Using clustering techniques for magnetic reconnection detection

EGU 2021 - NP6.1

M. Sisti [1,2] F. Finelli [1] G. Pedrazzi [3] M. Faganello [1]
F. Califano [2] F. Delli Ponti [3] V. Zalizniak [3]

[1]Aix-Marseille University, CNRS, PIIM UMR 7345, Marseille, France, EU

[2]Dipartimento di Fisica, Università di Pisa, Italy, EU

[3]HPC Department, Cineca, Bologna, Italy, EU

27/04/2021

Table of contents

Using Machine Learning
to recognize Magnetic Reconnection

Simulating a turbulent plasma: Hybrid Vlasov-Maxwell code [Valentini et al., 2007]

Normalized[1] equations used:

- Vlasov equation for the ion distribution function:

$$\frac{\partial f}{\partial t} + \mathsf{v} \cdot \nabla f + (\mathsf{E} + \mathsf{v} \wedge \mathsf{B}) \cdot \frac{\partial f}{\partial \mathsf{v}} = 0 \tag{1}$$

- The Ohm's law for the electric field:

$$\mathsf{E} - d_e^2 \nabla^2 \mathsf{E} = -(\mathsf{u} \wedge \mathsf{B}) + \frac{1}{n}(\mathsf{J} \wedge \mathsf{B}) + \frac{1}{n} d_e^2 \nabla \cdot \Pi - \frac{1}{n} \nabla P_e + \frac{d_e^2}{n} \nabla \cdot [\mathsf{uJ} + \mathsf{Ju}] - \frac{1}{n} d_e^2 \nabla \cdot (\frac{\mathsf{JJ}}{n}) \tag{2}$$

  the ion density n, the ion bulk velocity u and the ion pressure tensor Π are obtained as the moments of the ion distribution function f.

- Faraday's equation:

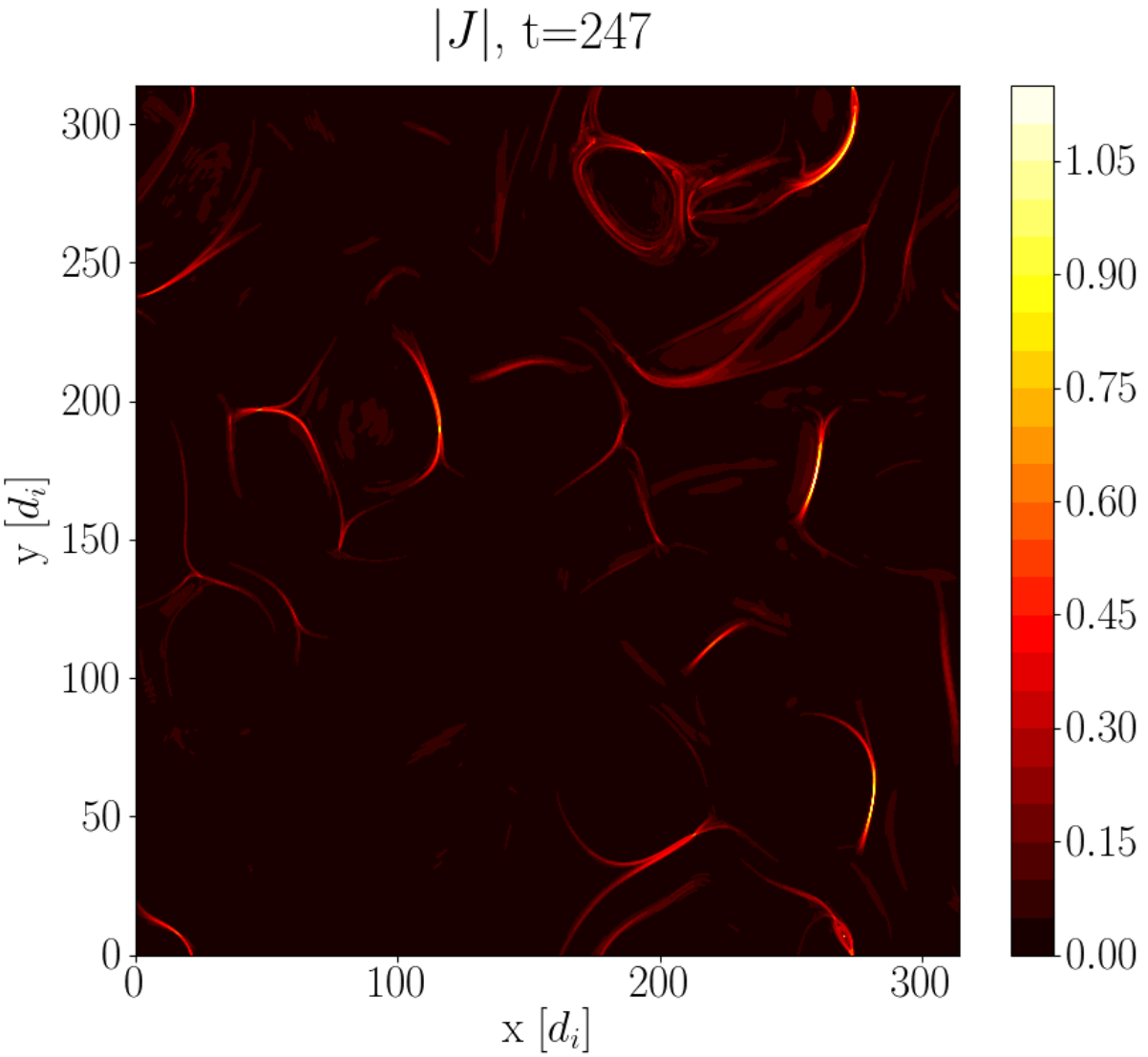
$$\frac{\partial \mathsf{B}}{\partial t} = -\nabla \wedge \mathsf{E} \qquad \mathsf{J} = \nabla \wedge \mathsf{B} \tag{3}$$

- $n_i \simeq n_e$

Box: 3072*3072 grid points, resolution $\sim 0.1 d_i$

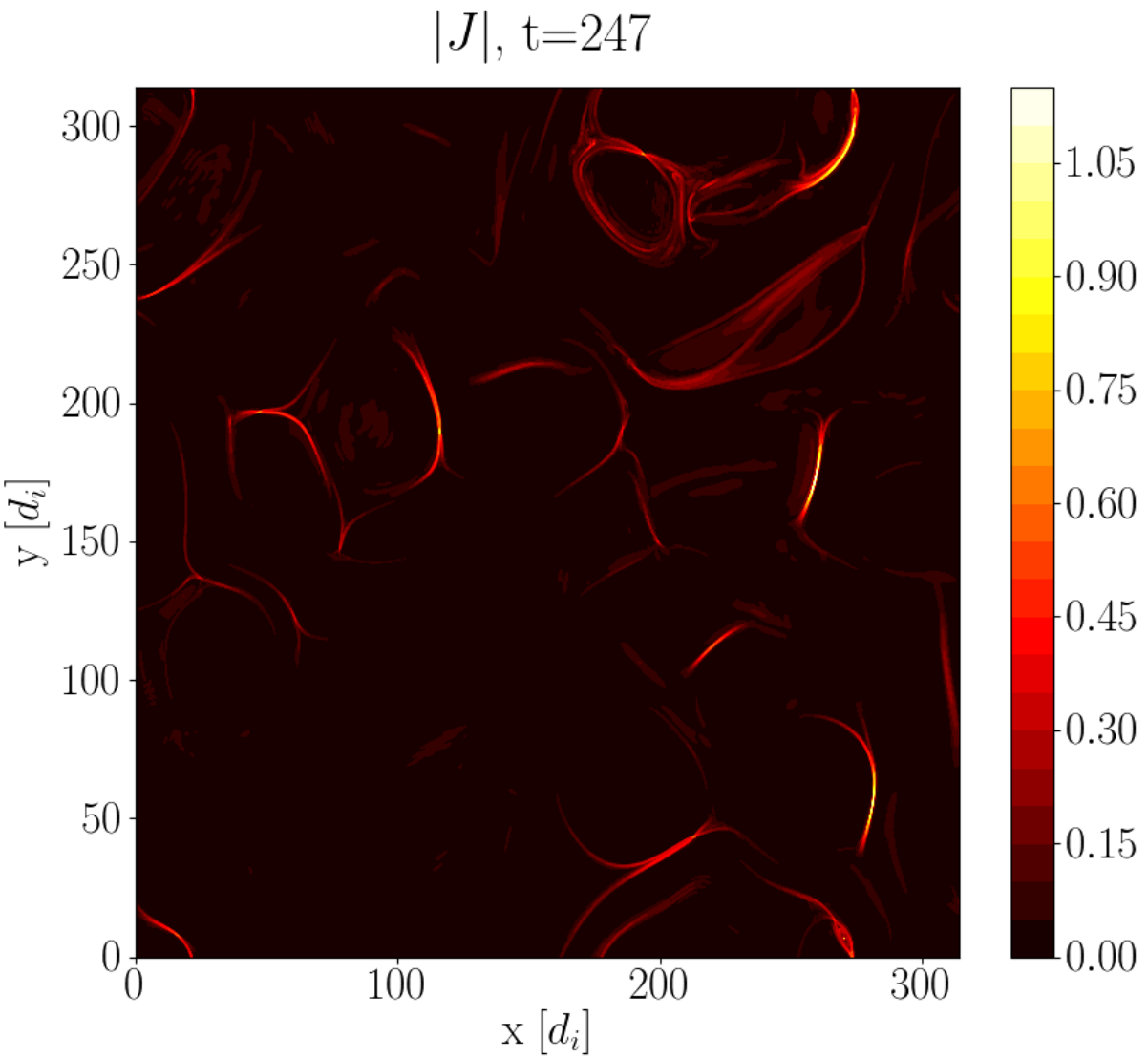
[1] to ion mass, ion cyclotron frequency Ω_{ci}, Alfvèn velocity and ion skin depth d_i

Motivations (context: magnetized turbulent space plasmas)



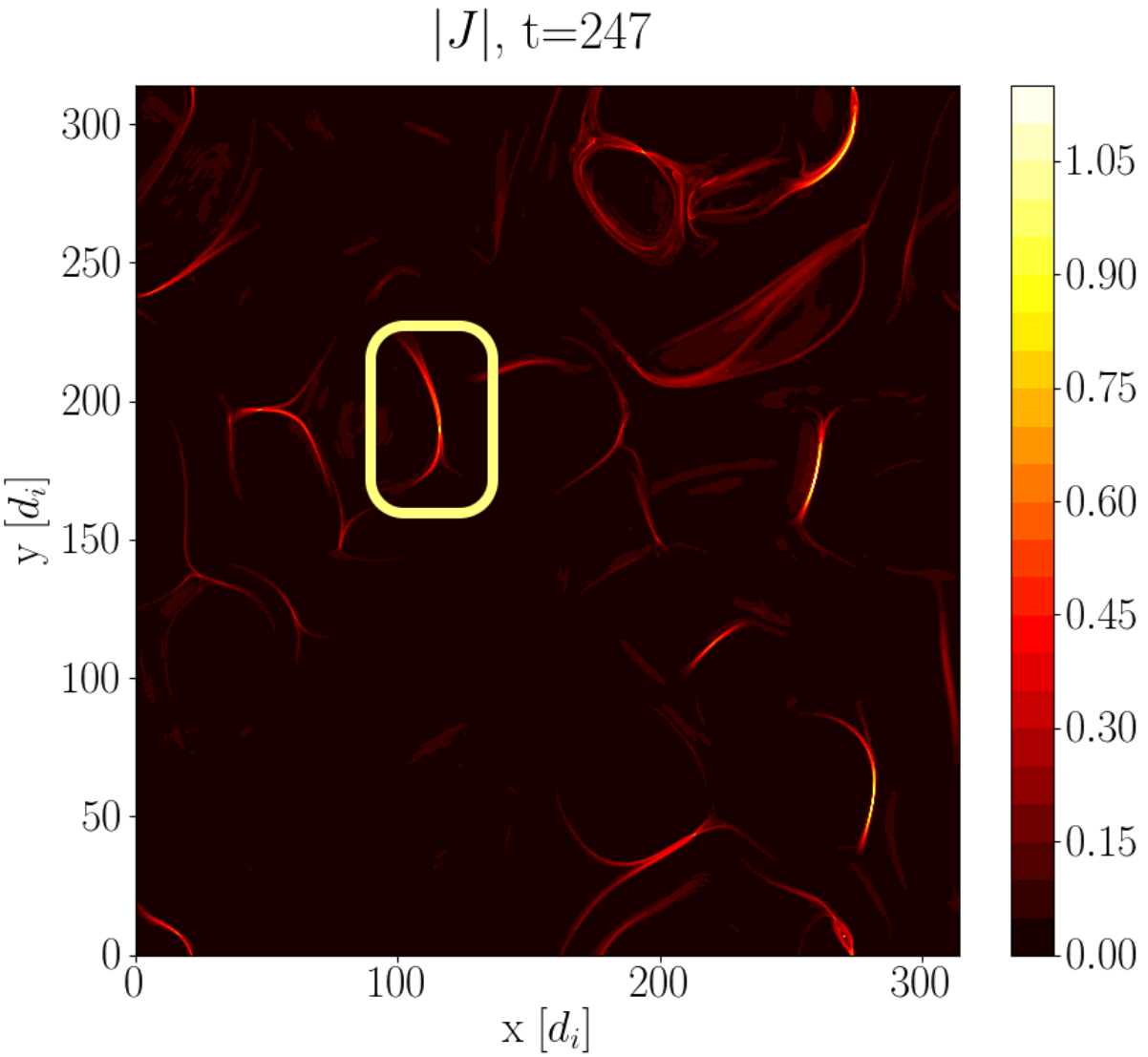
Reconnection occurs in the presence of strong current density

