Segmentation of turbulent computational fluid dynamics simulations with unsupervised ensemble learning

Maarja Bussova,b,∗, Joonas Nättiläc,d

aTartu Observatory, University of Tartu, Observatooriumi 1, 61602-zip, Tõravere, Estonia
bDepartment of Physics, University of Helsinki, P.O. Box 64, FI-00014, Helsinki, Finland
cPhysics Department and Columbia Astrophysics Laboratory, Columbia University, 538 West 120th Street, New York, NY 10027, USA
dCenter for Computational Astrophysics, Flatiron Institute, 162 Fifth Avenue, New York, NY 10010, USA

Abstract
Computer vision and machine learning tools offer an exciting new way for automatically analyzing and categorizing information from complex computer simulations. Here we design an ensemble machine learning framework that can independently and robustly categorize and dissect simulation data output contents of turbulent flow patterns into distinct structure catalogues. The segmentation is performed using an unsupervised clustering algorithm, which segments physical structures by grouping together similar pixels in simulation images. The accuracy and robustness of the resulting segment region boundaries are enhanced by combining information from multiple simultaneously-evaluated clustering operations. The stacking of object segmentation evaluations is performed using image mask combination operations. This statistically-combined ensemble (SCE) of different cluster masks allows us to construct cluster reliability metrics for each pixel and for the associated segments without any prior user input. By comparing the similarity of different cluster occurrences in the ensemble, we can also assess the optimal number of clusters needed to describe the data. Furthermore, by relying on ensemble-averaged spatial segment region boundaries, the SCE method enables reconstruction of more accurate and robust region of interest (ROI) boundaries for the different image data clusters. We apply the SCE algorithm to 2-dimensional simulation data snapshots of magnetically-dominated fully-kinetic turbulent plasma flows where accurate ROI boundaries are needed for geometrical measurements of intermittent flow structures known as current sheets.

Keywords: plasma — turbulence — clustering analysis — image segmentation — unsupervised machine learning

1. Introduction
Turbulent, seemingly chaotic flows, are known to create self-similar structures and flow patterns on multiple scales. Especially interesting are any discrete long-lived structures that appear close to the dissipation scales, originating from the intermittency of the turbulence. In magnetohydrodynamical (MHD) turbulence these intermittent structures manifest themselves as the so-called current sheets that are regions of intense electric current flowing in thin two-dimensional sheet-like configurations [1]. Analyzing the geometrical shapes and sizes of these structures can help us better quantify the role of intermittency in turbulent fluids and plasmas.

In the past, different statistical methods for automating the detection of such structures have been conceived but the algorithms are known to be computationally very demanding [e.g.,2,3,4]. Machine learning methods and computer vision algorithms offer a new promising avenue for constructing such detection frameworks since they are fast to evaluate and highly optimized [e.g.,5,6,7]. Here, we present an ensemble unsupervised machine learning algorithm for image structure detection and automated segmentation that is specifically tailored for structure detection from computer simulation outputs.

In general, computer vision algorithms and the related machine learning tools offer an interesting new way of studying physical systems. These algorithms enable an easy-to-use and efficient automation of visual segmentation of computer simulation outputs. Image segmentation is an actively studied problem in the field of computer vision where the aim is to label each pixel in an image into a distinct category. Typically, supervised learning models are used to perform the pixel category assignment — in this case, large training sets containing labels for each pixel needs to be compiled a priori. The state-of-the-art models for real-world images are prominently deep learning algorithms [8,9,10] and they are rapidly evolving and improving [e.g.,11,12,13]. However, the large computa-
tional cost and requirement for a vast pixel-level labeled pool of data make some of the best performing supervised models not suitable for practical analysis. Additionally, obtaining a desired accuracy for deep learning models is known to be notoriously hard \[14, 15\].

A major class of machine learning methods tackling data with no ground truth is the unsupervised learning method. Unsupervised learning algorithms are prominently used for cluster detection, dimensionality reduction of high dimensional data, or aiming at representing the data with a few prototypes. Unsupervised algorithms, which can be applied to data with no knowledge of the ground truth or even of the amount of possible clusters, offer an easy-to-use alternative for segmentation of images and videos \[16, 17\].

A major drawback of unsupervised machine learning methods is that the resulting image segments and their associated clusters tend to be varying from one evaluation of the algorithm to another. This can be highly unwanted especially in applications where robust cluster classification, segmentation, and pixel-level image dissection are needed. The validity of unsupervised clustering algorithms depends on the algorithm and prior knowledge about the data; this estimation is usually done with external or internal metrics \[18, 19\].

A common cure for increasing the "signal strength" of individual analysis results is to combine multiple evaluations together, hence increasing the signal-to-noise ratio. These algorithms are typically known as ensemble machine learning methods. Ensemble frameworks have been developed to increase the robustness and stability of unsupervised clustering algorithms in many previous studies \[20, 21, 22\]. Graph partitioning models to create cluster ensembles show promising results in \[23, 24\]. Multi-modal hypergraph methods are developed, for example, in \[25, 26\]. In \[27\] an unsupervised ensemble framework is developed for Self-organizing Map clustering results, similarly to this work.

In this paper we construct a new ensemble framework that combines independent realizations of multiple clustering algorithm results, with the aim to provide reliable and robust segmentation regions obtained via unsupervised learning methods. The ensemble framework is devised to increase stability and robustness of the clusters detected by an unsupervised learning algorithm from image data. We use the Self-Organizing Map (SOM) algorithm [also known as Kohonen’s map; \[28, 29\] ] as our base clustering algorithm, but labels obtained from other clustering algorithms may also be used. We show that the resulting region of interest (ROI) of the objects from the combined ensemble algorithm are more easily interpretable and geometrically more stable against fluctuations. This makes the proposed method ideal for automating the segmentation procedure of computer simulations. We, however, note that the new ensemble method may also be used for many other similar image segmentation tasks with no ground truth information, where high accuracy of the ROIs are needed like, for example, in medicine, remote sensing, or biology where images need to be dissected and segmented into different structures automatically \[30, 31\].

