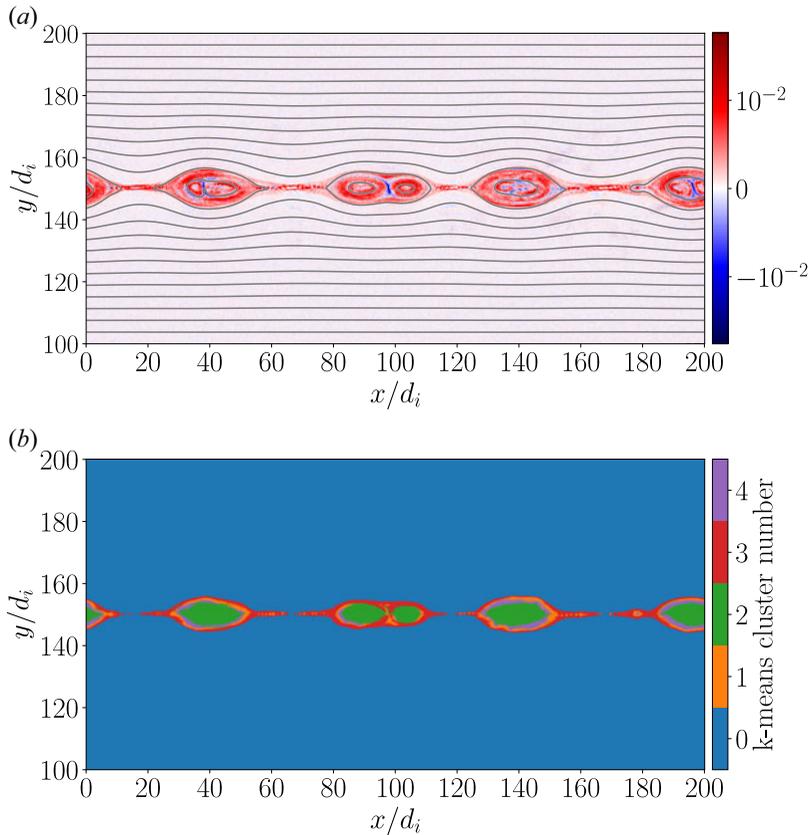


FIGURE 11. Out-of-plane electron current, $J_{z,e}$, (a), and clustering of the upper current sheet data points, (b), at $\Omega_{ci}t = 160$. The SOM used for the clustering has been trained at $\Omega_{cet} = 320$.

from the intermediate plasmoid region. The individual matching factors for each cluster in the case with the lowest matching factor $R = 85.78\%$ are: orange(1): 86.45%, green(2): 99.4%, red(3): 91.79%, purple (4): 64.37%. We see that results for clusters 0, 3, 2 (the clusters that physics tell us should be clearly distinct) are stable with all hyper-parameters, even in the worst case scenario, as we expect from the physics of the problem. From this point of view, the fact that a small subset of simulated points tends to be assigned, in different clustering experiments, to either one of the two most similar clusters is not considered a cause for concern.

We also want to check the robustness of our clustering procedure to temporal variation in the simulation. To do this, data samples from an earlier time in the simulation, $\Omega_{ci}t = 160$, were classified using the map and k -means centroids described in §§ 4 and 5, trained on the data from the time step $\Omega_{ci}t = 320$ scaled with RobustScaler. The BMUs of the data samples from the time step $\Omega_{ci}t = 160$ were found in that trained map and coloured according to the k -means clusters visible in figure 3(e). The out-of-plane electron current and the resulting classification are shown in figure 11(a,b).

We immediately notice that, at $\Omega_{ci}t = 160$, the simulation is at a quite different stage. More, smaller plasmoids are present, since the merging process is just at the beginning. This is reflected in the clustering results.

In figure 11(b), we observe that the inflow and plasmoid region are very well separated. Inside the plasmoid, the inner plasmoid region (cluster 2, green) occupied a larger percentage of the plasmoid area with respect to later times. The clusters associated with plasmoid merging, cluster 4, purple, and cluster 1, orange, have shrunk accordingly, possibly due to a combination of two factors; smaller plasmoid size (the plasmoid size increases after coalescing) and the fact that plasmoid merging is just at the beginning.