Motivations (context: magnetized turbulent space plasmas)



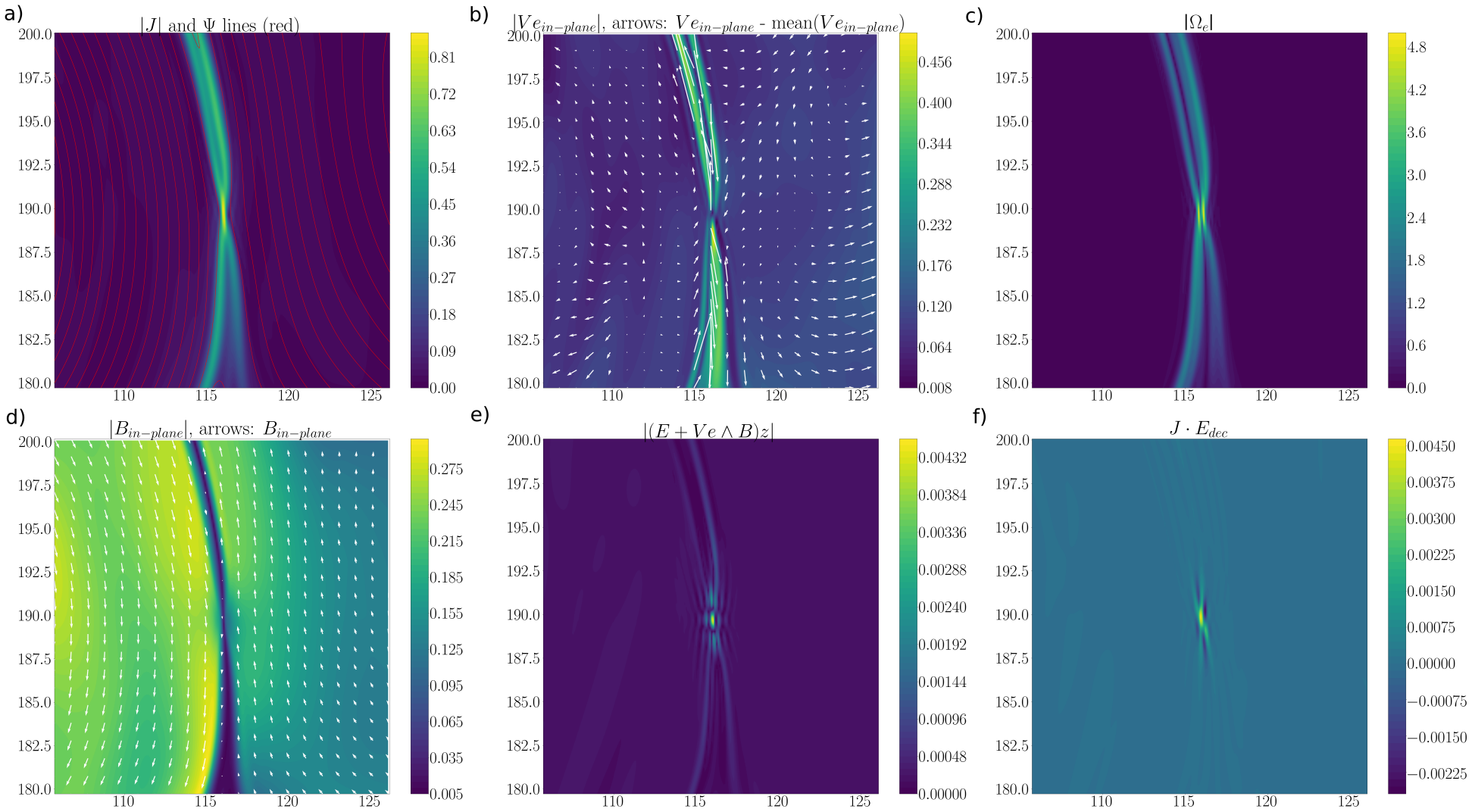
BUT, WHERE?

Motivations (context: magnetized turbulent space plasmas)



We have to check possible sites one by one...

Motivations (context: magnetized turbulent space plasmas)



We were lucky! It's a reconnection site!

Motivations (context: magnetized turbulent space plasmas)

- At today magnetic reconnection can be accurately identified by human analysis, but no well verified technique to automatically detect it has been developed[2]
- Big data $\rightarrow$ need to speed up analysis

Setting-up some algorithms (which use Machine Learning) aimed at automatically detecting the presence of current sheet (CS) and magnetic structures where reconnection is occurring (2D)

[Sisti et al., 2021]

The research is developed in the framework of an european project called AIDA project (Artificial Intelligence Data Analysis) `http://www.aida-space.eu/`.

[2] Some times Ψ flux function has been used BUT it cannot in 3D sims or in satellites' data!

Differences between supervised and unsupervised ML

Supervised Machine Learning

- A ML algorithm is said to be "supervised" if it uses a ground truth or "labeled" data.
- Training set of labeled data $\rightarrow$ the algorithm learns the rules $\rightarrow$ the rules can be applied to other data sets.

$\Downarrow$

In collaboration with Centrum Wiskunde & Informatica, Amsterdam, The Netherlands [Hu et al., 2020]

Unsupervised Machine Learning

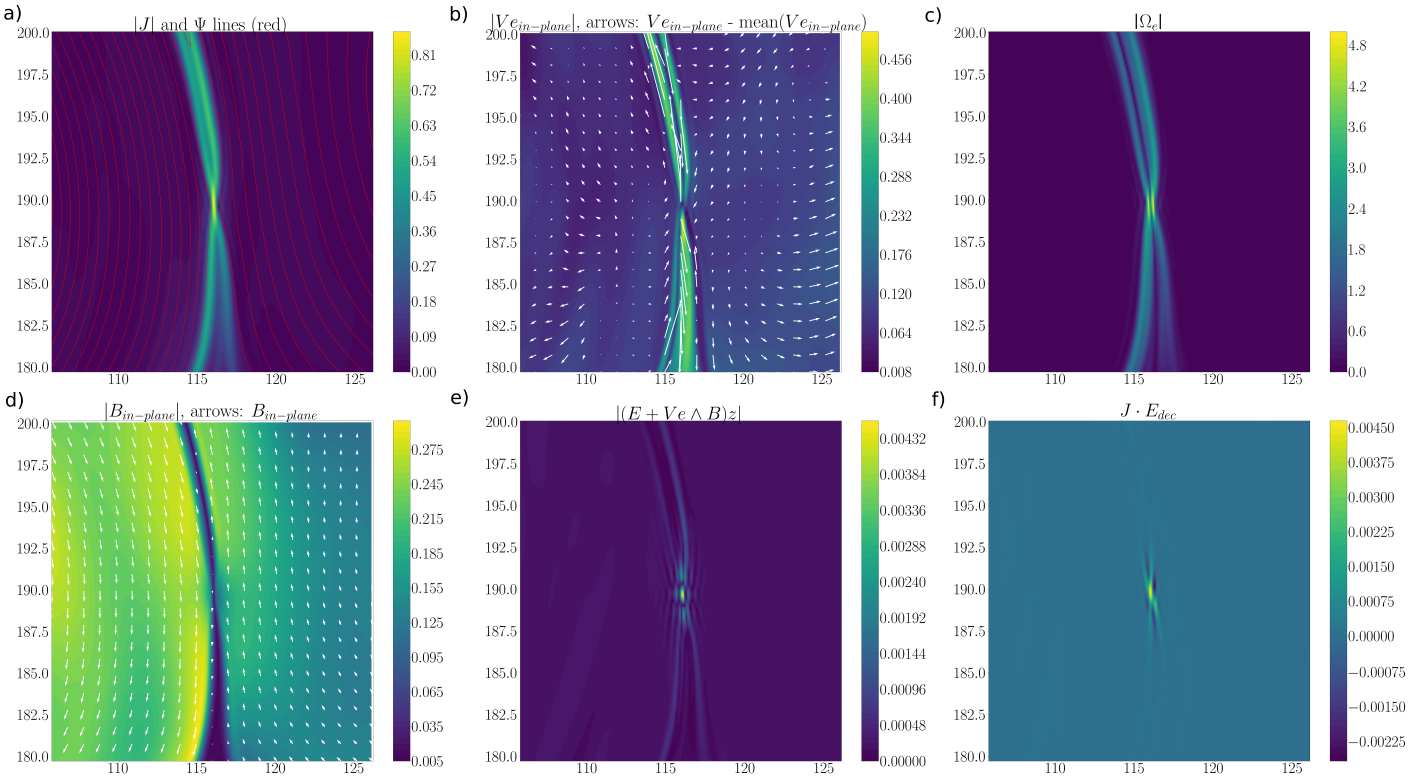
- A machine learning algorithm is said to be unsupervised if it does not use a ground truth or "labeled" data.
- It looks on its own for patterns in the data that hint at some underlying structure.

$\Downarrow$

In collaboration with CINECA, Bologna, Italy [Sisti et al., 2021]

Algorithm with unsupervised ML: variables

(a) current density $|J|$
(b) in-plane electron velocity
(c) electron vorticity $\Omega_e = \nabla \wedge V_e$
(d) in-plane magnetic field
(e) electron decoupling: $E' = E + V_e \wedge B$, z-component
(f) $J \cdot E'$



This is a very clear reconnection site, but often we are not so lucky!

Algorithm with unsupervised ML: algorithm description

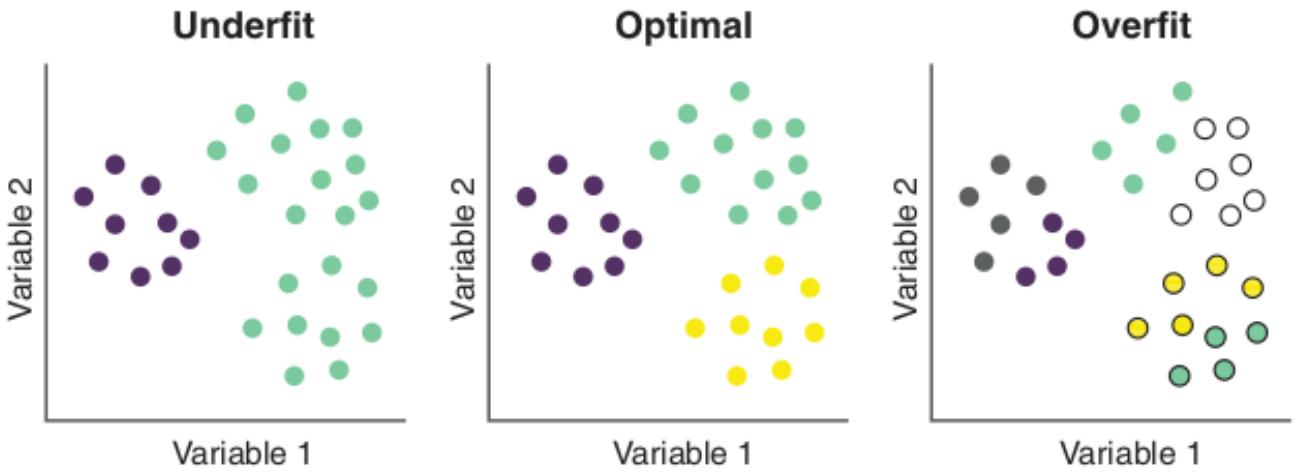
1. Tuning k for the k-means model
2. K-means (Lloyd's algorithm) $\rightarrow$ variable space
3. DBscan algorithm $\rightarrow$ physical space
4. Threshold on clusters' aspect ratio

Both K-means and DBscan are techniques of unsupervised ML with the aim to learn a grouping structure in a dataset (clustering techniques).