2. Data

Our main aim is to dissect the simulation data snapshot pixels into distinct physical structures based on the similarity of their feature vectors. We are, in particular, interested in automating the grouping of pixels into different physical categories whose geometrical size distributions we, in the end, want to measure.

We apply our image segmentation algorithms to fully-kinetic particle-in-cell simulations of decaying relativistic fully-kinetic turbulence [see e.g., \[32, 33, 34\]]. Here we omit most of the data interpretation and physics discussion and focus instead on the results and performance of the image segmentation algorithms. We refer the reader to [34] for a presentation and discussion of the various technical simulation parameters. We also note that no ground truth data exists in our case. The situation is common for many non-linear physics-motivated problems that have no a-priori known solution; instead, both the new algorithm and the system are studied simultaneously.

The analyzed input data is complex, sophisticated and the features are highly correlated. It consists of large 2-dimensional rectangular images where each pixel carries multiple features (i.e., a feature vector). Generalization to 3-dimensional data input is straightforward; here we focus on 2-dimensional case only to simplify the visualization of the results. We describe each pixel with a set of restricted invariant physical features obtained from the initial data because image analysis features need to be as invariant as possible \[30\].

One simulation image snapshot consists of roughly 1 million pixels with each pixel storing multiple physical quantities. We perform the clustering by using 8 consecutive snapshots from the simulations, with equal time steps between the samplings (corresponding to roughly one eddy turnover time in physical units). This increases the total number of data points to roughly 8 million pixels. For our segmentation analysis we select a 3-dimensional data vector $X_k = (B_{\perp}, J_{||}, [\vec{J} \cdot \vec{E}])$. These characteristics were chosen by relying on domain knowledge and on the fact that they capture most of the variability in the data.

Physically we expect that the perpendicular magnetic field averages out, $\langle \delta B_{\perp} \rangle = 0$, but has a finite root-mean-square value, $\sqrt{\langle \delta B_{\perp}^2 \rangle} / B_0 \approx 1$. The quantity is, to a first approximation, seen to follow a normal distribution. The parallel component of the current, $J_{||}$, should be small except for a few highly localized regions; its distribution

\[1\] The SOM is chosen for its topology preservation quality, ability of obtaining fine structures from image data, and conceptual simplicity. The SOM algorithm has been used and further developed in thousands of scientific research papers and used in many fields of science including medical sciences, financial sciences, and speech analysis [see e.g., \[29\].
is therefore expected to strongly deviate from a Gaussian normal distribution. In our analysis we normalize the value of $J_i$ to the theoretical maximum that a uniform charged particle background can support, yielding $\max(|J_i|) \approx 1$. The quantity is seen to strongly deviate from a normal distribution due to heavy tails. The third feature, $F - \vec{E}$, is a measure of an energy transport from the electromagnetic fields to the plasma; for a volumetric dissipation we would expect it to be a constant—in reality its value is highly variable, reminiscent of the intermittency of the turbulence. In our segmentation analysis we normalize its value with its root-mean-square value. The quantity deviates from a normal distribution due to over-pronounced tails. All of the described features have non-trivial mutual correlations.

Figure 1 visualizes the used simulation data. The visualizations shows five physical features processed from the raw simulation data. The upper left panel shows the whole simulation box consisting of a little over 1 million pixels. The depicted feature is the plasma number density $n$ (in units of initial number density $n_0$) at a physical time of $t \approx 5$ eddy turnover times. Rest of the images are zoom-in views to the simulation box. Left panel of the second row shows the current along the out-of-the-plane $z$-axis, $J_z$. Right panel of the second row shows the magnetic field strength perpendicular to the out-of-plane $z$-axis, $B_L$. The lower left panel visualize the work done by the parallel electric field ($J_x \cdot \vec{E}$) and lower right panel the plasma bulk Lorentz factor, $\Gamma$. All of these visualized features have multiple self-similar structures that are clearly visible in Fig. 1. For example, we identify circular islands that coincide with a high signal in $n/n_0$, $B_L$ and in $J_z$. Another prominent feature are the current sheets that correspond to a maximum in $J_z$ and $\Gamma$, and a minimum in $B_L$.

3. SOM algorithm

We start by presenting a short overview of the SOM algorithm and its process parameters, following Kohonen [29]. Kohonen and Mäkisara [35] and Kohonen [28]. Next, we discuss some of its shortcomings that motivated the development of the subsequent stacking framework.

3.1. Theory

The data sample is described by a vector $X = \{X_1, \ldots, X_k\}$, where each element $X_i$ describes a set of input variables $\xi_i$, such that $X_i = [\xi_1, \ldots, \xi_l] \in \mathbb{R}^l$, $\forall i \in \{1, \ldots, k\}$. We adopt a regular 2-dimensional rectangular shaped Kohonen neural map of dimensions $(m \times n)$, Kohonen [29] advises to select map dimensions such that they describe the lengths of the first principal components and for a bigger map size to detect fine structures. A rigorous choice for the dimensions and architecture of the map will result in a faster convergence.

Each neuron on the map is associated with a parametric vector $w_i = [\mu_1, \ldots, \mu_l] \in \mathbb{R}^l$. The initial values for the parametric vectors of nodes may be sampled randomly from the domain of the input vector parameter space. A more sophisticated nomination could also be used, if needed [29, 38].