REFERENCES

- ALEXANDROVA, A., RETINO, A., DIVIN, A., MATTEINI, L., CONTEL, O.L., BREUILLARD, H., CATAPANO, F., COZZANI, G., ZAITSEV, I. & DECA, J. 2020 In situ evidence of firehose instability in multiple reconnection. [arXiv:2004.08280](https://arxiv.org/abs/2004.08280).
- AMAYA, J., DUPUIS, R., INNOCENTI, M.E. & LAPENTA, G. 2020 Visualizing and interpreting unsupervised solar wind classifications. *Front. Astron. Space Sci.* **7**, 66.
- ANGELIS, L. & STAMELOS, I. 2000 A simulation tool for efficient analogy based cost estimation. *Empir. Softw. Engng* **5**, 35–68.
- ARGALL, M.R., SMALL, C.R., PIATT, S., BREEN, L., PETRIK, M., KOKKONEN, K., BARNUM, J., LARSEN, K., WILDER, F.D., OKA, M., *et al.* 2020 MMS SITL ground loop: automating the burst data selection process. *Front. Astron. Space Sci.* **7**, 54.
- ASHOUR-ABDALLA, M., LAPENTA, G., WALKER, R.J., EL-ALAOUI, M. & LIANG, H. 2015 Multiscale study of electron energization during unsteady reconnection events. *J. Geophys. Res.* **120** (6), 4784–4799.
- BAKER, D.N., RIESBERG, L., PANKRATZ, C.K., PANNETON, R.S., GILES, B.L., WILDER, F.D. & ERGUN, R.E. 2016 Magnetospheric multiscale instrument suite operations and data system. *Space Sci. Rev.* **199** (1–4), 545–575.
- BIRN, J., DRAKE, J.F., SHAY, M.A., ROGERS, B.N., DENTON, R.E., HESSE, M., KUZNETSOVA, M., MA, Z.W., BHATTACHARJEE, A., OTTO, A., *et al.* 2001 Geospace environmental modeling (GEM) magnetic reconnection challenge. *J. Geophys. Res.* **106** (A3), 3715–3719.
- BISHOP, C.M. 2006 Pattern recognition. *Mach. Learn.* **128** (9).
- 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. *Front. Astron. Space Sci.* **7**, 55.
- BURCH, J.L., MOORE, T.E., TORBERT, R.B. & GILES, B.L. 2016 Magnetospheric multiscale overview and science objectives. *Space Sci. Rev.* **199** (1–4), 5–21.
- CAMPOREALE, E. 2019 The challenge of machine learning in space weather: nowcasting and forecasting. *Space Weather* **17** (8), 1166–1207.
- CAMPOREALE, E., CARÈ, A. & BOROVSKY, J.E. 2017 Classification of solar wind with machine learning. *J. Geophys. Res.* **122** (11), 10–910.
- CAZZOLA, E., INNOCENTI, M.E., GOLDMAN, M.V., NEWMAN, D.L., MARKIDIS, S. & LAPENTA, G. 2016 On the electron agyrotropy during rapid asymmetric magnetic island coalescence in presence of a guide field. *Geophys. Res. Lett.* **43** (15), 7840–7849.
- CAZZOLA, E., INNOCENTI, M.E., MARKIDIS, S., GOLDMAN, M.V., NEWMAN, D.L. & LAPENTA, G. 2015 On the electron dynamics during island coalescence in asymmetric magnetic reconnection. *Phys. Plasmas* **22** (9), 092901.
- CHEN, G., CHACÓN, L. & NGUYEN, T.B. 2021 An unsupervised machine-learning checkpoint-restart algorithm using Gaussian mixtures for particle-in-cell simulations. *J. Comput. Phys.* **436**, 110185.
- DALDORFF, L.K.S., TÓTH, G., GOMBOSI, T.I., LAPENTA, G., AMAYA, J., MARKIDIS, S. & BRACKBILL, J.U. 2014 Two-way coupling of a global hall magnetohydrodynamics model with a local implicit particle-in-cell model. *J. Comput. Phys.* **268**, 236–254.
- DA SILVA, D., BARRIE, A., SHUSTER, J., SCHIFF, C., ATTIE, R., GERSHMAN, D.J. & GILES, B. 2020 Automatic region identification over the MMS orbit by partitioning NT space. [arXiv:2003.08822](https://arxiv.org/abs/2003.08822).
- DRAKE, J.F., SWISDAK, M., CHE, H. & SHAY, M.A. 2006 Electron acceleration from contracting magnetic islands during reconnection. *Nature* **443** (7111), 553–556.