1. Tuning k for the k-means model

Annoing-fact: K-means approach need us to define how many clusters (k) we believe there are in our dataset.

Risk: underfitting or overfitting of the data

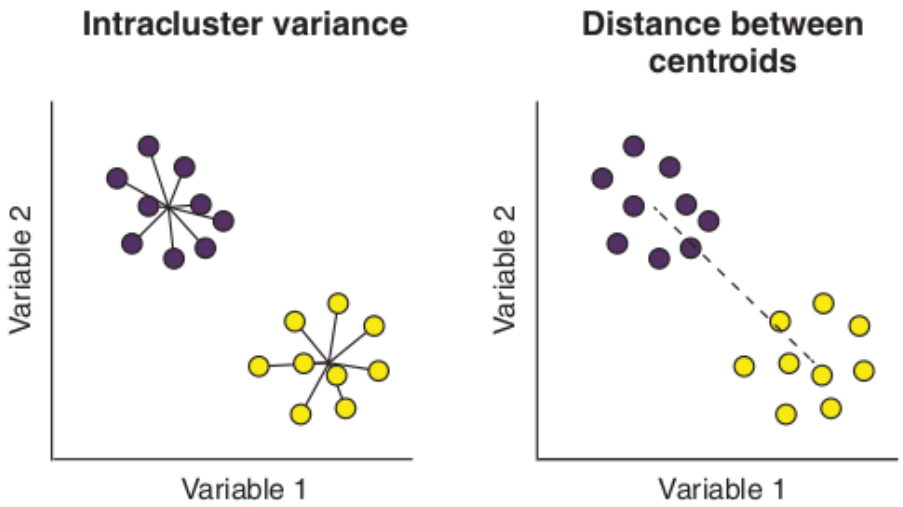


What under- and overfitting looks like for clustering tasks. In the left-side plot, the clusters are underfit (fewer clusters have been identified than actually exist). In the right-side plot, the clusters are overfit (real clusters are broken up into smaller clusters). In the center plot, an optimal clustering model has been found that faithfully represents the structure in the data.

Figure: from [Rhys, 2020]

1. Tuning k for the k-means model

When we don't know how many clusters we expect to find, we can use optimization techniques, such as Davis-Bouldin index.

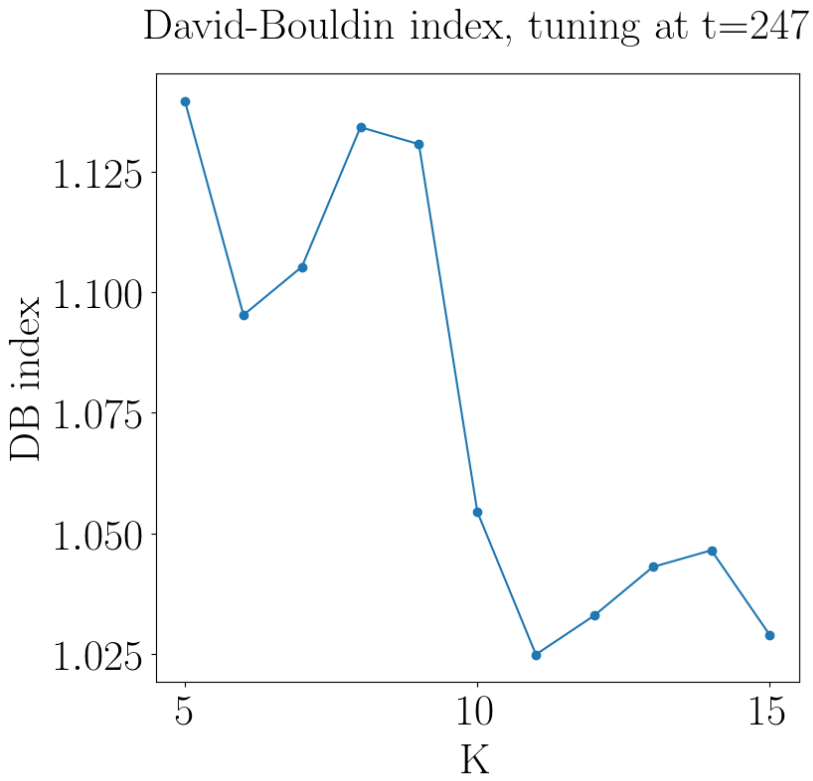


The Davies-Bouldin index calculates the intracluster (within-cluster) variance (left-side plot) and the distance between the centroids of each cluster (right-side plot). For each cluster, its nearest neighboring cluster is identified, and the sum of their intracluster variances is divided by the difference between their centroids. This value is calculated for each cluster, and the Davies-Bouldin index is the mean of these values.

Figure: from [Rhys, 2020]

The smaller the value of the Davies-Bouldin index, the better the separation between clusters.

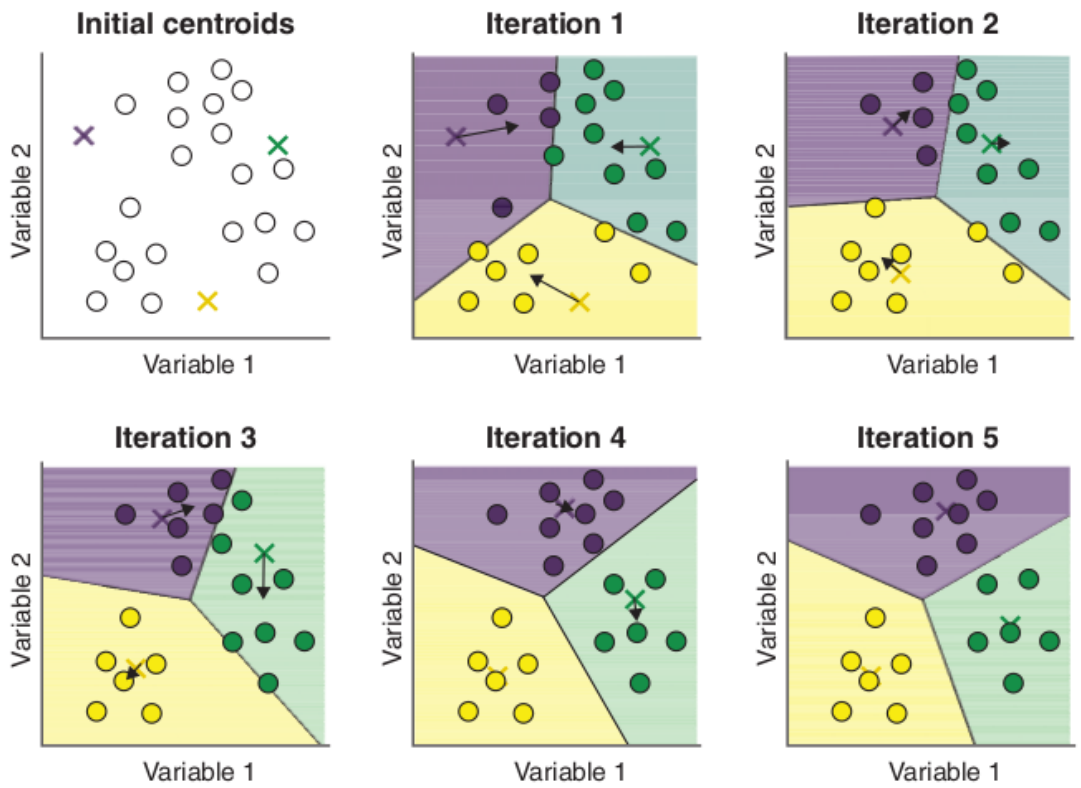
1. Tuning k for the k-means model: our results



- Tuning at a "central" time for our simulation ($t \simeq 247\Omega_{ci}$), for which the current sheets are well developed
- k = 11

Figure: Davies-Bouldin index for our data set, $t \simeq 247$ (Sisti et al. 2020 ApJ, accepted)

2. K-means (Lloyd's algorithm)

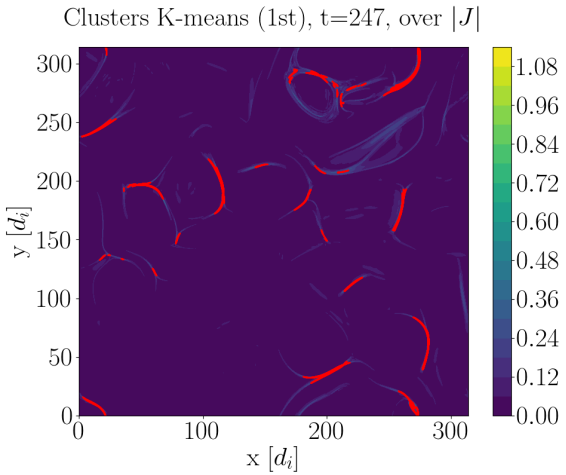


Five iterations of k-means clustering. In the top-left plot, three initial centers are randomly generated in the feature space (crosses). Cases are assigned to the cluster of their nearest center. At each iteration, each center moves to the mean of the cases in its cluster (indicated by arrows).

Figure: from [Rhys, 2020]

2. K-means (Lloyd's algorithm): our results

Cluster	$\overline{\lvert\mathbf{J}\rvert}$	$\overline{\lvert\mathbf{V}_e\rvert}$	$\overline{\lvert\mathbf{\Omega}_e\rvert}$	$\overline{E_{dec,e}}10^{-2}$	$\overline{\lvert\mathbf{B}_{in-plane}\rvert}$	$\overline{\lvert\mathbf{J}\cdot(\mathbf{E}+\mathbf{V}_e\times\mathbf{B})\rvert}10^{-3}$	Grid point number
1	**0.369**	**0.253**	**1.239**	**0.031**	**0.138**	**0.930**	**37 776**
2	0.059	0.125	0.162	0.021	0.402	0.170	160 928
3	0.036	0.239	0.049	0.022	0.109	0.079	593 803
4	0.033	0.173	0.055	0.021	0.242	0.060	660 889
5	0.027	0.078	0.045	0.021	0.286	0.053	685 840
6	0.031	0.145	0.040	0.021	0.049	0.062	813 751
7	0.025	0.153	0.031	0.021	0.168	0.041	1 165 723
8	0.022	0.063	0.023	0.021	0.072	0.039	1 243 427
9	0.022	0.069	0.029	0.021	0.207	0.039	1 278 303
10	0.022	0.070	0.024	0.021	0.144	0.037	1 325 111
11	0.021	0.128	0.022	0.021	0.108	0.034	1 471 633



From [Sisti et al., 2021]

- k-means can be applied to every time
- We choose the interesting cluster: 1
- In the physical space the cluster 1 is given by ALL the red regions in the figure (all together!)
- We see the cluster 1 is made up by different structures $\rightarrow$ we need to differentiate these structures in the physical space $\rightarrow$ DBscan algorithm