Let $d(X_s, w_i)$ denote the distance metric used to determine the Best Matching Unit, $w_{BMU}$ (BMU) from the 2-dimensional neuron map, which is the neuron most similar to the sampled data vector $X_s$,

$$w_{BMU} = \argmin_i \{d(X_s, w_i)\}. \quad (1)$$

In this work we use the Euclidean distance on normalized data vectors $X_s$ and neuron models $w_i$, but other measures can also be used. The data vector $X_s$ is compared to all the neurons $w_i$. The neuron with the smallest Euclidean distance is chosen as the BMU.

A neighborhood $N_c$ consists of all the nodes up to a certain distance $r$ from a node $w_c$ on the neuron map, according to some geometric distance. The set usually shrinks with time and is determined by the neighborhood function $h_{ci}(t)$. The choice of $N_c$ influences the map’s ability to order and learn the underlying data distribution [28, 37]. In the SOM algorithm learning step, both the BMU and its spatial neighbors $N_{BMU}$ learn from the input vector.

The learning step to modify the neuron weight vectors is given as

$$w_i(t+1) = \begin{cases} w_i(t) + \alpha(t) \cdot h_{ci}(t)[X_s(t) - w_i(t)], & w_i \in N_c, \\ w_i(t), & w_i \notin N_c, \end{cases} \quad (2)$$

where $t = 0, 1, 2, \ldots$ is a discrete time value. The value $0 < \alpha(t) < 1$ is the learning rate, which determines the statistical accuracy of the neuron map and the ordering of the neurons on the map. The function $h_{ci}(t)$ acts like a neighborhood function and for convergence $h_{ci}(t) \rightarrow 0$ as $t \rightarrow \infty$. Neighborhood function determines the rate of change for neurons on the map [28, 29, 38].

Neurons in the neighborhood $N_{BMU}$ of the winning neuron $w_{BMU}$ will be updated in accordance to a chosen criterion during the learning step. During this the smoothing of the map takes place. The matching law used in Equation (1) and the updating law in Equation (2) need to be compatible.

The algorithm is stochastic, which means that the reliability of the learning is also dependent on the number of training steps. Kohonen [29] proposes an empirical rule of thumb of 500 times the count of network units (neuron map size) for the total number of training steps.

The result of the SOM is a 2-dimensional neural map representing the $l$-dimensional input space. Each input sample vector $X_s$ has a neuron on the neuron map, which describes its parametric vector the best on the two-dimensional map.

3.2. Stability of SOM clustering

The obtained clusters depend on the initial neuron map size $(m \times n)$, learning rate $\alpha(t)$, neighborhood function
Figure 1: Visualization of the turbulent plasma simulation data. Each pixel has four different features that we visualize here: \( n/n_0 \) (top left), zoom-in of \( n/n_0 \) (top right panel), \( J_\parallel \) (middle left panel), \( B_\perp \) (middle right panel), \( (J_\parallel \cdot E) \) (bottom left panel) and \( \Gamma \) (bottom right panel.) The top left panel shows the full image domain whereas the rest of panels show a close-up region of the image (marked with red-dashed line in the top left panel.)
Applying the SOM algorithm to an image of size $r \times t$ will result in dividing the original data into $n$ clusters. Each pixel $p_{ij}$, where $i \in \{1, \ldots, r\}$ and $j \in \{1, \ldots, t\}$, on this image can be associated with a cluster from the set of detected $n$ clusters $\{C_1, \ldots, C_n\}$, where $n, C_1, C_n \in \mathbb{N}$.

For every cluster $C_k$ in the set of clusters $\{C_1, \ldots, C_n\}$ a mask $M_k$ can be defined. A mask is a boolean matrix $M_k = (m_{i,j}) : r \times t$, such that:

$$m_{i,j} = \begin{cases} 0 & \text{if } p_{ij} \notin \text{cluster } C_k, \\ 1 & \text{if } p_{ij} \in \text{cluster } C_k. \end{cases}$$

Elements in this mask matrix $M_k$ consists of values 1 if the pixel is in the observed cluster $C_k$ and 0 otherwise. Using the set of clusters $\{C_1, \ldots, C_n\}$ we have now defined $n$ mask matrices $M_1, \ldots, M_n : r \times t$. The mask matrix describes the locations and area of the structure on the image providing a mapping from the cluster groupings into the initial data view.

We can then apply the SOM algorithm $N$ independent times on the same data. This means that for each pixel $p_{ij}$ on the image we will have $N$ independent clusters. In other words, we have gained $N$ cluster sets $\{C_1, \ldots, C_n\}, \ldots, \{C_1, \ldots, C_n\}$, with $n_1, \ldots, n_N$ elements. From these, we can then create $N$ independent sets of mask matrices $M = \{M_1^1, M_1^2, \ldots, M_n^N\}$ according to Equation (3). We use a notation where the upper index of a mask matrix refers to the SOM realization index and the lower index to the detected cluster in that realization.

4.2. Mask-to-mask stacking

Let us randomly select a set of masks $M^b = \{M_1^1, \ldots, M_n^b\}$, $1 \leq b \leq N$, from the set of independent mask sets $M$. We call this mask set the base mask set. We compare each mask in $M^b$ to every other mask in the set $M \setminus M^b$. For every mask in $M^b$ we obtain $n_1 + n_2 + \cdots + n_N - n_b$ comparisons. The comparison is performed between every mask in a randomly chosen base mask set against every other mask set independent of $M^b$. This process is carried out as long as each set of masks has been chosen as the base mask set.

In supervised segmentation the obtained result is compared to the known truth about the data. In the unsupervised case the base mask matrix is compared to every other cluster realization from $M \setminus M^b$. The stacking of masks from $M \setminus M^b$ on the base mask from $M^b$ is used to evaluate the reliability of the base mask.

