Application of a self-organizing map to identify the turbulent-boundary-layer interface in a transitional flow

Zhao Wu, Jin Lee, Charles Meneveau, and Tamer Zaki

Phys. Rev. Fluids **4**, 023902 — Published 7 February 2019

DOI: 10.1103/PhysRevFluids.4.023902

Application of a self-organizing map to identify the turbulent-boundary-layer interface in a transitional flow

Zhao Wu, Jin Lee, Charles Meneveau, and Tamer Zaki
Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA

(Dated: December 9, 2018)

Existing methods to identify the interfaces separating different regions in turbulent flows, such as turbulent/non-turbulent interfaces, typically rely on subjectively chosen thresholds, often including visual verification that the resulting surface meaningfully separates the different regions. Since machine learning tools are known to help automate such classification tasks, we here propose to use an unsupervised self-organizing map (SOM) machine learning algorithm, as an automatic classifier. We use it to separate a boundary layer undergoing bypass transition into two distinct spatial regions, the turbulent boundary layer (TBL) and non-TBL regions, the latter including the laminar portion prior to transition and the outer flow which possibly contains weak free-stream turbulence. Both regions are separated by the turbulent boundary layer interface (TBLI). The data used in this study are from a direct numerical simulation, and are available on an open database system. In our analysis of one snapshot in time, every spatial point is characterized by a 16-dimensional vector containing the magnitudes of the components of total and fluctuating velocity, magnitudes of the velocity gradient tensor elements, and the stream-wise and wall-normal coordinates, all normalized by their global standard deviation. In an unsupervised fashion, the SOM classifier separates the points into TBL and non-TBL regions, thus identifying the TBLI without the need for user-specified thresholds. Remarkably, it avoids including vortical streaky structures that exist in the laminar portion prior to transition as well as the weak free-stream turbulence in the turbulent boundary layer region. The approach is compared quantitatively with existing methods to determine the TBLI (vorticity magnitude, cross-stream velocity fluctuation). Also, the SOM classifier is cast as a linear hyperplane that separates the two clusters of data points, and the method is tested by finding the TBLI of other snapshots in the transitional boundary layers data set, as well as in a fully turbulent boundary layer with similar levels of free-stream turbulence. Variants in which the approach failed are also summarized.

I. INTRODUCTION

One of the most striking properties of inhomogeneous turbulent shear flows is the turbulent boundary layer interface (TBLI), the notional surface that separates regions of near-wall turbulence from the outer flow, or free stream, that can be either nearly irrotational or weakly turbulent. The study of this surface has a long history that goes back several decades [1–3]. A variety of tools and methods exist to define and measure ~~the~~ such interfaces [4, 5]. The first step in identifying turbulent interfaces is to define the criterion or detector flow variable to distinguish between turbulent and non-turbulent or weakly turbulent (outer) regions in the flow. One can then introduce an indicator function that depends on the detector flow variable, and a threshold.

A most basic feature characterizing turbulence is vorticity[1, 4, 6, 7], which is thus a natural choice to use as a detector flow variable. At first sight, it would seem relatively straightforward to discriminate between vortical and irrotational regions and to define the interface position based on a very low threshold on vorticity magnitude. However, the free-stream turbulence intensity in experiments or numerical simulations may be finite, or data may be noisy. Moreover, the situation is particularly difficult in flows in which the laminar portion of the flow may have small-scale vorticity such as in transitional boundary layers, where the high values of wall vorticity and streaky structures in the laminar regions complicate choosing an appropriate threshold of vorticity.

Another property of turbulence, velocity fluctuations, motivates using turbulent kinetic energy as a detector flow variable. For example, Chauhan *et al.* [8] and de Silva *et al.* [9] used the turbulent kinetic energy measured in a frame moving with the free stream as the detector function, while Anand, Boersma, and Agrawal [10] used instantaneous streamwise velocity directly in a jet flow. In some other studies, passive scalar concentration fields have been used, e.g. Westerweel *et al.* [5], Prasad and Sreenivasan [11]. Many of the methods have been reviewed in [12] and more recent contributions can be found in Borrell and Jiménez [6], Jahanbakhshi and Madnia [7], Philip *et al.* [13], Watanabe *et al.* [14], Wu *et al.* [15], Zhou and Vassilicos [16].