3. DBscan (Density Based Spatial Clustering of Applications with Noise)

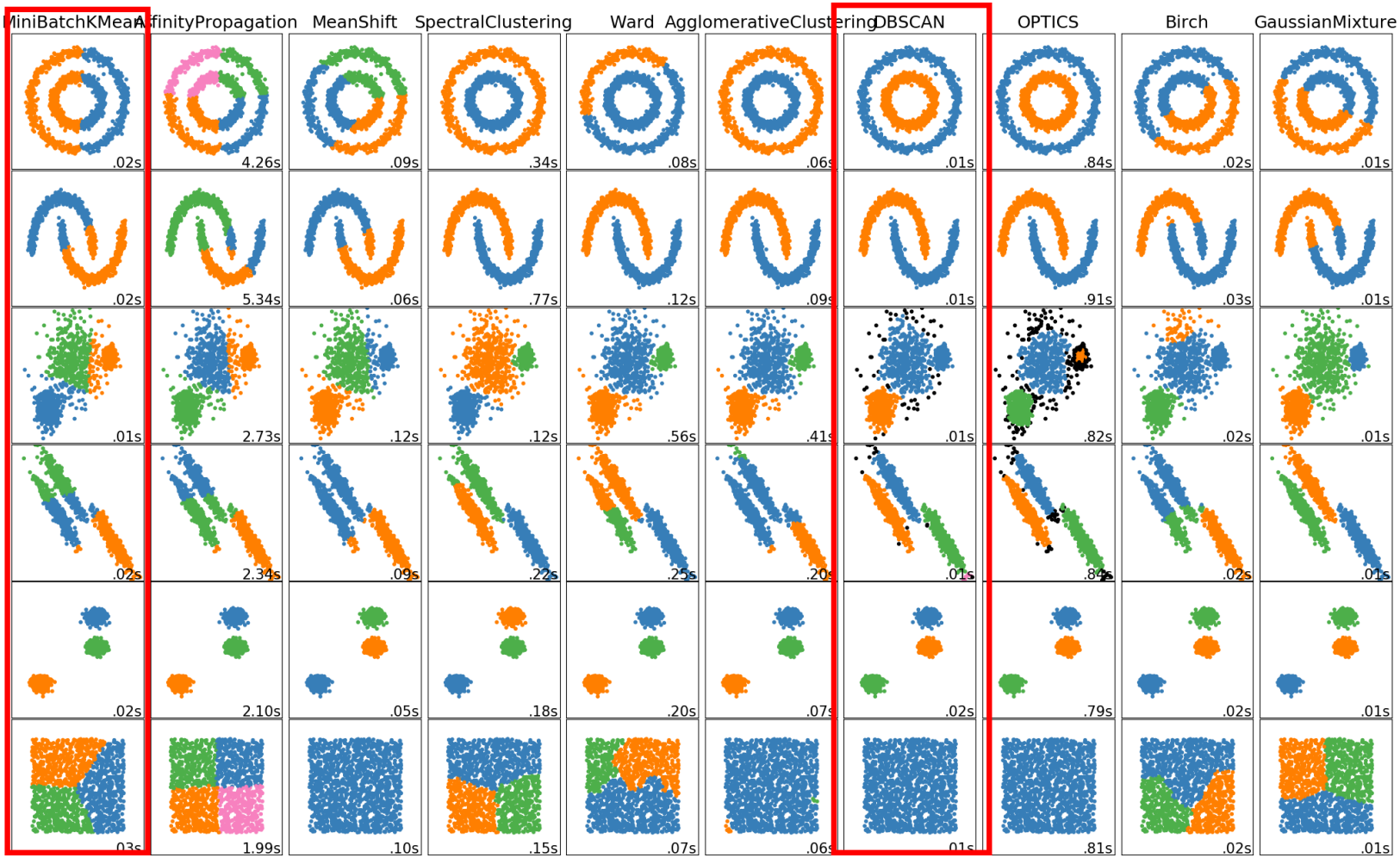
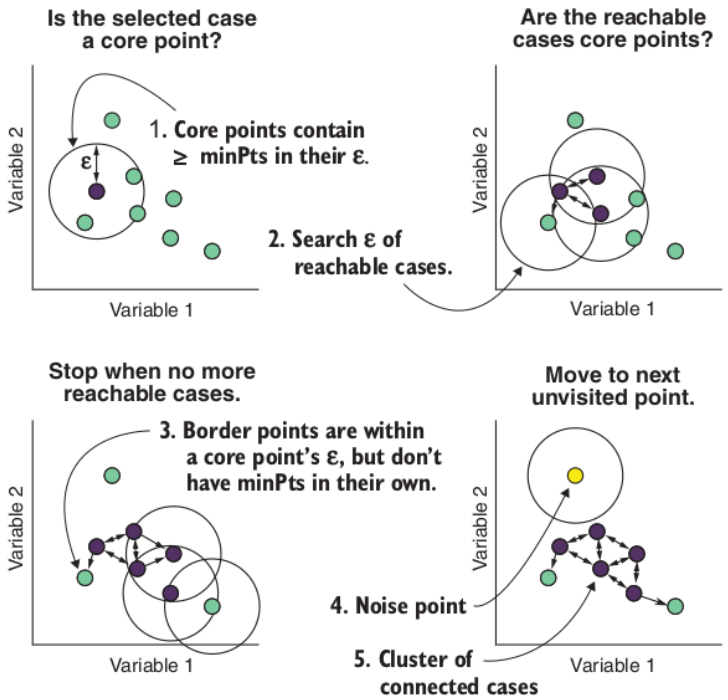


Figure: from https://scikit-learn.org/stable/modules/clustering.html

3. DBscan (Density Based Spatial Clustering of Applications with Noise)



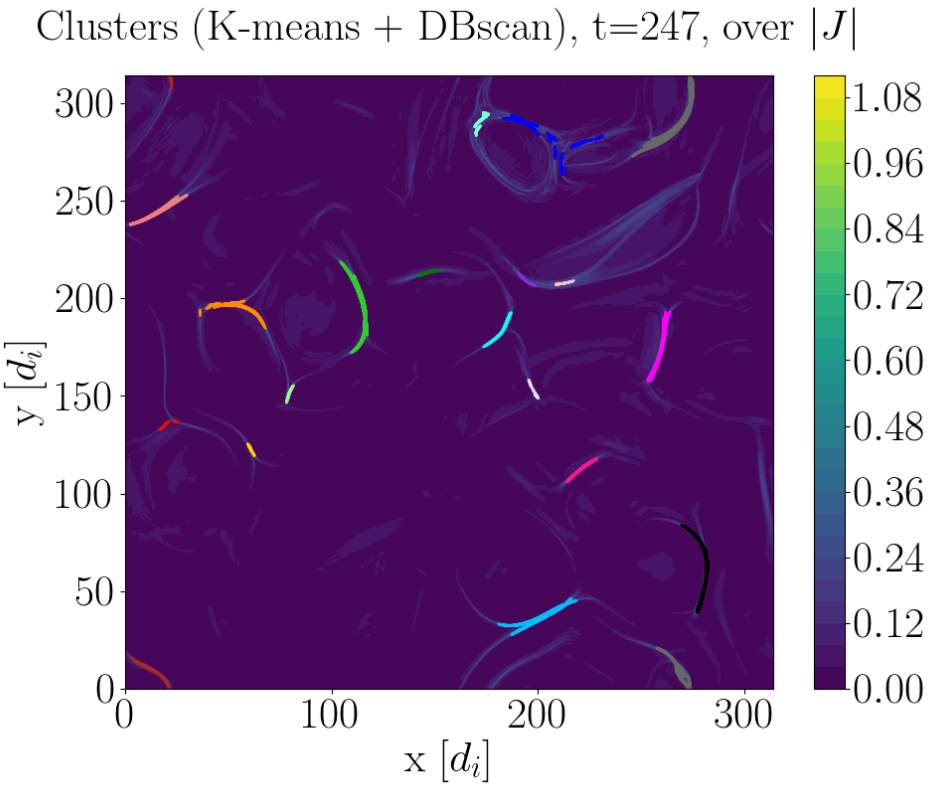
The DBSCAN algorithm. A case is selected at random, and if its *epsilon* radius (ε) contains *minPts* cases, it is considered a core point. Reachable cases of this core point are evaluated the same way until there are no more reachable cases. This network of density-connected cases is considered a cluster. Cases that are reachable from core points but are not themselves core points are border points. The algorithm moves on to the next unvisited case. Cases that are neither core nor border points are labeled as noise.

Parameters:

- ϵ: search radius
- minPts: minimum number of points that a cluster must have in order to be defined as a cluster.

Figure: from [Rhys, 2020]. In this example minPts $= 3$.

3. DBscan: our results



From [Sisti et al., 2021]

Our parameters for DBscan:

- $\epsilon = 50$ (in grid points unit, i.e. $\simeq 5 d_i$)
- minPts $= 100$

4. MR recognition: Aspect Ratio threshold

$$AR = \frac{length}{width}$$

- Sweet-Parker model: $\frac{\delta}{L} \sim \frac{v_{in}}{v_{out}} \sim \frac{v_{in}}{v_A} \sim S^{-1/2} \sim R$ where 2δ is the width, $2L$ is the length, S is the Lundquist number, which gives the ratio between the diffusive time and the Alfvèn one, and R is the reconnection rate $\rightarrow$ too slow
- Fast reconnection: $R \simeq 0.1$
  - Hall term included in Ohm's law and decoupling of the electron dynamics because of inertial term
  - $\rightarrow AR \sim 10$
- Karimabadi stated: "An interesting idea that yields an estimate for the aspect ratio of the diffusion region is based on linear theory mode. Since tearing mode is the eigenfunction of a current sheet, one may suppose that a sufficient condition for reconnection is for the tearing to be unstable. The wavenumber for the most unstable tearing mode [Karimabadi et al., 2005] is given by $k\delta = \frac{2\pi\delta}{L} \sim 0.5$ or $\frac{\delta}{L} \sim 0.08$, where δ and L are the width and length of the diffusion region. This is in reasonable agreement with the rate of ~ 0.1 which is often observed in various reconnection regimes." $\rightarrow AR \sim 12.5$

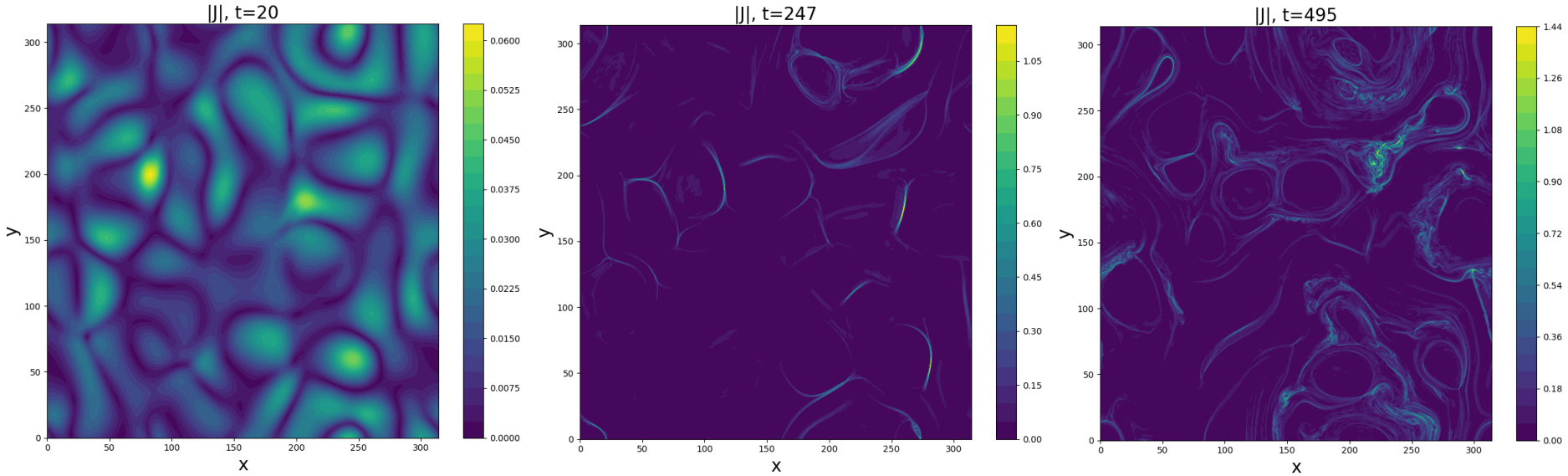
4. MR recognition: Aspect Ratio threshold

$$AR = \frac{length}{width}$$

- In the local peak of |J| for each cluster we compute the Hessian matrix, with its eigenvalues and eigevectors.
- length and width extimation using the same method of [Califano et al., 2020]:
  - width: we look at the interpolated profile of J along the direction of strongest variation (given by Hmatrix) $\rightarrow$ full width at half-maximum of |J|.
  - length: the maximal distance between two points belonging to the same structure

Algorithm with unsupervised ML: our results

- Comparison of the results for five times: $t \sim 20[1/\Omega_{ci}]$ (beginning of the simulation, no reconnections, PHASE I), $t \sim 230[1/\Omega_{ci}]$, $t \sim 247[1/\Omega_{ci}]$, $t \sim 282[1/\Omega_{ci}]$ (three central times, with current sheets well developed, PHASE II), $t \sim 494[1/\Omega_{ci}]$ (fully developed turbulence, PHASE III).