- DUPUIS, R., GOLDMAN, M.V., NEWMAN, D.L., AMAYA, J. & LAPENTA, G. 2020 Characterizing magnetic reconnection regions using Gaussian mixture models on particle velocity distributions. *Astrophys. J.* **889** (1), 22.
- FOX, N.J., VELLI, M.C., BALE, S.D., DECKER, R., DRIESMAN, A., HOWARD, R.A., KASPER, J.C., KINNISON, J., KUSTERER, M., LARIO, D., *et al.* 2016 The solar probe plus mission: humanity's first visit to our star. *Space Sci. Rev.* **204** (1–4), 7–48.
- GARY, S.P. & NISHIMURA, K. 2003 Resonant electron firehose instability: particle-in-cell simulations. *Phys. Plasmas* **10** (9), 3571–3576.
- GARY, S.P. & WANG, J. 1996 Whistler instability: electron anisotropy upper bound. *J. Geophys. Res.* **101** (A5), 10749–10754.
- GONZALEZ-HERRERO, D., BOELLA, E. & LAPENTA, G. 2018 Performance analysis and implementation details of the energy conserving semi-implicit method code (ECsim). *Comput. Phys. Commun.* **229**, 162–169.
- GOODFELLOW, I., BENGIO, Y. & COURVILLE, A. 2016 *Deep Learning*. MIT Press.
- GUO, F., JOKIPII, J.R. & KOTA, J. 2010 Particle acceleration by collisionless shocks containing large-scale magnetic-field variations. *Astrophys. J.* **725** (1), 128.
- GUO, F., LI, X., LI, H., DAUGHTON, W., ZHANG, B., LLOYD-RONNING, N., LIU, Y.-H., ZHANG, H. & DENG, W. 2016 Efficient production of high-energy nonthermal particles during magnetic reconnection in a magnetically dominated ion–electron plasma. *Astrophys. J. Lett.* **818** (1), L9.
- HESSE, M. & WINSKE, D. 1998 Electron dissipation in collisionless magnetic reconnection. *J. Geophys. Res.* **103**, 26479.
- HOCKNEY, R.W. & EASTWOOD, J.W. 2021 *Computer Simulation using Particles*. CRC Press.
- HUANG, J., LI, Y.-F. & XIE, M. 2015 An empirical analysis of data preprocessing for machine learning-based software cost estimation. *Inf. Softw. Technol.* **67**, 108–127.
- INNOCENTI, M.E., AMAYA, J., RAEDER, J., DUPUIS, R., FERDOUSI, B. & LAPENTA, G. 2021 Unsupervised classification of simulated magnetospheric regions. *Ann. Geophys.* **39** (5), 861–881.
- INNOCENTI, M.E., BECK, A., PONWEISER, T., MARKIDIS, S. & LAPENTA, G. 2015*b* Introduction of temporal sub-stepping in the multi-level multi-domain semi-implicit particle-in-cell code Parsek2D-MLMD. *Comput. Phys. Commun.* **189**, 47–59.
- INNOCENTI, M.E., CAZZOLA, E., MISTRY, R., EASTWOOD, J.P., GOLDMAN, M.V., NEWMAN, D.L., MARKIDIS, S. & LAPENTA, G. 2017*a* Switch-off slow shock/rotational discontinuity structures in collisionless magnetic reconnection: what to look for in satellite observations. *Geophys. Res. Lett.* **44** (8), 3447–3455.
- INNOCENTI, M.E., GOLDMAN, M., NEWMAN, D., MARKIDIS, S. & LAPENTA, G. 2015*a* Evidence of magnetic field switch-off in collisionless magnetic reconnection. *Astrophys. J. Lett.* **810** (2), L19.
- INNOCENTI, M.E., JOHNSON, A., MARKIDIS, S., AMAYA, J., DECA, J., OLSHEVSKY, V. & LAPENTA, G. 2017*b* Progress towards physics-based space weather forecasting with exascale computing. *Adv. Engng Softw.* **111**, 3–17.
- INNOCENTI, M.E., LAPENTA, G., MARKIDIS, S., BECK, A. & VAPIREV, A. 2013 A multi level multi domain method for particle in cell plasma simulations. *J. Comput. Phys.* **238**, 115–140.
- INNOCENTI, M.E., NORGREN, C., NEWMAN, D., GOLDMAN, M., MARKIDIS, S. & LAPENTA, G. 2016 Study of electric and magnetic field fluctuations from lower hybrid drift instability waves in the terrestrial magnetotail with the fully kinetic, semi-implicit, adaptive multi level multi domain method. *Phys. Plasmas* **23** (5), 052902.
- INNOCENTI, M.E., TENERANI, A., BOELLA, E. & VELLI, M. 2019*a* Onset and evolution of the oblique, resonant electron firehose instability in the expanding solar wind plasma. *Astrophys. J.* **883** (2), 146.
- INNOCENTI, M.E., TENERANI, A. & VELLI, M. 2019*b* A semi-implicit particle-in-cell expanding box model code for fully kinetic simulations of the expanding solar wind plasma. *Astrophys. J.* **870** (2), 66.
- KOEHNE, S., BOELLA, E. & INNOCENTI, M.E. 2022*a* <https://zenodo.org/record/7463339#.Y6RvwhWZPEa>.
- KOEHNE, S., BOELLA, E. & INNOCENTI, M.E. 2022*b* <https://github.com/SophiaKoehne/unsupervised-classification>.