To overcome the difficulties inherent for transitional boundary layer flow, Nolan and Zaki [17] proposed a function based on the velocity fluctuations. Since turbulence is manifest by significant fluctuating velocity events, the quantity they suggested is the sum of the absolute values of the wall-normal and spanwise fluctuation field, excluding the streamwise component since Klebanoff streaks are predominantly streamwise velocity perturbations. Since the interface separates the TBL from both the free stream and also the upstream non-turbulent region, it extends down to the wall. As a result, a single threshold on the detector flow variable is not possible, which led Nolan and Zaki [17] to set different thresholds at different wall-normal heights using Otsu's method[18], and then reconstruct the 3D turbulent structure plane by plane. Meanwhile, Lee and Zaki [19] utilized the streamwise vorticity component to separate the turbulent regions from the transitional boundary layer and the free stream.

Even with a suitable choice for a detector flow variable, the choice of the threshold can be challenging. The selection of the appropriate threshold often relies on the common observation that there is a range in which many statistics are only weakly affected on the threshold value, like conditional velocities relative to the TBLI shape or fractal dimension [9]. Usually, this process is based on examination of the PDF profile of the detector flow variable. If a plateau or minimum in the PDF can be observed, a value within this plateau or at the minimum can be assessed as a threshold to detect the TBLI (da Silva *et al.* [12], Lee, Sung, and Zaki [20]). However, the choice of the threshold within plateau regions could cover wide ranges if the plateaus are extensive [6], and sometimes the PDFs do not display distinct minima or plateaus. In such scenarios, selecting a threshold becomes a trial-and-error process, and the final choice is strongly based on the researcher's subjective judgement.

Independent of the quantitative measures used to detect the TBLI, when we examine flow visualizations of turbulent flow, distinguishing what is turbulent and non-turbulent appears visually rather clear to us, perhaps because of a natural ability to make such visual distinctions. Automating such intuitive classifications is an area where new "machine learning" tools are known to perform well, especially in cases when large amounts of data can be used for training. For example, Hack and Zaki [21] used supervised learning to successfully distinguish stable and unstable laminar streaks in a transitional boundary layer. In the present study, we explore the use of one such machine-learning tool to detect the TBLI aiming to avoid having to choose thresholds and detector functions. We will find that users must still make some informed *a-priori* choices, and that the proposed methodology cannot be regarded as fully automatic or agnostic about the physics involved. Still, the proposed method will be shown to provide successful identification of the TBLI and other interesting results.

In the present study, we utilize clustering into two arbitrary categories, which is a form of unsupervised machine learning, to classify the flow into what will turn out to be (*a-posteriori*) the turbulent boundary layer (TBL) and non-TBL regions. The unsupervised clustering classifies objects so that similar objects are grouped as the same group. Here, unsupervised means that the input data, or observations, are "unlabelled" – an *a-priori* classification or categorization is not included in the observations. This is very important to the current problem since we do not know ahead of time whether a point is turbulent or non-turbulent~~, even in a "training set"~~. We wish to avoid having to first label some data points as TBL or non-TBL correctly for a supervised classification training process, since then we would need to "set" some "threshold" while labelling. We choose the self-organizing map (SOM) by Teuvo Kohonen [22] as the clustering algorithm, but other methods such as the "k-means" algorithm [23, 24] lead to similar results.

It is important to include a clarifying note about nomenclature: TBL region in this paper refers to the turbulent spots in the transitional region and the near-wall boundary-layer turbulence after transition. Meanwhile, non-TBL refers to the laminar boundary layer, laminar portions in the transitional region and the outer flow, the last of which might be turbulent if it contains free-stream turbulence. Both are separated by the TBLI. In the context of the transitional boundary layer flow studied here, we think TBLI is a better term to describe the interface separating the TBL and non-TBL regions than "turbulent/non-turbulent interface" (TNTI), which is usually used in literature.

Section II details the data set used in this paper, obtained from a Direct Numerical Simulation (DNS) of bypass transitional boundary layer at Reynolds numbers up to $Re_\theta = 1070$. The section also provides a brief description of the open database system that now includes this transitional boundary-layer data set. Section III provides basic background on the SOM clustering method used in this study as well as the particular data that are used to construct the input vector for the SOM algorithm. Section IV presents results. First, for illustrative purposes, a lower-dimensional (three dimensional) case is considered, namely on the wall where only the two wall stress components and downstream distance are used as input vectors to distinguish TBL and non-TBL regions on the wall. Then the method is applied to the full 3D flow domain, where the input vectors form a 16-dimensional data space. In section V the performance of the SOM is compared to existing traditional detector functions to find the TBLI. Also, we characterize the TBLI by reporting PDFs of several variables on the TBLI which typically display wide range of variation, in order to further demonstrate that using thresholds can be challenging. Approaches that did not lead to successful clustering are briefly discussed. Finally, conclusions are presented in section VI.