- Precision $= \dfrac{\text{number of reconnection sites among sites selected}}{\text{number of sites selected}}$
- nMR-precision $= \dfrac{\text{number of non-reconnections among sites excluded}}{\text{number of sites excluded}}$

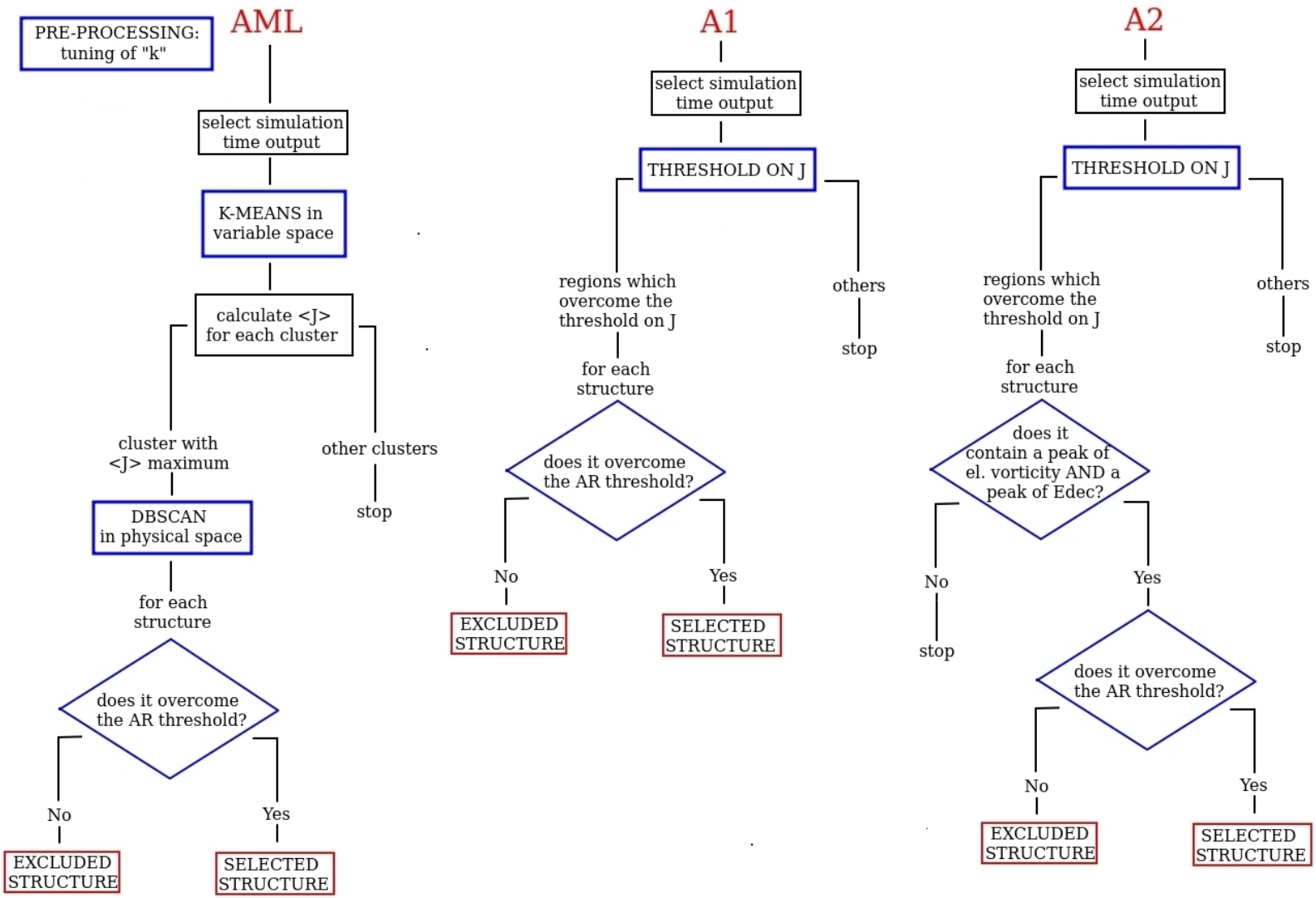
Algorithm with unsupervised ML: our results

Method: kmeans + DBscan + threshold over AR

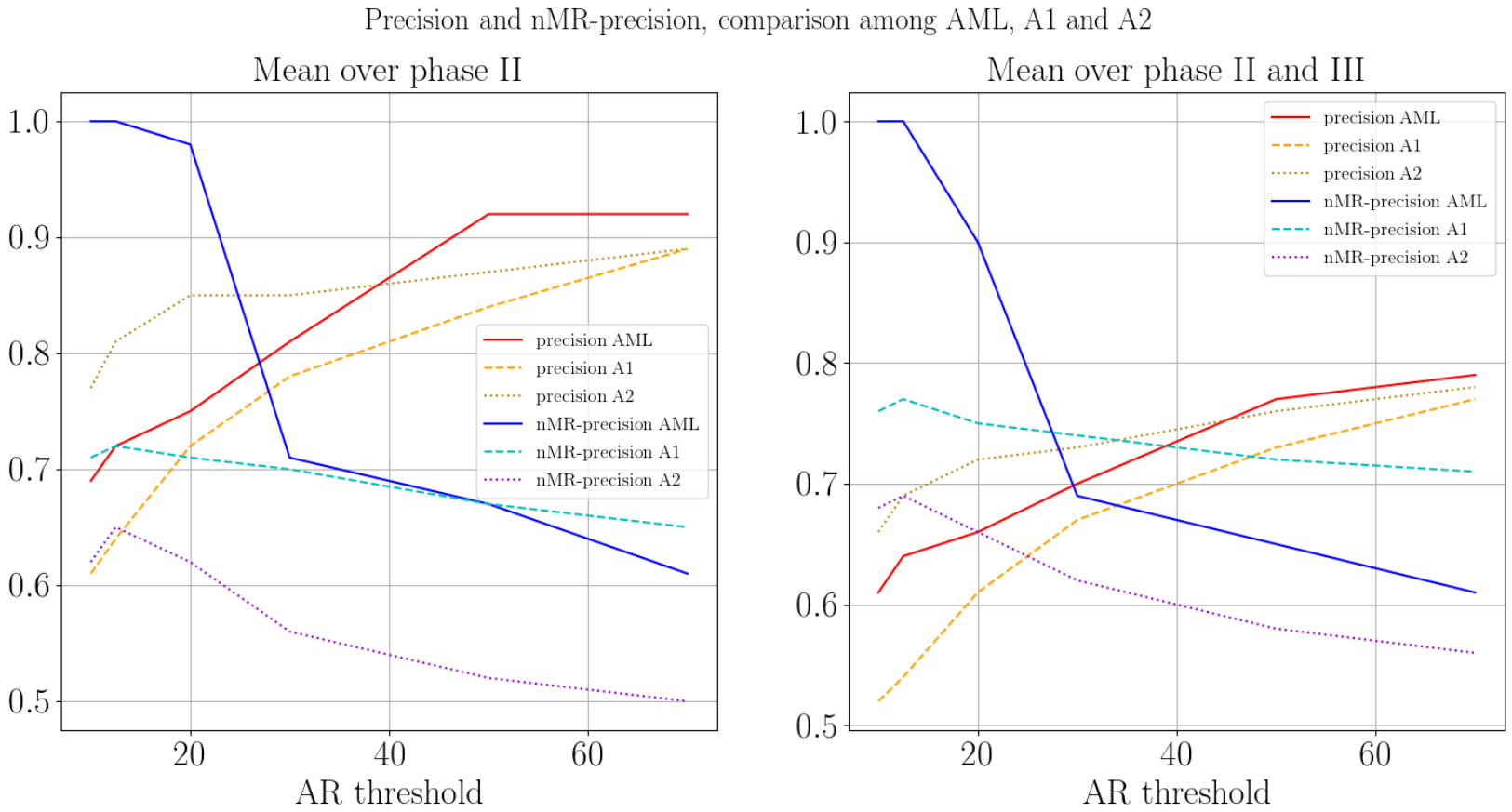
Tempo $[1/\Omega_{ci}]$		20	230	247	282	494	Mean (3t)	Mean (4t)
N. structures		35	29	19	24	32		
N. structures	$AR > 10$	0	20	17	22	30		
	$AR > 12.5$	0	18	17	21	30		
	$AR > 20$	0	15	17	20	23		
	$AR > 30$	0	14	13	14	18		
	$AR > 50$	0	8	10	13	9		
	$AR > 70$	0	6	9	12	7		
precision	$AR > 10$	–	0.6	0.82	0.64	0.4	0.69	0.61
	$AR > 12.5$	–	0.67	0.82	0.67	0.4	0.72	0.64
	$AR > 20$	–	0.73	0.82	0.7	0.39	0.75	0.66
	$AR > 30$	–	0.79	0.85	0.79	0.39	0.81	0.70
	$AR > 50$	–	1	1	0.77	0.33	0.92	0.77
	$AR > 70$	–	1	1	0.75	0.43	0.92	0.79
nMR-precision	$AR < 10$	1	1	1	1	1	1	1
	$AR < 12.5$	1	1	1	1	1	1	1
	$AR < 20$	1	0.93	1	1	0.67	0.98	0.9
	$AR < 30$	1	0.93	0.5	0.7	0.64	0.71	0.69
	$AR < 50$	1	0.81	0.55	0.64	0.61	0.67	0.65
	$AR < 70$	1	0.74	0.5	0.58	0.64	0.61	0.61

Table: Number of structures found, precision and nMR-precision for AML, for different AR threshold (from 10 to 70). The results are shown for five different times.

Algorithm with unsupervised ML: comparison with two simplified methods A1 and A2



Algorithm with unsupervised ML: our results



Conclusions

- development and test of a method to automatically find magnetic reconnection in 2D simulations of plasma turbulence. The AML method uses unsupervised machine learning and a threshold on the aspect ratio of the structures
- AML method performs better than simplified A1 and A2 methods that don't use machine learning but only thresholds on important physical quantities
- we found optimal aspect ratio threshold for which precision and nMR-precision $\sim 80\%$
- aspect ratio turns out to be an important parameter to select truly reconnecting current sheets

References I


Califano, F., Cerri, S. S., Faganello, M., Laveder, D., Sisti, M., and Kunz, M. W. (2020).
Electron-only reconnection in plasma turbulence.
Frontiers in Physics, 8:317.


Hu, A., Sisti, M., Finelli, F., Califano, F., Dargent, J., Faganello, M., Camporeale, E., and Teunissen, J. (2020).
Identifying magnetic reconnection in 2d hybrid vlasov maxwell simulations with convolutional neural networks.
The Astrophysical Journal, 900(1):86.


Karimabadi, H., Daughton, W., and Quest, K. (2005).
Physics of saturation of collisionless tearing mode as a function of guide field.
Journal of Geophysical Research, 110.


Rhys, H. (2020).
Machine Learning with R, the tidyverse, and mlr.
Manning Publications.

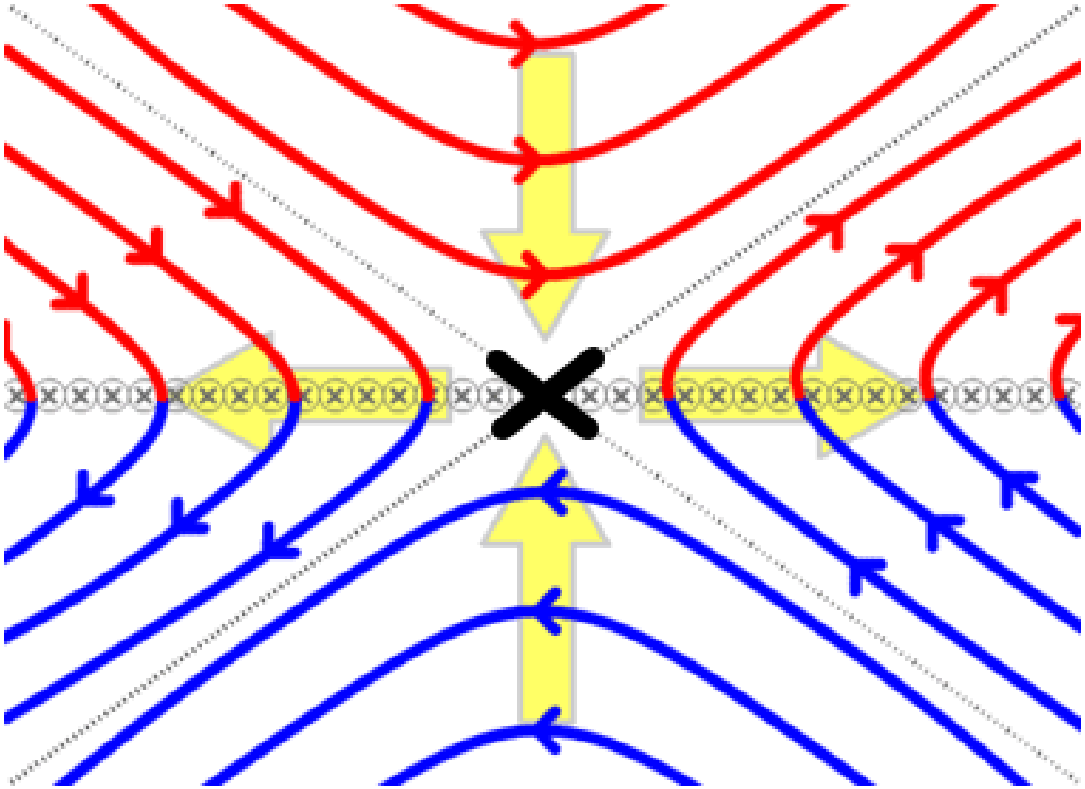
References II

Sisti, M., Finelli, F., Pedrazzi, G., Faganello, M., Califano, F., and Ponti, F. D. (2021).
Detecting reconnection events in kinetic vlasov hybrid simulations using clustering techniques.
The Astrophysical Journal, 908(1):107.