4.2.1. Similarity measures

We define three different similarity measure matrices for a mask $M^b_e$ in the base mask set $M^b$, with $n_b$ masks and $1 \leq e \leq n_b$, and a mask $M_f^c$ from a mask set $M^c$, with $n_c$ masks, $1 \leq f \leq n_e$ and $b \neq c$. These matrices are:

The union matrix $U : r \times t$, with possible values of $\{0, 1\}$, defined as

$$U(i,j) = \begin{cases} 0 & \text{if } M^b_e(i,j) = 0 \land M^c_f(i,j) = 0, \\ 1 & \text{otherwise}, \end{cases}$$

where $i \in \{1, \ldots, r\}$ and $j \in \{1, \ldots, t\}$. The intersection matrix $I : r \times t$, with possible values of $\{0, 1\}$, defined as

$$I(i,j) = \begin{cases} 1 & \text{if } M^b_e(i,j) = 1 \land M^c_f(i,j) = 1, \\ 0 & \text{otherwise}, \end{cases}$$
where \( i \in \{1, \ldots, r\} \) and \( j \in \{1, \ldots, t\} \). The sum matrix \( R : r \times t \), with possible values of \( \{0, 1, 2\} \), is defined as

\[
R(i, j) = \begin{cases} 
0 & \text{if } M_b^k(i, j) = 0 \land M_f^j(i, j) = 0 \\
1 & \text{if } M_b^k(i, j) = 1 \land M_f^j(i, j) = 0 \\
2 & \text{if } M_b^k(i, j) = 0 \land M_f^j(i, j) = 1 \\
\end{cases} \quad (6)
\]

where \( i \in \{1, \ldots, r\} \) and \( j \in \{1, \ldots, t\} \).

We calculate these similarity measures between mask \( M_b^k \) in the base mask set \( \mathcal{M}^b \) and every mask in the set \( \mathcal{M} \setminus \mathcal{M}^b \). This is done until all masks in the set \( \mathcal{M} \) have been chosen as a base mask set.

### 4.2.2. Signal strength

Using the similarity matrices we can construct a so-called signal strength \( s_I \) measure. It quantifies how well the observed cluster resembles other clusters. We denote two independent clusters similar, if their mask matrices are identical.

The signal strength metric \( s_I \) is identical to the Mean Intersection-over-Union (MIoU) metric commonly used in supervised segmentation. In the supervised learning the MIoU metric is used to compare the detected segmented objects to the ground truth — information that is lacking in the unsupervised learning case. The \( s_I \) metric is defined between independent realizations of unsupervised clustering algorithms. Every evaluation set is taken as the ground truth and each of these ”true” cluster masks matrices are compared to every other cluster result from independent realizations.

The signal strength gives an estimate for the strength of intersection of the mask matrices, measuring the intersection in comparison to the union of the two masks. For any base mask matrix \( M_b^k \) and a random mask \( M_f^c \), where \( b \neq c \), the signal strength scalar \( s_I \) is defined as

\[
s_I = \frac{\sum_{i,j=1}^{r \times t} I(i, j)}{\sum_{i,j=1}^{r \times t} U(i, j)} \quad (7)
\]

The signal strength scalar \( s_I \) weighs the intersection matrix (Equation (5)) of two masks with their union matrix (Equation (4)). This is done to eliminate the cluster size dependency; large clusters will likely have more common pixels with any other clusters in independent SOM runs, so we need to counter this effect by normalizing the quantities with the union measure.

If \( \sum_{i,j=1}^{r \times t} I(i, j) \rightarrow \sum_{i,j=1}^{r \times t} U(i, j) \), then \( s_I \rightarrow 1 \). This means that the two masks have the value 1 in the same locations and in same quantity in their \( r \times t \) matrices. In that case, these two independent SOM algorithm evaluations have detected the same shape structures in the same pixel locations.

The signal strength scalar \( s_I \) is a normalized quantity, which makes it a quantitative measure to order and rank different masks. The mask \( M^k_b \) will have \( n_1 + n_2 + \cdots + n_N - n_b \) elements in the vector of all signal strength scalars, a vector noted as \( s^k_I \). Each mask in the set of all masks \( \mathcal{M} \) will obtain a signal strength vector. Using this measure we can order the masks.

#### 4.2.3. Quality of cluster unions

Another accompanying measure to the strength is the so-called quality measure as it gives an estimate to the quality of the union of the stacked masks. The best quality measure for a specific cluster corresponds to having a union of two masks close to their intersection. This corresponds to a small residual area, area of the symmetric difference in set theoretical notion, between the masks. This means that their mask matrices overlap perfectly resulting in a union that is equal to the intersection. The worst quality measure, on the other hand, corresponds to a case when the area of the union of two masks is close to the area of their sum.

For any base mask matrix \( M_b^k \) and a random mask \( M_f^c \), where \( b \neq c \), the quality scalar \( q_U \) is defined as

\[
q_U = \frac{\sum_{i,j=1}^{r \times t} U(i, j)}{\sum_{i,j=1}^{r \times t} R(i, j)} - \frac{\sum_{i,j=1}^{r \times t} I(i, j)}{\sum_{i,j=1}^{r \times t} R(i, j)} \quad (8)
\]

The value of quality scalar \( q_U \) goes to zero as \( \sum_{i,j=1}^{r \times t} I(i, j) \rightarrow \sum_{i,j=1}^{r \times t} U(i, j) \), which means the two independent stacked masks \( M_b^k \) and \( M_f^c \) have detected exactly the same structure in the same locations on the image. On the other hand, as \( q_U \) goes to unity, the element sum of the union matrix \( U(i, j) \) and element sum of the sum matrix \( R(i, j) \) is equal. This means that the intersection of the two masks is negligible and they correspond to completely different structures. The union quality measure \( q_U \) can be used to discard clusters from independent SOM runs that have detected completely different structures from other evaluations.