II. TRANSITIONAL BOUNDARY-LAYER DATA SET

A data set of a transitional boundary layer with free-stream turbulence, which was simulated by Lee and Zaki [19] is used to demonstrate the capability of the SOM for TBLI detection. The data set is archived in the Johns Hopkins Turbulence Database (JHTDB) system (`http://turbulence.pha.jhu.edu`) [25–27]. The flow configuration is shown in figure 1(a) and other simulation and data set details can be found at `https://doi.org/10.7281/T17S7KX8`. The streamwise, wall-normal and spanwise axes are represented by x, y and z in Cartesian coordinates, and the corresponding velocity components are u, v and w. Unless otherwise stated, all subsequent results are normalized by the free stream velocity, U_∞, and δ_{99_0}, which is the 99% boundary-layer thickness at the inlet of the stored region.

Figure 1(b) shows the skin-friction coefficient C_f plotted against the streamwise location. The gray lines are the skin-friction correlations for the boundary layer, in which the turbulent skin-friction is estimated by $0.370(\log_{10}\mathrm{Re}_x)^{-2.584}$ [28] while the curve for laminar flow is given by $0.664\mathrm{Re}_x^{-1/2}$. The boundary-layer thickness δ_{99} is shown as a function of x in figure 1(c).

Figure 2 shows contours of streamwise velocity on a single plane at height $y = 0.50$. It shows the streaky structures in the laminar portion that are vortical but should not be counted as part of the TBL portions in the flow. The latter appear first as spots that grow and merge, ultimately forming the TBL region (see Zaki [29] for a recent review of bypass transition). Distinguishing among these regions is the main challenge considered in this work.

III. SELF-ORGANIZING MAP AS CLUSTERING TOOL

Since our main goal is to distinguish between TBL and non-TBL regions in the flow, we consider machine learning "classifier" methods that can cluster the data into groups. The SOM by Teuvo Kohonen [22] is an unsupervised machine learning algorithm and is often used as a clustering tool. "Unsupervised" means that humans do not need to interfere in the training process, and it also means that the data need not be "labeled", meaning that we do not need to know ahead of time how to distinguish the flow regions.

A. Review of SOM method

A SOM ~~consists of a competitive learning neural network of nodes (or "neurons")~~ is a competitive learning algorithm that is fundamentally different from machine learning methods that apply error-correction learning (e.g. multilayer feedforward networks). For clarity the description below is the original SOM ~~with only two nodes~~, and will be followed by a summary of a more efficient variant (batch SOM) adopted in this work. Here, we also refer to an illustrative physical example in order to assist in the description of the SOM algorithm, but the generality is retained. As the