Valentini, F., Califano, F., Hellinger, P., and Mangeney, A. (2007).
A hybrid-vlasov model based on the current advance method for the simulation of collisionless magnetized plasma.
Journal of Computational Physics, 225:753–770.

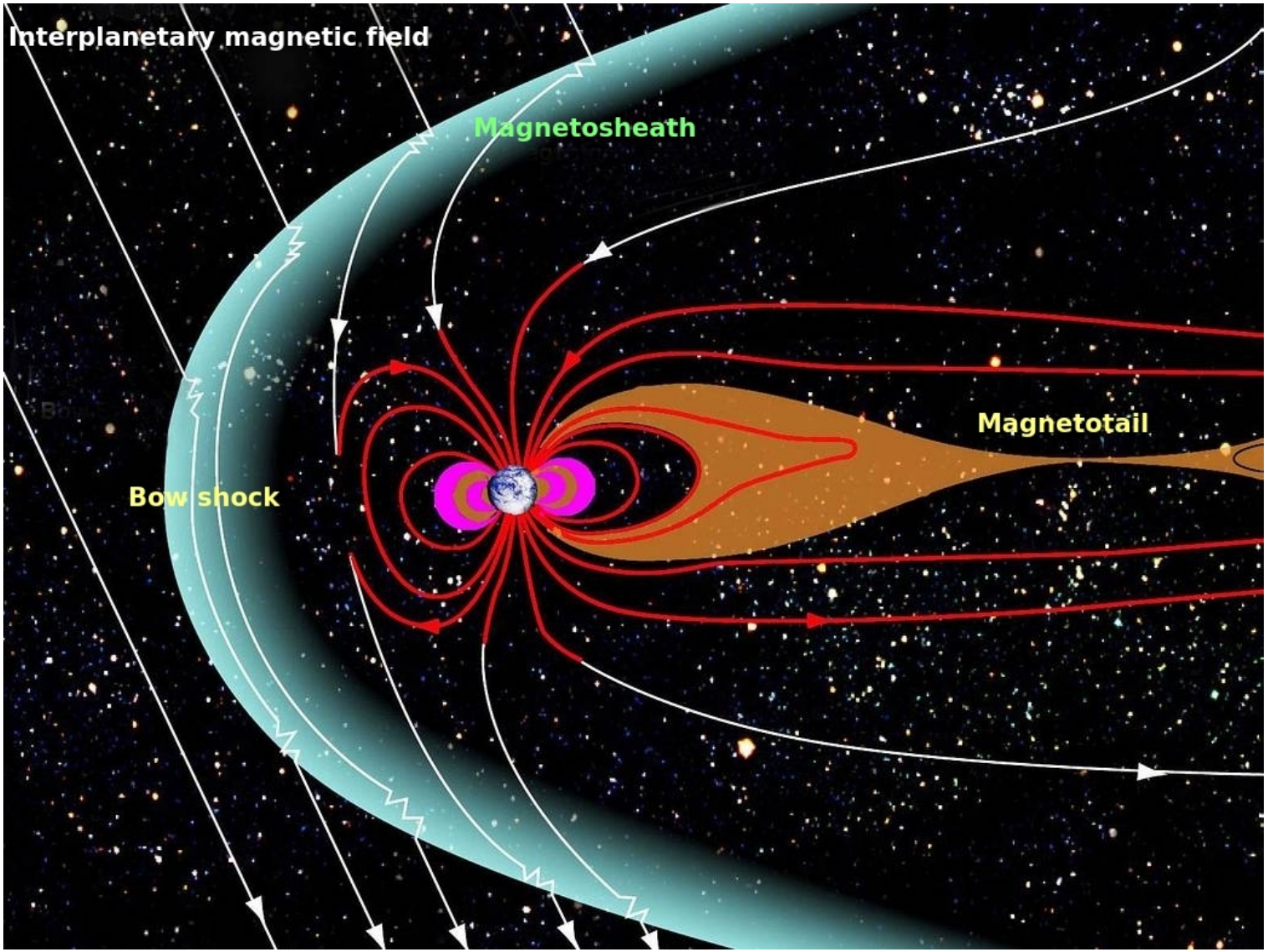
Zweibel, E. G. and Yamada, M. (2009).
Magnetic reconnection in astrophysical and laboratory plasmas.
Annual Review of Astronomy and Astrophysics, 47(1):291–332.

Backup slides: magnetic reconnection



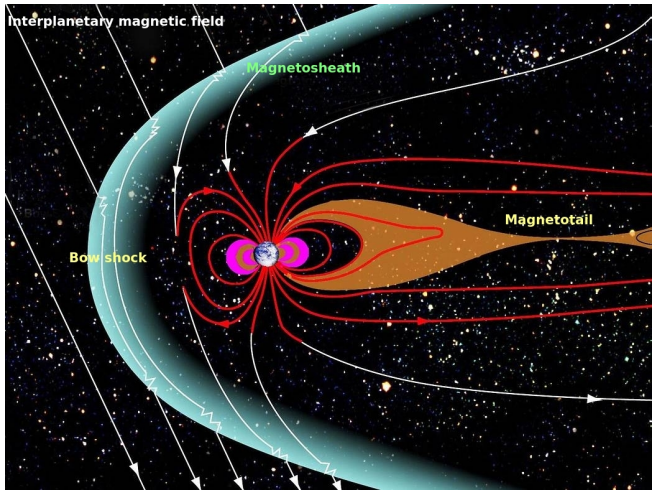
Breaking of conservation theorems ← non-ideal terms into Ohm's law

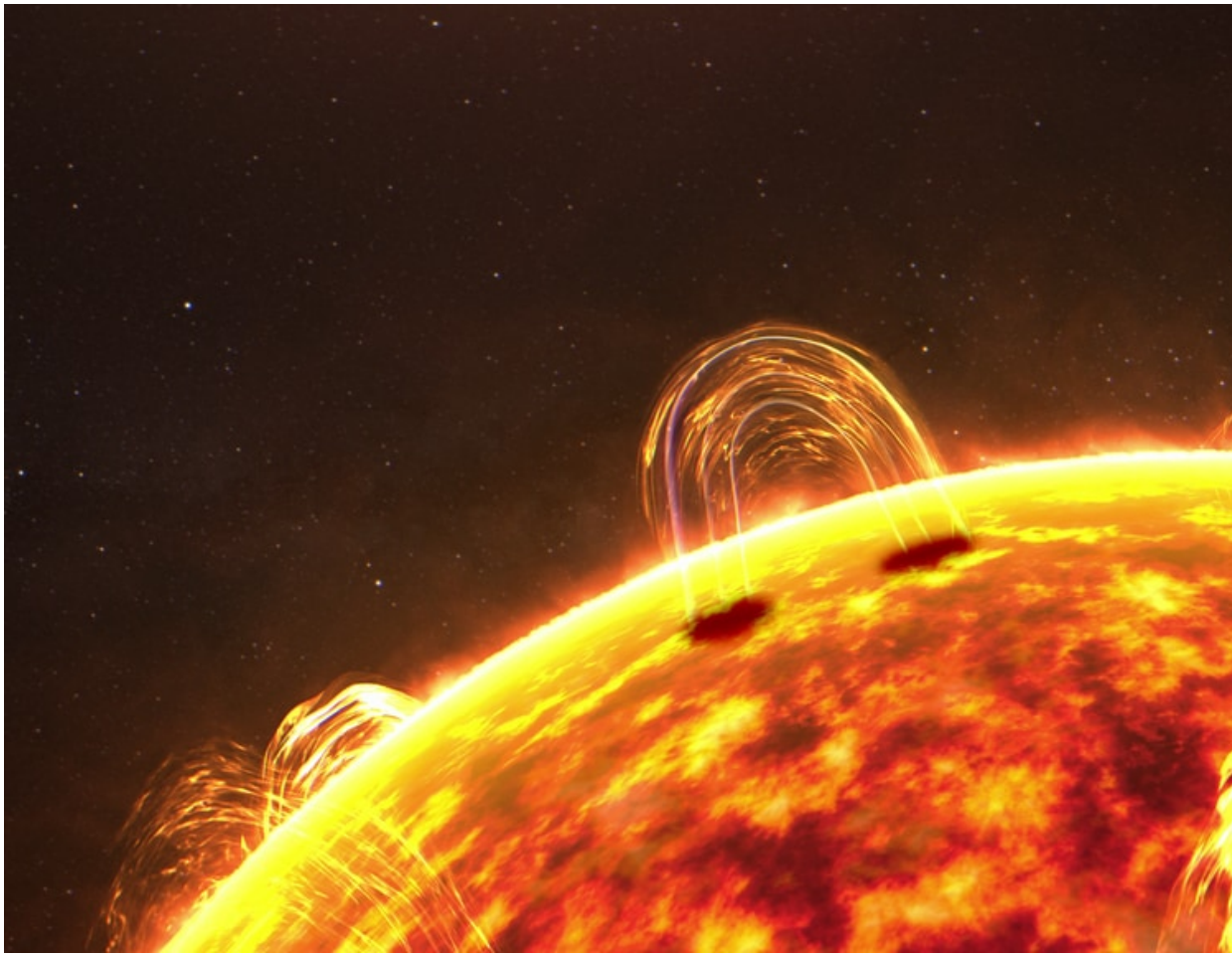
Backup slides: magnetic reconnection



credits: https://www.nasa.gov/mission_pages/sunearth/multimedia/magnetosphere2-unlabeled.html

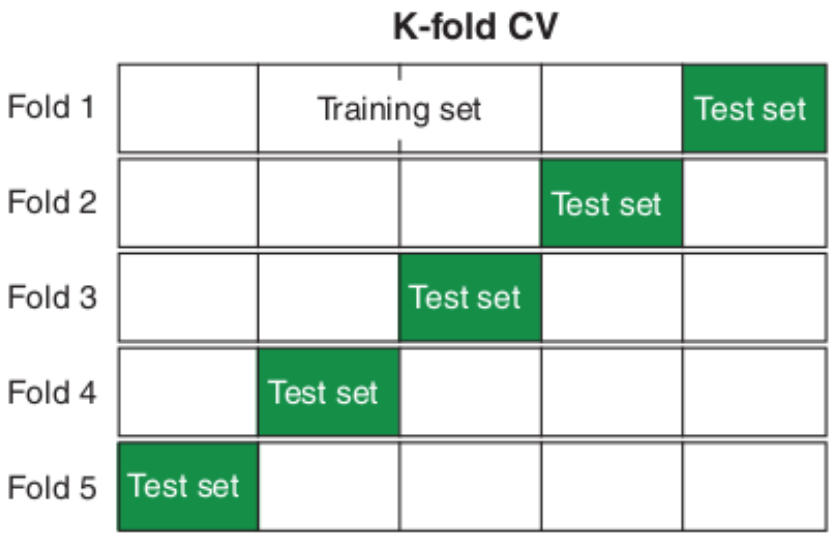
Backup slides: magnetic reconnection





credits: https://www.nasa.gov/mission_pages/sunearth/multimedia/magnetosphere2-unlabeled.html
https://svs.gsfc.nasa.gov/20310

Backup slides: crossivalidation



1. The data is randomly split into *k* equal-sized folds.
2. Each fold is used as the test set once, where the rest of the data makes the training set.
3. For each fold, predictions are made on the test set.
4. The predictions are compared to the true values.

K-fold CV. The data is randomly split into near equally sized folds. Each fold is used as the test set once, with the rest of the data used as the training set. The similarity of the predictions to the true values of the test set is used to evaluate model performance.