This metric is similar, but not equivalent, to the Dice coefficient that is widely used in the supervised image segmentation. Using the intersection and union matrices defined in this paper, the Dice coefficient is defined as

\[
D = \frac{2 \sum_{i,j=1}^{r \times t} I(i, j)}{\sum_{i,j=1}^{r \times t} R(i, j)} \quad (5)
\]

The Dice coefficient is positively correlated with the MIoU whereas the \( q_U \) of cluster masks is negatively correlated to the MIoU, or \( s_I \) in our concept.

### 4.3. Stacking of multiple masks

In this subsection we will generalize the mask-to-mask comparisons to general quantities averaged over the complete set of independent realizations. To do this, we combine the signal strength \( s_I \) and quality of union \( q_U \). We will denote this matrix with \( G \), as it estimates the goodness of the fit between independent cluster masks. The measure peaks in areas, where the base mask has often a high percentage of its area in common with other detection algorithm clusters. This results in highlighting the highest probability structures on the image. The measure gives a quantitative value to estimate the goodness of
fit between independently detected cluster masks. Cluster
masks, which have detected the same structures, will
fit together well and contribute more to the total sum of
the measure. Similarly, cluster masks that do not repre-
sent the same physical structures end up not contributing
significantly to the total integrated goodness measure for
that base mask.

In subsections 4.2.2 and 4.2.3 a mask $M^b_c$ in the base
mask set $\mathcal{M}^b$ is combined with every mask in the set $\mathcal{M} \setminus
\mathcal{M}^b$. This corresponds to $n_1 + n_2 + \cdots + n_N - n_b$ signal
strengths $s_I$ and qualities of union $q_U$. The goodness of fit
of a cluster mask $M^b_c$ to any other cluster mask $M^b_l$ from
$\mathcal{M} \setminus \mathcal{M}^b$, is defined as

$$G = \frac{s_I}{q_U} \cdot (M^b_c \cup M^b_l), \quad (9)$$

where $s_I$ and $q_U$ are the signal strength scalar and quality
of union scalar of the masks $M^b_c$ and $M^b_l$. The matrix
$M^b_c \cup M^b_l$ refers to all the pixels on the image that belong
to either of these two cluster masks. For an observed mask
in the base mask set $\mathcal{M}^b$ we obtain $n_1 + n_2 + \cdots + n_N - n_b$ matrices according to the Equation (9), which have
the corresponding quotient value in the union of the two
compared masks. Then for every base mask $M^b_c$ in $\mathcal{M}^b$ all
$n_1 + n_2 + \cdots + n_N - n_b$ of its $G$ matrices can be summed
together to yield

$$G_{\text{sum}} = \sum_{k=1}^{N} \sum_{l=1}^{n_k} \left( \frac{s_I}{q_U} \cdot (M^b_c \cup M^b_l) \right), \quad (10)$$

where $k \neq b$. This is done for every base mask, until all
mask sets in $\mathcal{M}$ have been chosen as a base mask set. We
then obtain $n_1 + n_2 + \cdots + n_N$ summed goodness of fit
matrices $G_{\text{sum}}$. Each of them characterizing the fit of a
cluster mask to all other cluster masks detected in other
independent SOM cluster evaluations.

By summing the quotients of the signal strength and
the union quality in the Equation (10), we will accumulate
value to pixels, where the base mask and other indepen-
dent masks have detected a structure. The value added
to the $G_{\text{sum}}$ by a base mask and a compared mask will be
high for those pairs, that have a similar number of pixels
assigned to the cluster and their masks have value 1 in
same location. The contribution from mask pairs, which
detect distinct structures on the image, is negligible since
the $\frac{s_I}{q_U}$ will be close to zero in value for those cases.

We apply the stacking framework on clusters detected
by the SOM algorithm. From SOM results we attain in
total $n_1 + n_2 + \cdots n_N$ cluster masks. After calculating the
$G_{\text{sum}}$ for a randomly chosen base mask $M^b_c$, each pixel on
the image obtains a value between 0 and 1 illustrating its
stability of belonging to the cluster with that shape and
location. In a sense every cluster realization from all of
the SOM realizations is viewed as the ground truth for a
cluster.

Next we define a scalar value for the goodness of fit for
$\mathcal{M}^b$ as the following

$$g_{\text{sum}} = \sum_{\mathcal{M} \setminus \mathcal{M}^b} \frac{s_I}{q_U}, \quad (11)$$

where the sum is taken over all the comparisons between
$\mathcal{M}^b$ and every other mask in $\mathcal{M} \setminus \mathcal{M}^b$.

By ordering the scalar $g_{\text{sum}}$ of all cluster masks, we can
find the base mask, which has learned most of the informa-
tion that describes this cluster. In other words, this mask
fits the best with all the other independent realizations
of that cluster. Same cluster realization base masks will
likely follow the winning base mask stack on the ordering
of the $g_{\text{sum}}$ value. Correlated classifiers for clusters from
an unsupervised ensemble framework indicate an accurate
classification [40, 41, 42]. A base mask detecting a cluster
from the image is a classifier for a given object. This
means that cluster masks with $g_{\text{sum}}$ values that are highly
related indicate that the cluster realization is accurate.

Therefore, same clusters are bound to have similar
goodness of fit measures; a large change in its value indi-
cates a physically different cluster group or an non-accurate
classifier for the given group.