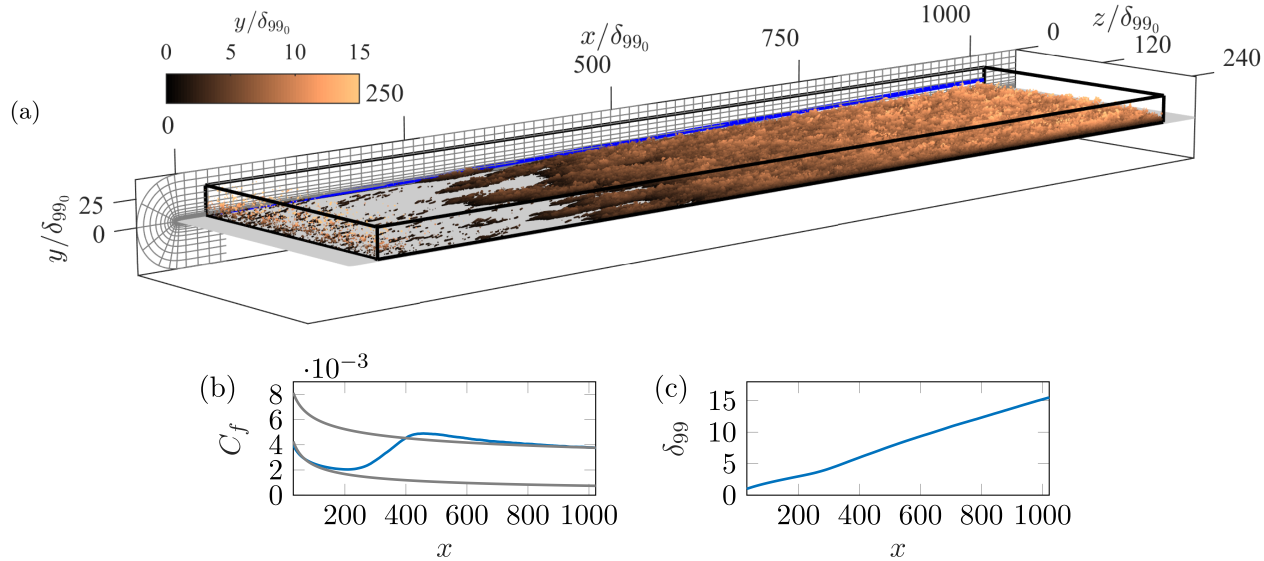

FIG. 1. (a) Flow configuration of the current transitional boundary layers with free stream turbulence. Flow is in the x direction and from left to right. The region covered by data provided in JHTDB is shown as black box. Instantaneous coherent structures are identified by iso-surface of Q-criterion, colored by their wall normal heights. The boundary layer thickness δ_{99} is shown as blue line, and the value at the inlet of the stored region, δ_{99_0}, is used as reference length scale. (b) Skin-friction coefficient C_f and (c) boundary layer thickness δ_{99} as a function of streamwise position x. The lengths are normalized by δ_{99_0}.

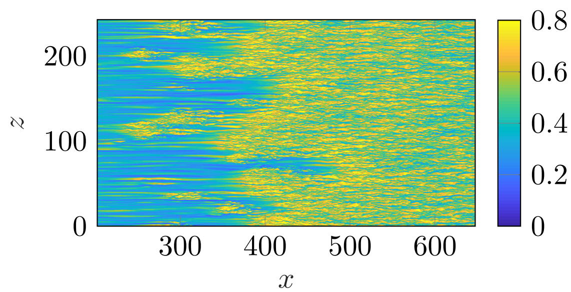

FIG. 2. Contours of streamwise velocity on a single plane at height $y = 0.50$. In the laminar region there are streaks with streamwise vorticity that should not be counted as turbulent region.

method is introduced, we will refer to how it would be applied to the identification of the TBLI at the wall—the problem examined in detail in §III C.

SOM consists of components called nodes. In general, an M-group clustering task involves M nodes ~~– here we will use $M = 2$ since we only wish to classify each point in the flow as either TBL or non-TBL~~. In the present study, we will use $M = 2$ nodes since we only wish to classify each point in the flow as either TBL or non-TBL. Each of the nodes has a position ("weight") in the space of input vectors that we wish to classify. In our application, we will use a list of certain flow variables as the components of the input vector. ~~In our first illustrative example to be presented below in~~ In the example of wall TBLI identification (§III C), we will use three variables: magnitudes of the two wall-stress components, $|\partial u/\partial y|_{y=0}$ and $|\partial w/\partial y|_{y=0}$, and the streamwise location x. We will be interested in distinguishing TBL and non-TBL regions on the wall plane.

In the space of input vectors, the two nodes are first initialized with random positions $\mathbf{X}(v)$, where $v = 1$ (for the TBL node) or $v = 2$ (for the non-TBL one). From there a sample vector from the input data set is selected randomly ($\mathbf{D}_k$) and its Euclidean distances to the two node vectors are calculated. The node who is closest to the selected input data is termed as the best matching unit (BMU) u, i.e.

$$||\mathbf{D}_k - \mathbf{X}(u)|| \le ||\mathbf{D}_k - \mathbf{X}(v)||, \forall v = 1, \ldots, M. \tag{1}$$

The update of the SOM weights, i.e. the update of the positions of the SOM nodes in the input space, corresponding