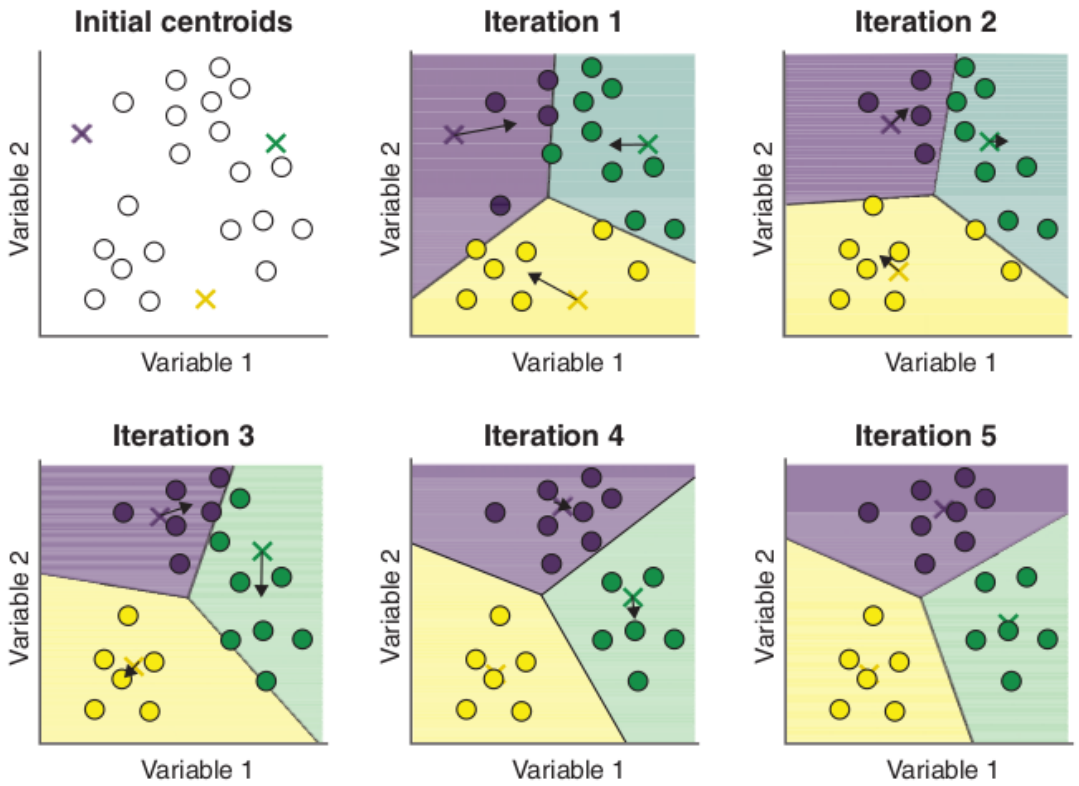
Figure: from [Rhys, 2020]

Backup slides: the Devies-Bouldin index for k tuning

It calculates the ratio of the within-cluster variance ("scatter") to the separation between clusters centroids.

- $Scatter_k = (\frac{1}{n_k} \sum_{i \in k}^{n_k} (x_i - c_k)^2)^{1/2}$
- $Separation_{j,l} = ||c_j - c_l||_{1/2}$
- $ratio_{j,k} = \frac{Scatter_k + Scatter_j}{Separation_{j,k}}$
- the ratio is calculated for all pairs of clusters, and for each cluster
- R_k is the largest ratio between a cluster and the other clusters
- $DB = \frac{1}{N} \sum_{k=1}^{N} R_k$

Backup slides: K-means (Lloyd's algorithm)

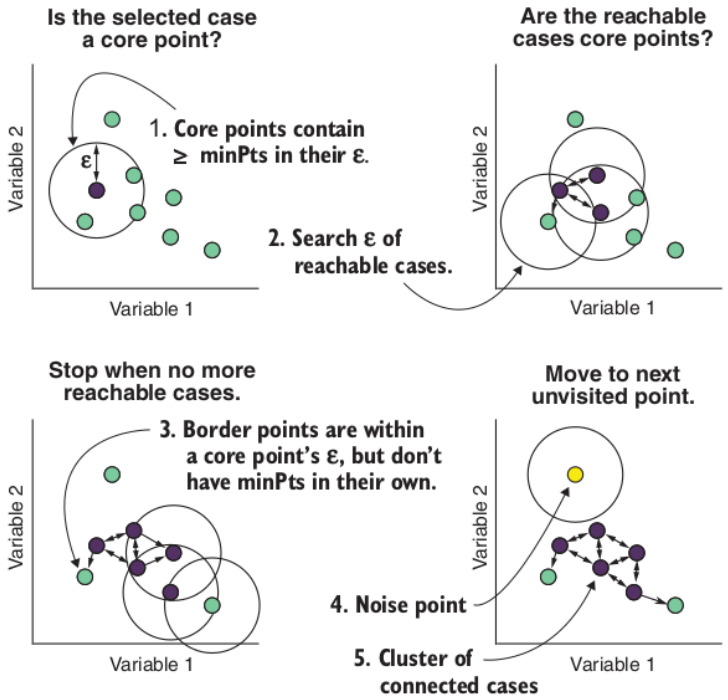


Five iterations of k-means clustering. In the top-left plot, three initial centers are randomly generated in the feature space (crosses). Cases are assigned to the cluster of their nearest center. At each iteration, each center moves to the mean of the cases in its cluster (indicated by arrows).

Figure: from [Rhys, 2020]

1. It randomly initializes k-centroid.
2. It calculates the Euclidean distance between each case and each centroid. A case it's assigned to the cluster represented by its nearest centroid.
3. The centroids are moved: they are placed at the mean of the cases of each cluster.
4. The process repeats until no case change cluster from one iteration to another.

Backup slides: DBscan

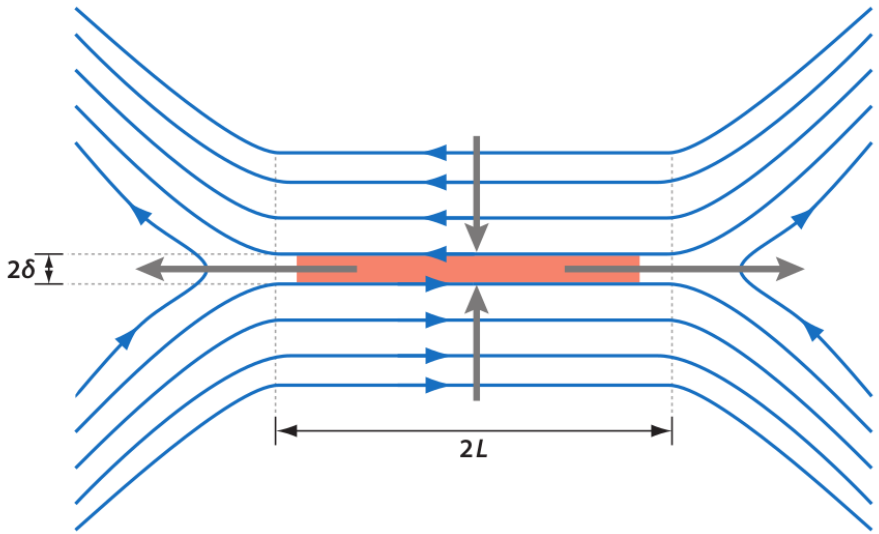


The DBSCAN algorithm. A case is selected at random, and if its *epsilon* radius (ε) contains *minPts* cases, it is considered a core point. Reachable cases of this core point are evaluated the same way until there are no more reachable cases. This network of density-connected cases is considered a cluster. Cases that are reachable from core points but are not themselves core points are border points. The algorithm moves on to the next unvisited case. Cases that are neither core nor border points are labeled as noise.

Figure: from [Rhys, 2020]

1. It selects a case randomly from the data set.
2. It searches for other cases in a n-dimensional sphere width radius ϵ.
3. If this case contains at least minPts cases inside its search radius it is marked as a core point, else it moves on.
4. Each case within the search radius is checked to see if it is a core point.
5. The process repeats.

Backup slides: Sweet-Parker model



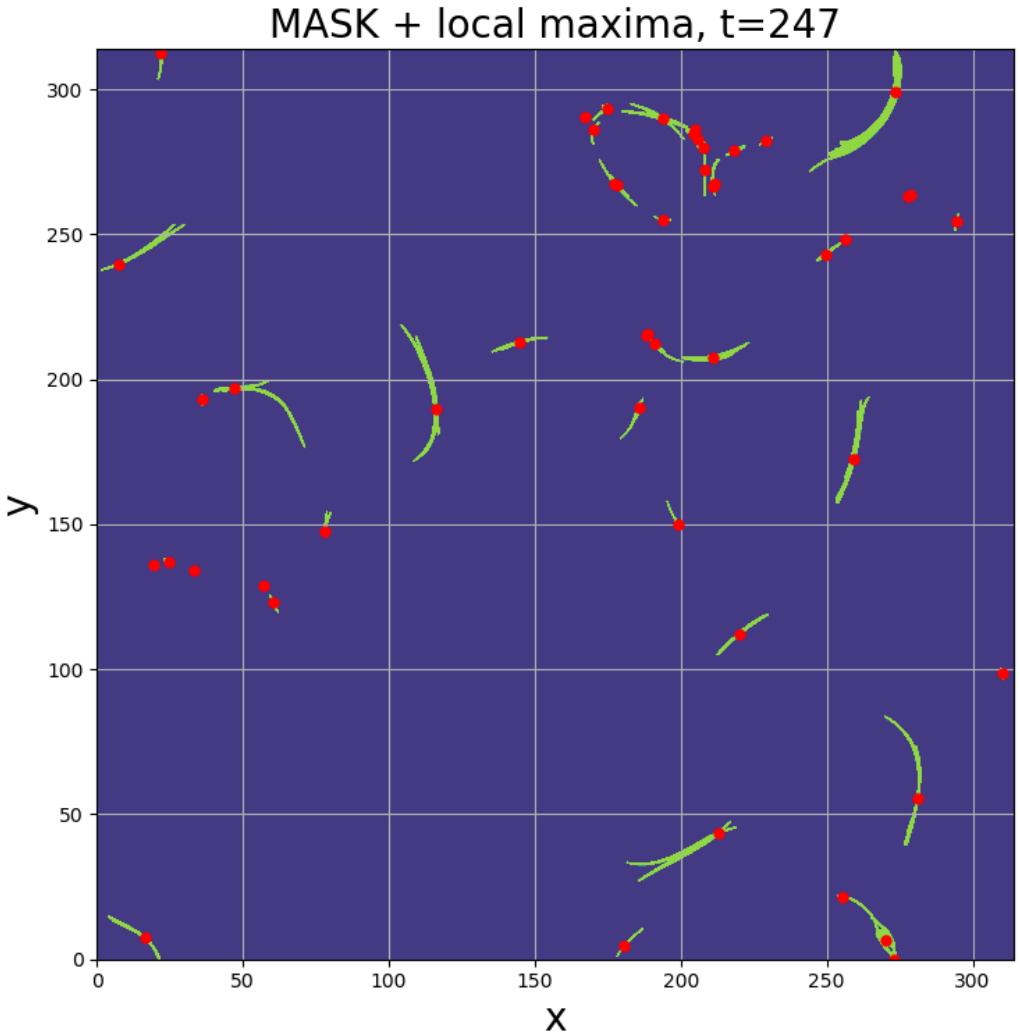
Sketch of magnetic field geometry in Sweet-Parker reconnection. Oppositely directed magnetic fields are brought together over a length $2L$ and reconnect in a diffusion layer of width 2δ.

Figure: from [Zweibel and Yamada, 2009]

Sweet-Parker model:

- The outflow speed is the Alfvèn speed
- Mass conservation: $v_{in} L = V_A \delta$
- The electric field is given by the resistive Ohm's law

Backup slides: simplified method, threshold over J



Backup slides: algorithm with unsupervised ML, a comparison with a simplified method (2)

Method (2): multi-threshold (quantities: $|J|$, $|\Omega_e|$, electron decoupling) + DBscan (minPts = 2)

Tempo $[\Omega_{ci}]$	20	230	247	282	494	over 3t	over 5t
N. clusters ($\epsilon = 10$)	2	31	30	19			
N. mr found ($\epsilon = 10$)	0	14	17	14		45	
precision ($\epsilon = 10$)	0	0.45	0.57	0.74		0.59	
N. clusters ($\epsilon = 30$)	2	25	24	13			
N. mr found ($\epsilon = 30$)	0	15	17	13		45	
precision ($\epsilon = 30$)	0	0.6	0.71	0.85		0.72	
N. clusters ($\epsilon = 50$)	2	23	21	12			
N. mr found ($\epsilon = 50$)	0	15	16	10		41	
precision ($\epsilon = 50$)	0	0.65	0.76	0.83		0.75	
N. clusters ($\epsilon = 70$)	2	20	21	12			
N. mr found ($\epsilon = 70$)	0	14	16	10		40	
precision ($\epsilon = 70$)	0	0.7	0.76	0.83		0.76	

Problems:

- We cannot apply threshold over AR, thus it finds sites also for times at the beginning of the simulation
- We cannot increase ϵ too much: lots of sites would be merged and we would loose good sites