4.4. Summary of the SCE algorithm

SCE framework is summarized in Figure 2. The input
data is a set of simulation images obtained from an
astrophysical simulation, each pixel has 3 different phys-
ical features attached. These images are fed into $N$ in-
dependent unsupervised clustering algorithms, to obtain
$\{C_1, \ldots, C_{n_1}\}, \ldots, \{C_1, \ldots, C_{n_N}\}$ cluster realizations, with
$n_1, \ldots, n_N$ elements (in this work the Self-Organizing Map
was used). Clustering results are then deconstructed into
sets of mask matrices

$$\mathcal{M} = \left\{ \{M^1_1, \ldots, M^1_{n_1}\}, \ldots, \{M^N_1, \ldots, M^N_{n_N}\} \right\},$$

where each mask $M$ represent a cluster detected in the
image in the specific clustering run. These masks are
then compared with all other independent cluster masks
with a quality of union metric and signal strength metric.
These similarity measures are combined together to create
a goodness-of-fit metric the $G$. Then for each cluster mask
all $n_1 + n_2 + \cdots + n_N - n_b$ of its $G$ matrices are summed
together to yield the $G_{\text{sum}}$ matrix. These stacked matrices
for each detected cluster are the output of the SCE
framework.

5. Applying SOM and SCE

In this section we apply the SOM and then the SCE
method on complex images of turbulent magnetically dom-
ninated collisionless plasma simulations described in Section
2. The images we dissect — and that inspired us to de-
velop the discussed method — are 2-dimensional computer
simulation images. Our main aim is to dissect the simula-
tion snapshot pixels into distinct physical structures based
Figure 2: Step-wise description of the SCE framework. The framework (i) takes images as an input, (ii) performs multiple independent segmentations of them, (iii) combines the different segmentation results into so-called quality and signal strength measures, and (iv) outputs statistically combined ensemble averages of these realizations.
on the similarity of their feature vectors. We are, in particular, interested in automating the grouping of pixels into three different physical shapes: magnetic flux tubes (islands), current sheets (long stripes), and background medium [see e.g., [32, 33, 34] for a more detailed discussion of the physics].

Similar problem setups can also be envisioned in many other fields of science like, for example, in medicine where medical images need to be dissected and segmented into different structures automatically.

5.1. Results of SOM segmentation

We apply the SOM algorithm to 8 consecutive simulation image snapshots. We use rectangular shaped neuron map with dimensions (15, 10). The learning rate values are chosen to be 0.6, 0.7, or 0.8. Number of total iterations is chosen to be 10 000, 20 000, 30 000, 40 000, or 50 000.

SOM algorithm is applied on the studied images with all the possible aforementioned parameter combinations. This gives 15 independent SOM clustering results for the astrophysical plasma simulation images. Figure 3 visualizes results from four representative SOM clustering algorithm realizations projected back to the original image view. The resulting clusters can be directly compared to Figure 2. The clusters detected by the SOM algorithm on Figure 3 correspond to distinct regions in the original images that are also visible in Figure 1.

The output of the algorithm is dependent of the randomized nature of the initial neural map, sampled input data and the process parameters. As a consequence, the resulting clusters differ for each SOM outcome. The geometric sizes of resulting segmented cluster regions vary significantly between independent realizations.

5.2. Results of SCE segmentation

In our showcase we stack a set of 62 SOM cluster masks, which were obtained from the 15 independent SOM runs with different process parameters. The goodness-of-fit metrics described in Section 4 were used to find most stable structures in the image and discard bad SOM maps.

As an example, Figure 4 visualizes two cluster maps of two SOM algorithm results labeled as map A and map B. Map A has detected 5 clusters and map B 3 clusters. Thus, the corresponding mask sets are $M^A = \{ M_0^A, M_1^A, M_2^A, M_3^A, M_4^A \}$ and $M^B = \{ M_0^B, M_1^B, M_2^B \}$. By taking the cluster mask $M_i^A$ from map A and cluster mask $M_j^B$ from map B, we can calculate the intersection matrix $I$ (Equation (5)) and union matrix $U$ (Equation (6)). Figure 4 bottom row images show the intersection $I$ and union $U$ matrices for these masks. The pixels colored black correspond to the value 1 and white to 0. The $I$ matrix highlights the common pixels of $M_i^A$ and $M_j^B$. The $U$ matrix highlights pixels belonging to both $M_i^A$ and $M_j^B$. One can see that $I$ and $U$ are similarly shaped and have similarly positioned structures, suggesting that map A and map B have detected the same structure from the input image.

The signal strength $s_I$ (Equation (7)) for base mask $M_i^A$ and a mask $M_j^B$ is shown on the left panel of Figure 5. The metric compares the fraction of overlap between mask $M_i^A$ and mask $M_j^B$ (see Figure 4). Value of the quantity is the highest on the location where the two masks overlap.

The right panel of Figure 5 visualizes the quality of the union metric $q_U$ (Equation (8)). The $q_U$ depicts how well the two cluster masks $M_i^A$ and $M_j^B$ align on top of each other. If their position is exactly the same, then the value $q_U$ for the pixels approaches 0. This corresponds to having identical masks. In Figure 5 the pixels with values close to 0 indicate a good fit between the two masks.

Figure 6 shows the total integrated scalars of goodness of fit, $g_{sum}$ (Equation (11)) for all the masks detected by the 15 SOM evaluations. We have visually inspected the resulting clusters and recovered the 3 major physical structures from these results: background pixels, circular islands, and sheets.

The background plasma (red points in Figure 6) is detected the best among all the independent SOM algorithm evaluations, since the sum of the stacked $\frac{s_I}{s_I}$ values are the highest. The second best cluster detected is that of the magnetic flux tubes or islands, (dark-blue diamonds in Figure 6). The third best detected structure is the current sheets (violet triangles in Figure 6). This manual classification is seen to correlate fairly well with the corresponding goodness-of-fit value. A large change in the value of $g_{sum}$ is seen to match well with the change of the physical meaning of the clusters. As was discussed in Section 4.3, this fact could be further used to group different clusters between different SOM realizations together, since a same structure is expected to have a similar $g_{sum}$ even if a different base map is used. Additionally, highly correlated base masks detecting same physical structures indicate a high accuracy for detecting the structure.