- KOHONEN, T. 1982 Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **43** (1), 59–69.
- KOHONEN, T. 2014 *Matlab Implementations and Applications of the Self-Organizing Map*. Unigrafia Oy.
- LALTI, A., KHOTYAINTEV, Y.V., DIMMOCK, A.P., JOHLANDER, A., GRAHAM, D.B. & OLSHEVSKY, V. 2022 A database of MMS bow shock crossings compiled using machine learning. *J. Geophys. Res.* **127** (8), e2022JA030454.
- LAPENTA, G. 2017 Exactly energy conserving semi-implicit particle in cell formulation. *J. Comput. Phys.* **334**, 349–366.
- LAPENTA, G., GONZALEZ-HERRERO, D. & BOELLA, E. 2017 Multiple-scale kinetic simulations with the energy conserving semi-implicit particle in cell method. *J. Plasma Phys.* **83**, 705830205.
- LAUTENBACH, S. & GRAUER, R. 2018 Multiphysics simulations of collisionless plasmas. *Front. Phys.* **6**, 113.
- LEDVINA, S.A., MA, Y.-J. & KALLIO, E. 2008 Modeling and simulating flowing plasmas and related phenomena. *Space Sci. Rev.* **139**, 143–189.
- LI, H., WANG, C., TU, C. & XU, F. 2020 Machine learning approach for solar wind categorization. *Earth Space Sci.* **7** (5), e2019EA000997.
- LI, X., GUO, F., LI, H. & LI, G. 2015 Nonthermally dominated electron acceleration during magnetic reconnection in a low- β plasma. *Astrophys. J. Lett.* **811** (2), L24.
- LI, X., GUO, F., LI, H. & LI, G. 2017 Particle acceleration during magnetic reconnection in a low-beta plasma. *Astrophys. J.* **843** (1), 21.
- LLOYD, S. 1982 Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28** (2), 129–137.
- LOUREIRO, N.F. & UZDENSKY, D.A. 2015 Magnetic reconnection: from the sweet–Parker model to stochastic plasmoid chains. *Plasma Phys. Control. Fusion* **58** (1), 014021.
- LU, S., ANGELOPOULOS, V., ARTEMYEV, A.V., PRITCHETT, P.L., LIU, J., RUNOV, A., TENERANI, A., SHI, C. & VELLI, M. 2019 Turbulence and particle acceleration in collisionless magnetic reconnection: effects of temperature inhomogeneity across pre-reconnection current sheet. *Astrophys. J.* **878**, 109.
- MARKIDIS, S., LAPENTA, G. & RIZWAN-UDDIN. 2010 Multi-scale simulations of plasma with iPIC3D. *Math. Comput. Simul.* **80** (7), 1509–1519.
- MICERA, A., BOELLA, E., ZHUKOV, A.N., SHAABAN, S.M., LÓPEZ, R.A., LAZAR, M. & LAPENTA, G. 2020 Particle-in-cell simulations of the parallel proton firehose instability influenced by the electron temperature anisotropy in solar wind conditions. *Astrophys. J.* **893**, 130.
- MICERA, A., ZHUKOV, A.N., LÓPEZ, R.A., BOELLA, E., TENERANI, A., VELLI, M., LAPENTA, G. & INNOCENTI, M.E. 2021 On the role of solar wind expansion as a source of whistler waves: scattering of suprathermal electrons and heat flux regulation in the inner heliosphere. *Astrophys. J.* **919** (1), 42.
- MISTRI, M. 2018 CUDA-SOM. <https://github.com/mistrello96/CUDA-SOM/releases>, blacklast accessed: Mar 22, 2023 (bugfix in src/utility_functions.cu in line 49: change “i<nElements;” to “j<nElements;”).
- MÜLLER, D., ST. CYR, O.C., ZOGANELIS, I., GILBERT, H.R., MARSDEN, R., NIEVES-CHINCHILLA, T., ANTONUCCI, E., AUCHÈRE, F., BERGHMANS, D., HORBURY, T.S., *et al.* 2020 The solar orbiter mission-science overview. *Astron. Astrophys.* **642**, A1.
- NGUYEN, G., AUNAI, N., MICHOTTE DE WELLE, B., JEANDET, A., LAVRAUD, B. & FONTAINE, D. 2022 Massive multi-mission statistical study and analytical modeling of the earth’s magnetopause. 1. A gradient boosting based automatic detection of near-earth regions. *J. Geophys. Res.* **127** (1), e2021JA029773.
- OLSHEVSKY, V., KHOTYAINTEV, Y.V., LALTI, A., DIVIN, A., DELZANNO, G.L., ANDERZÉN, S., HERMAN, P., CHIEN, S.W.D., AVANOV, L., DIMMOCK, A.P., *et al.* 2021 Automated classification of plasma regions using 3D particle energy distributions. *J. Geophys. Res.* **126** (10), e2021JA029620.
- PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., *et al.* 2011 Scikit-learn: machine learning in Python. *J. Machine Learning Res.* **12**, 2825–2830.