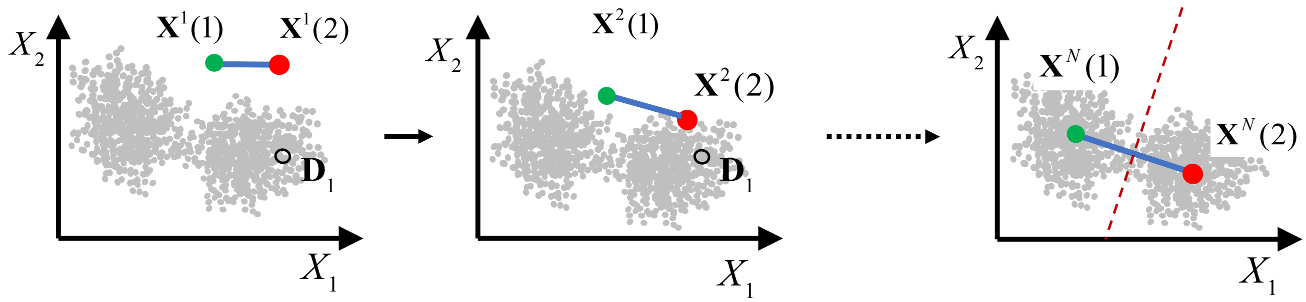

FIG. 3. Sketch of original Self Organizing Map (SOM) as applied to a problem with two classes. Data (grey points) are described by two coordinates $\mathbf{X} = (X_1, X_2)$ (a 2D state vector). The green and red circles represent the two nodes. The open circle represents the first randomly chosen datapoint with position $\mathbf{D}_1$ towards which the closest of the initial nodes is drawn most strongly. After a few (N) iterations over the training data and nodes, the two nodes move to locations representing the clusters.

to $\mathbf{D}_k$ and BMU $\mathbf{X}(u)$ is

$$\mathbf{X}^{n+1}(v) = \mathbf{X}^n(v) + h^{(n)}(u,v)\left[\mathbf{D}_k - \mathbf{X}^n(v)\right], \forall v = 1, \ldots, M, \tag{2}$$

where n is the iteration index and $h^{(n)}(u,v)$ is a neighborhood function. From here we see that by selecting one input data $\mathbf{D}_k$, the entire SOM map weight vectors will be updated, and then the algorithm will advance to the next iteration after selecting all the data points (in a random order). The neighborhood function $h^{(n)}(u,v)$ can be, for example, the learning rate $\alpha^{(n)} \in (0,1)$ if the distance between $\mathbf{X}(v)$ and $\mathbf{X}(u)$ is smaller than the neighborhood radius $r^{(n)}$ and zero otherwise. Both $\alpha^{(n)}$ and $r^{(n)}$ are usually decreasing monotonically as iteration index n increases to ensure the convergence of the results. Usually, the iterations with $r^{(n)}$ greater than some threshold (which is typically chosen as unity if the input variables are properly normalized) are called ordering phase. In this phase, the network orders itself to maintain the topological features of the input data in the input space. After $r^{(n)}$ decreases to less than or equal to unity, the iterations are called tuning phase, since for $r^{(n)} \leq 1$ only the BMU itself will learn from the selected data sample.

The sketch in figure 3 illustrates the above SOM approach. The two nodes ($v = 1, 2$) are initially placed randomly at $n = 1$. After the first iteration $n = 2$, the node that is initially closest to some randomly chosen data point $\mathbf{D}_{k=1}$ (shown as empty circle), namely node $v = 2$ at initial position $\mathbf{X}^1(2)$, is drawn towards that data point to arrive at $\mathbf{X}^2(2)$. Note that $\mathbf{X}^1(1)$ is also dragged towards the data points and is repositioned at $\mathbf{X}^2(1)$, because the distance between $\mathbf{X}(1)$ and $\mathbf{X}(2)$ is smaller than the initial neighborhood radius r^1. Iterating by drawing randomly from all the data repeatedly, the two nodes tend to a configuration where they are placed near the "center" of each of the two distinctive groups of data. The data can now be classified by proximity to either of these nodes, thus defining a line (or hyperplane) separating the two clusters (shown as dashed red line in the sketch).