We also see that after the cluster ranked 50th there is a sharp drop in the value of $g_{sum}$ indicating that these clusters are only weakly correlated with other clusters in the SCE. We use this fact to discard these points and select only the masks with rank < 50 as having a meaningfully strong signal (in comparison to being just noise). A statistically robust method to analyse, group, and select different data clusters based on their goodness-of-fit metrics is left for future work.

5.3. Advantage of applying the SCE

The best base mask clusters (highest $g_{sum}$ values) with their corresponding goodness-of-fit matrix $G_{sum}$ values are visualized in Figure 7. Here each pixel has accumulated a value (see Equation (10)) to the matrix $G_{sum}$ from every comparison in the SCE. These four panels describe the four best detected distinct clusters in the input data images. These $G_{sum}$ matrices are the main outcome of the SCE algorithm and can be used at least in three different ways to help with the data analysis. The four “winning” clusters of each physical cluster set are denote with star symbols in the Figure 6.
Figure 3: Visualization of four different SOM evaluations projected back to the original image view. Each of the SOM runs group the initial data pixels into clusters, $C_i$. The pixels are colored based on the cluster it is associated with in order to highlight the physical structures that the data clusters correspond to.
Figure 4: Visualization of the map-to-map operations performed between two SOM cluster map realizations. Top panels show a comparison of two SOM runs (focusing on a small region of the complete image) for which the pixel group information is projected back to the original image view. The SOM map A has detected 5 clusters and the SOM map B has detected 3 clusters. The lower-left panel shows the intersection matrix of $M_A^{C_1}$ ($C_1$ cluster mask of the SOM A) and $M_B^{C_0}$ ($C_0$ cluster mask of the SOM B). The lower-right panel shows the union matrix for the same cluster mask combinations.

Figure 5: Resulting signal strength $s_I$ (left panel) and quality $q_U$ (right panel) values computed between the two clusters of map A and map B (visualized in Figure 4).
Background

Islands
Sheets

Other

\begin{equation}
\log(\frac{\sum U}{\sum U_E})
\end{equation}

Figure 6: Total scalar values of the goodness-of-fit, (see Equation 11) for all the 15 independent SOM runs in the analysis. The clusters are shown in descending order. Each result was visually inspected and labeled into three different empirical categories: background pixels (red), circular islands (blue), sheets (violet), and other unclassified shapes (black). The cluster mask with the highest \(g_{sum}\) value and the best accuracy inside each physical cluster is denoted with the star symbol.

Firstly, the activated pixels in each \(G_{sum}\) images are visually seen to capture each empirically defined classification category; the benefit of the framework is that this work is now fully automated as it could also be performed by just ranking the masks by their \(g_{sum}\). In the practical sense, these matrices can now be used for easily viewing the different physical structures that emerge from the clustering analysis.

Secondly, the resulting \(G_{sum}\) matrices provide a clear way of defining the actual segmentation boundaries: we can now set a direct contour cutoff value for the \(G_{sum}\) providing a quantitative way of selecting which pixels belong to which cluster. This is especially helpful since we need to perform statistically robust geometrical analysis of the resulting shapes.

Thirdly, throughout this analysis we have used the 8 million pixels composed of a feature vector \(X_k = (B_\perp, J_\parallel, [\vec{J}_I \cdot \vec{E}]\). However, nothing guarantees that these features are the best to detect these physical shapes. Therefore, the presented framework can also be used to determine the best combination of the features for detecting the clusters of interest from the observed images. In practice, the goodness of fit measures can be compared between the different SOM algorithm realizations that have been trained with distinct feature vectors on the same input data. The actual numerical value can be easily used to rank the different feature vector combinations. We leave this analysis for further work.

5.4. Analysis of SCE results

The SCE framework was designed to combine decisions of independent unsupervised clustering results for image data. The clusters present in our astrophysical plasma simulations represent fine geometrical structures. The SCE algorithm is designed to tackle the prevailing issue of not gaining robust regions of interest for the geometrical structures of interest. This is especially important for fine geometrical shapes, whose geometrical features such as width and length are of interest.

Figure 8 compares clusters from two independent ensembles, SCE\(_1\), and SCE\(_2\), and different SOM segmentations to assess the quantitative improvement of using the proposed method. It shows pairwise comparisons of all clusters detected by the different methods. We use the signal strength as our comparison metric. It is mathematically defined for the comparison of independent unsupervised clustering realizations in Section 4.2.2 (Equation 2). The signal strength \(s_I\) describes the amount of pixels similarly classified by the two classifiers in relation to the union of the compared clusters. The metric is equal to the IOU metric defined in supervised learning. Each diamond on the Figure 8 represent the \(s_I\) value between one cluster from SCE\(_1\) and another cluster from SCE\(_2\). The \(s_I\) estimates the agreement between two independent cluster realizations. If the clusters consist of pixels describing the same phenomenon on the image, their \(s_I\) will be close to 1. For the contrary, if the clusters describe completely different phenomenon, the metric will be close to 0.

As described in Section 1.3 the SCE framework actually results in a continuous \(G_{sum}\) scalar for each pixel on the image, and we therefore have a freedom in selecting how to quantize each mask. We have tested that the SCE threshold level does not play an important role in the results by testing threshold values of 1 and 2 (blue vs. orange diamonds).