Backup slides: satellite measurements, trajectory

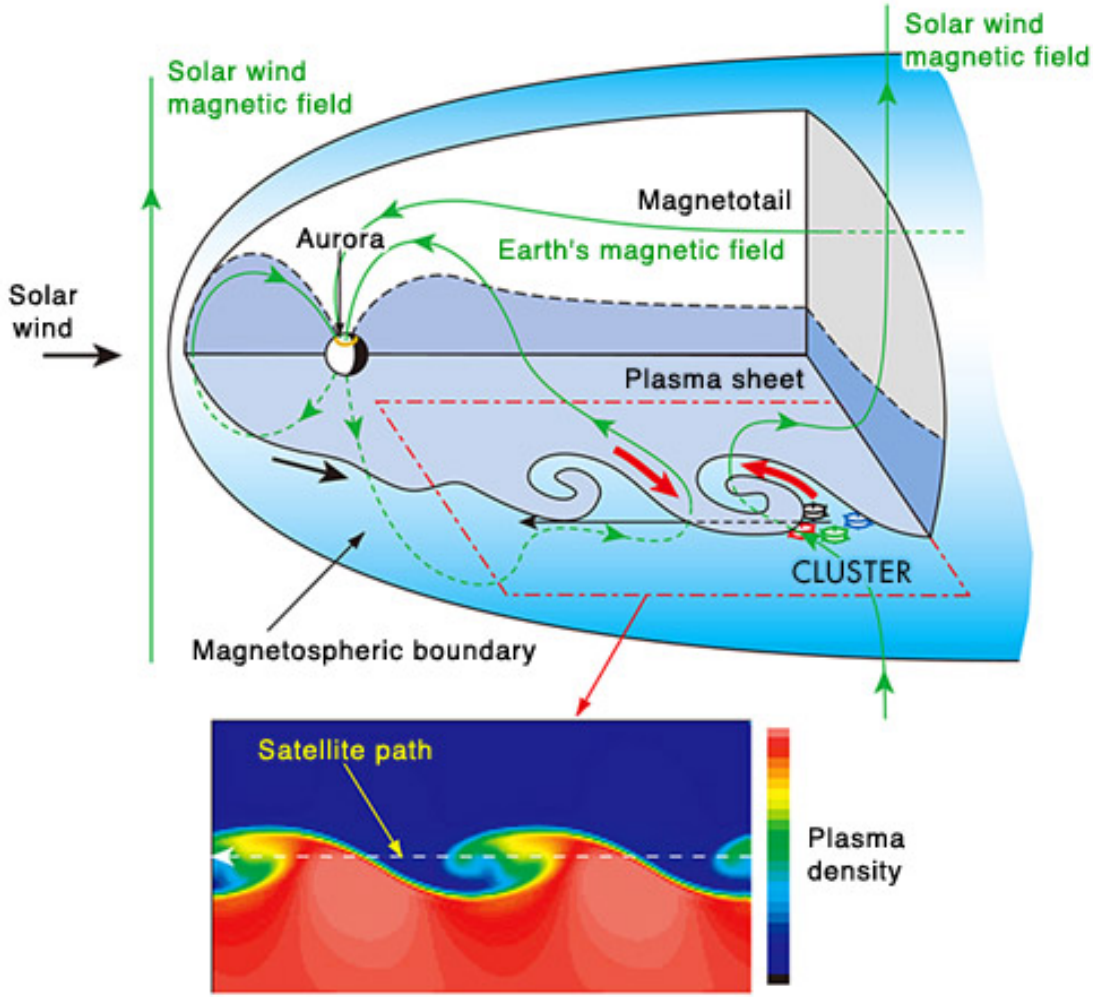
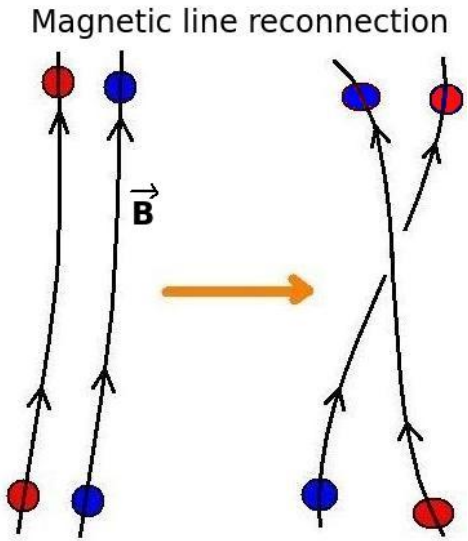


Figure: Hasegawa&al.,Nat.,2004

Backup slides: magnetic reconnection instability



Term which break the conservation theorems if added to the Ohm's law:

- Electron inertia: $\frac{m_e}{ne^2}\frac{d\vec{J}}{dt}$
- **Resistivity term:** $\eta\vec{J}$

Conservation theorems:

- *Alfvén theorem*: $d_t \int_{S(t)} \vec{B} \cdot d\vec{S} = 0$
- *Connection theorem*: if, for $t = 0$, $d\vec{l} \wedge \vec{B} = 0$, then $d_t(d\vec{l} \wedge \vec{B}) = 0, \quad \forall t$

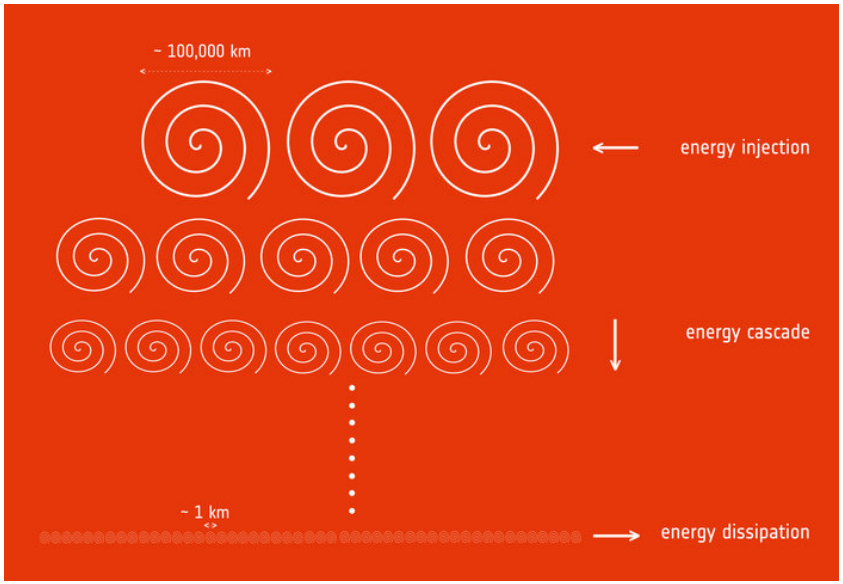
Hyphotesis:

Ideal Ohm's law: $\vec{E} + \frac{1}{c}\vec{u} \wedge \vec{B} = 0$

or

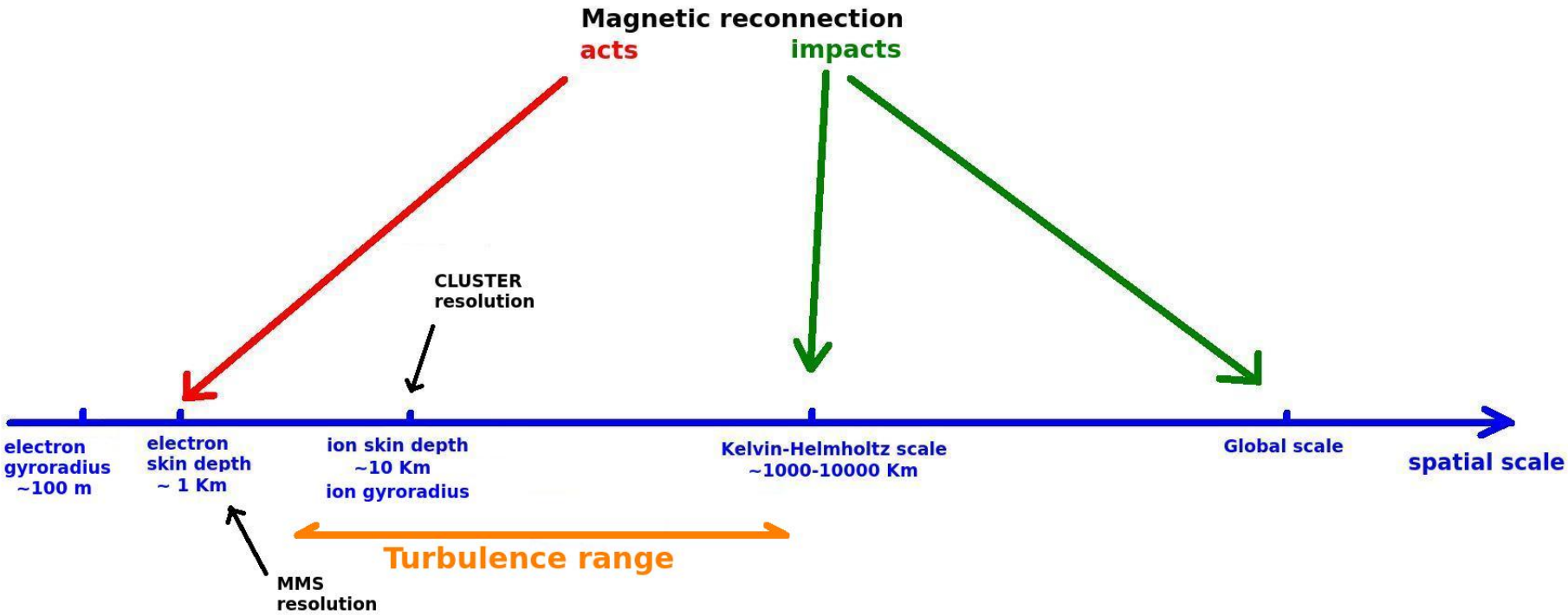
$$\vec{E} + \frac{1}{c}\vec{u} \wedge \vec{B} = \frac{1}{nec}\vec{J} \wedge \vec{B} - \frac{1}{ne}\nabla P_e$$

Backup slides: brief introduction to the topic: turbulence



- Reconnection (as an intermittency effect) can be a dissipation mechanism

Backup slides: scales



Backup slides: Hybrid Vlasov-Maxwell code

Normalized equations used:

- Vlasov equation for the ion distribution function:

$$\frac{\partial f}{\partial t} + \mathsf{v} \cdot \nabla f + (\mathsf{E} + \mathsf{v} \wedge \mathsf{B}) \cdot \frac{\partial f}{\partial \mathsf{v}} = 0 \tag{4}$$

- The Ohm's law for the electric field:

$$\mathsf{E} - d_e^2 \nabla^2 \mathsf{E} = -(\mathsf{u} \wedge \mathsf{B}) + \frac{1}{n}(\mathsf{J} \wedge \mathsf{B}) + \frac{1}{n} d_e^2 \nabla \cdot \Pi - \frac{1}{n} \nabla P_e + \frac{d_e^2}{n} \nabla \cdot [\mathsf{uJ} + \mathsf{Ju}] - \frac{1}{n} d_e^2 \nabla \cdot (\frac{\mathsf{JJ}}{n}) \tag{5}$$

  the ion density n, the ion bulk velocity u and the ion pressure tensor Π are obtained as the moments of the ion distribution function f.

- Maxwell's equations

Backup slides: proxies to identify reconnecting current sheets

The coupling between an electrically conducting fluid and the magnetic field immersed in the fluid is described by the generalized Ohm's law

$$\mathsf{E} + \frac{\mathsf{v}}{c} \wedge \mathsf{B} = \mathsf{R} \tag{6}$$

where R comprises the different so-called nonideal effects of the plasma, dissipative or nondissipative.
Faraday's law reads:

$$\frac{\partial \mathsf{B}}{\partial t} = -\nabla \wedge \mathsf{E} \tag{7}$$

Only $E_{\parallel} = R_{\parallel}$ can cause reconnection, since the perpendicular component $\mathsf{R}_{\perp}$ can always be incorporated in the general velocity u by rewriting $\mathsf{R}_{\perp} = \mathsf{u}' \wedge \mathsf{B}$, where $\mathsf{u} = \mathsf{v} - \mathsf{u}'$.

Backup slides: proxies to identify reconnecting current sheets

- Poynting equation (energy conservation for the EM-fields)

  $\frac{\partial}{\partial t}\frac{B^2}{8\pi} + \nabla \cdot (\frac{c}{4\pi}\mathsf{E} \wedge \mathsf{B})) = -\mathsf{E} \cdot \mathsf{J}$

- Conservation equation for the total energy (no viscosity, no diffusion, no heat flux):

  $\frac{\partial}{\partial t}(u + \frac{\rho}{2}v^2) = -\nabla \cdot [(u + p + \frac{\rho}{2}v^2)\mathsf{v}] + \mathsf{J} \cdot \mathsf{E}$

- $\rightarrow \frac{\partial}{\partial t}(\frac{B^2}{8\pi} + u + \frac{\rho}{2}v) = -\nabla \cdot [\frac{c}{4\pi}\mathsf{E} \wedge \mathsf{B} + (u + p)\mathsf{v} + \frac{\rho}{2}v^2\mathsf{v}]$
  with u internal energy.

Thus $\mathsf{J} \cdot \mathsf{E}$ gives the energy conversion.