- PETROPOULOU, M., CHRISTIE, I.M., SIRONI, L. & GIANNIOS, D. 2018 Plasmoid statistics in relativistic magnetic reconnection. *Mon. Not. R. Astron. Soc.* **475** (3), 3797–3812.
- PUCCI, F., VELLI, M., SHI, C., SINGH, K.A.P., TENERANI, A., ALLADIO, F., AMBROSINO, F., BURATTI, P., FOX, W., JARA-ALMONTE, J., *et al.* 2020 Onset of fast magnetic reconnection and particle energization in laboratory and space plasmas. *J. Plasma Phys.* **86** (6).
- RAEDER, J. 2003 *Global Magnetohydrodynamics—A Tutorial*, 212–246. Springer.
- RICCI, P., BRACKBILL, J.U., DAUGHTON, W. & LAPENTA, G. 2004 Collisionless magnetic reconnection in the presence of a guide field. *Phys. Plasmas* **11**, 4102.
- ROBERTS, D.A., KARIMABADI, H., SIPES, T., KO, Y.-K. & LEPRI, S. 2020 Objectively determining states of the solar wind using machine learning. *Astrophys. J.* **889** (2), 153.
- SATOPAA, V., ALBRECHT, J., IRWIN, D. & RAGHAVAN, B. 2011 Finding a “kneedle” in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pp. 166–171. doi:10.1109/ICDCSW.2011.20.
- SHLENS, J. 2014 A tutorial on principal component analysis. [arXiv:1404.1100](https://arxiv.org/abs/1404.1100).
- STONE, E.C., FRANDSEN, A.M., MEWALDT, R.A., CHRISTIAN, E.R., MARGOLIES, D., ORMES, J.F. & SNOW, F. 1998 The advanced composition explorer. *Space Sci. Rev.* **86** (1–4), 1–22.
- ŠTVERÁK, Š., TRÁVNÍČEK, P., MAKSIMOVIC, M., MARSCH, E., FAZAKERLEY, A.N. & SCIME, E.E. 2008 Electron temperature anisotropy constraints in the solar wind. *J. Geophys. Res.* **113** (A3).
- VAN HULLE, M.M. 2012 *Self-Organizing Maps*, 585–622. Springer.
- VETTIGLI, G. 2018 Minisom: minimalistic and numpy-based implementation of the self organizing map. GitHub.[Online]. <https://github.com/JustGlowing/minisom/>.
- VILLMANN, T. & CLAUSSEN, J.C. 2006 Magnification control in self-organizing maps and neural gas. *Neural Comput.* **18** (2), 446–469.
- NVIDIA, VINGELMANN, P. & FITZEK, F.H.P. 2020 CUDA, release: 10.2.89.
- XU, D. & TIAN, Y. 2015 A comprehensive survey of clustering algorithms. *Ann. Data Sci.* **2**.