Figure 8 demonstrates that the SCE algorithm is capable of stabilizing the regions of interest for geometrical structures on images, the \(s_I\) for SCE\(_1\) vs SCE\(_2\) is close to 1 for many clusters. The \(s_I\) for the pairwise comparisons of independent clusters in SOM set 1 and in SOM set 2 are significantly lower and they reach the highest agreement of around 0.8. We therefore conclude that the independent clusters from two different SCE results agree systematically better than the independent SOM cluster realizations.

We also note that the distributions of \(s_I\) for all comparisons are actually bimodal, as expected theoretically. The difference originates from comparing masks that have detected the same cluster (corresponding to high similarity and therefore high \(s_I\)) and from comparing physically different clusters (poor similarity and low \(s_I\)). The cutoff value is present around 0.6 for SCE comparisons and at 0.2 for SOM comparisons.

6. Conclusion

We use computer vision and machine learning tools to automate the segmentation of physical structures from 2-dimensional image snapshots originating from large supercomputer simulations of fully-kinetic turbulent plasma flows. Machine learning clustering algorithms provide a fast and powerful method for detecting and segmenting visually distinct structures in data. This makes them an attractive choice for automating data segmentation of com-
Figure 7: Goodness-of-fit matrices ($G_{\text{sum}}$) of the four best detected SCE cluster maps (with log$_{10}$-scale colors). Each panel shows the highest ranking and highest accuracy cluster from Figure 6 ordered based on their integrated goodness-of-fit quantity (Equation 11). The three empirically derived categories are clearly visible in the results: background pixels (top-left panel), islands (top-right panel), and current sheets (bottom-left panel). The last bottom-right panel is seen to be a mixture of many clusters (mainly circular islands and sheets); it is correctly ranked as only the fourth best category and could therefore be safely discarded from any further analysis.
computer simulation results: Firstly, such algorithms are typically fast to evaluate which allows to couple them directly to the simulation time advance loops. Secondly, unsupervised algorithms need very little initial input from the user.

Many such clustering algorithms are, however, non-deterministic and so the end result depends on the initial conditions and different technical process parameters that are used. This can be an especially prohibiting feature in science applications where accurate and stable ROI boundaries are needed for e.g., in order to perform geometrical measurements of the segmented objects. Here we designed a new machine learning framework that combines clustering results from multiple different segmentation realizations. By averaging segmentation results from many independent clustering realizations the presented Statistically Combined Ensemble (SCE) algorithm can yield much more accurate ROI objects. The SCE algorithm can be used to

- determine optimal number of meaningful clusters present in the input data,
- uniquely identify each pixel with the corresponding object cluster, and
- segment clustered pixels into stable contours that can be further analyzed for their geometrical shape, size, and area.

The SCE algorithm uses image cluster masks as a base unit for the stacking operations; each individual clustering realization (obtained here via SOM algorithm) can be split into different image masks where only the pixels from one specific cluster are ‘active’. We defined stacking operations of these masks by using the union, intersection, and sum of two masks. These quantities are shown to reflect the cluster signal strength and quality of the mask matching in each comparison. The resulting quality and strength measures can be combined with a weighted average over the complete SCE. This procedure enables to stack and combine information from many parallelly evaluated clustering realizations. This, in turn, enhances the accuracy of the detection algorithm and makes it suitable for use in many science applications where accurate and stable ROI boundaries are needed.

In the future, we plan to apply the SCE method for studying the spatial and temporal properties of intermittent turbulent structures found in the simulations discussed.

Acknowledgements

We thank Pekka Manninen for useful discussions related to designing the stacking operations and the two anonymous referees for their comments and suggestions that helped to improve the paper. MB would like to thank Elmo Tempel and Radu S. Stoica for support and discussions. MB acknowledges support from NORDITA via the Visiting PhD program and JN via the NORDITA Postdoctoral Fellowship. MB acknowledges the financial support by the institutional research funding IUT40-2 of the Estonian Ministry of Education and Research and the support by the Centre of Excellence “Dark side of the Universe” (TK133), which is financed by the European Union through the European Regional Development Fund. The work of MB was also supported by the European Research Council Consolidator grant 682068-PRESTISSIMO. The work has been partially performed under the Project HPC-EUROPA3 (INFRAIA-2016–1–730897), with the support of the EC Research Innovation Action under the H2020 Programme. The simulations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at PDC.

References

[1] D. Biskamp, Magnetohydrodynamic Turbulence, Cambridge University Press, 2003. doi:10.1017/cbo9780511535222
[2] V. M. Uritsky, A. Pouquet, D. Rosenberg, P. D. Mininni, E. F. Donovan, Structures in magnetohydrodynamic turbulence: Detection and scaling, Phys. Rev. E. 82 (2010) 056326.
[3] V. Zhelazny, D. A. Uzdensky, J. C. Perez, P. Boldyrev, Statistical Analysis of Current Sheets in Three-dimensional Magnetohydrodynamic Turbulence, ApJ 771 (2013) 124.
[4] A. Chatterjee Azizabadi, N. Jain, J. Büchner, Identification and characterization of current sheets in collisionless plasma turbulence, arXiv e-prints (2020) arXiv:2009.02881.
[5] R. Dupuis, M. V. Goldman, D. L. Newman, J. Amaya, G. Lapenta, Characterizing Magnetic Reconnection Regions Using Gaussian Mixture Models on Particle Velocity Distributions, ApJ 889 (2020) 22.
[6] A. Hu, M. Sisti, F. Finelli, F. Califano, J. Dargent, M. Fagnanello, E. Camporeale, J. Tenhissen, Identifying Magnetic Reconnection in 2D Hybrid Vlasov Maxwell Simulations with Convolutional Neural Networks, ApJ 900 (2020) 86.