For faster and simpler computations, a batch SOM algorithm is often used instead of the original SOM (Eq. 2) described above. In the batch algorithm, a sub-list of all the input data points $\mathbf{D}_v$, who all have the BMU $\mathbf{X}^n(v)$, are collected. The number of the data points in this sub-list is denoted as $m_v^{(n)}$, and the mean position of the data points within this sub-list is denoted as $\overline{\mathbf{D}}_v^n$. The weight (position) vector of $\mathbf{X}^n(v)$ is then moved to the center of all the input vectors for which it is a BMU or for which it is in the neighborhood of a BMU, i.e.

$$\mathbf{X}^{n+1}(v) = \frac{\sum_{w \in P_v^{(n)}} m_w^{(n)} \overline{\mathbf{D}}_w^n}{\sum_{w \in P_v^{(n)}} m_w^{(n)}}, \forall v = 1, \ldots, M, \tag{3}$$

where the neighborhood set $P_v^{(n)}$ consists of all nodes within the neighborhood radius $r^{(n)}$ from node v at iteration n. Similarly to the original SOM, the batch SOM (Eq. 3) contains an ordering phase (when $r^{(n)} > 1$) and a tuning phase (when $r^{(n)} <= 1$); the tuning phase of the batch SOM is identical to the k-means algorithm [23, 24]. However, the SOM is less likely to be trapped in local minima than k-means due to the coupling of nodes in the ordering phase [30]. It should be noted that the batch SOM contains no learning rate function $\alpha^{(n)}$. It provides more stable asymptotic values for the weight (position) vectors $\mathbf{X}(v)$ than the original SOM. For both the original or batch SOM, the weight vectors of the nodes can be initialized to random values.

This algorithm is implemented in many software packages, such as MATLAB's ~~neural network~~ machine learning toolbox which we use here, scikit-learn (a Python library) and TensorFlow (a Google's open-source software library).

In MATLAB, the neighborhood radius function $r^{(n)}$ is given as:

$$r^{(n)} = 1 + (r^{(1)} - 1)(1 - \frac{n-1}{n_o}), \tag{4}$$

where $r^{(1)} = 3$ is the initial neighborhood radius and $n_o = 100$ is the number of iterations of ordering phase. The memory and computational time requirements of this algorithm are linearly proportional to the product of the size of the input data $\mathbf{D}$ and the number of clusters M.

B. SOM inputs and outputs, and postprocessing

To identify TBL and non-TBL regions, the machine learning algorithm must have information about the flow. As reviewed in §I, velocity, velocity perturbations and vorticity (which is a linear function of velocity gradients) contain useful information to distinguish TBL/non-TBL regions. To make the current unsupervised machine learning method as simple as possible, we here use local velocity and first-order spatial information, i.e. we use velocity, velocity fluctuations and velocity gradients. It should be noted that the instantaneous spanwise velocity w and its fluctuation w' are the same, since the time-averaged spanwise velocity, which is in the homogeneous direction, is zero. Therefore, we only keep w (or w') in the input data. One should also note that, since the turbulence is manifest by large fluctuations, the magnitude of these variables, rather than the value itself, is used as representative of "turbulence". Therefore, we use the absolute values of these variables as input features. Besides these flow variables, the x and y coordinates are also important, since the flow in a boundary layer develops downstream and the turbulent region expands in the wall-normal direction. ~~To avoid biased sampling from the non-uniform DNS grid in the y-direction, the data are spatially interpolated onto a uniform grid.~~ We should also emphasize that the data are spatially interpolated onto a uniform grid in order to avoid biased sampling due to the non-uniform DNS grid in the y-direction. The clustering of grid point in the near-wall region, which was required to resolve the flow, would cause TBL data points to be much more numerous than the non-TBL points. Such imbalance in the data can have adverse influence on the performance of clustering algorithms [31].

How to scale, or non-dimensionalize the input features, is very important in the use of SOM algorithm. This is because that SOM use the Euclidean distance to measure the similarity between vectors. If one variable has values over three orders of magnitude (e.g. the x-coordinate) and another variable has values only up to one (e.g. the streamwise velocity u), the former will dominate the similarity metrics while the latter will show negligible impact. Thus, one would usually want the input features to be equally important at least in an initial guess. The easiest way to equalize the variables is to normalize them all to unit-variance. Hence, all variables f are standardized to f_s where $f_s = f/\sigma_f$ and σ_f is the standard deviation of f, computed over the entire flow domain considered in the analysis. We note that the mean of f is not subtracted since simple translations in the state space do not affect the results. As an example, the normalization of $|u'|$ is performed using its variance $\sigma^2_{|u'|} = \overline{\left(|u'| - \overline{|u'|}\right)^2}$, where the overline denotes averaging over the entire sample space.

As mentioned before, the number of clusters has to be specified in advance. As the goal of this work is to develop a method to identify the TBL/non-TBL regions with the least possible user intervention, we set the number of clusters to $M = 2$, with the expectation that two clusters will represent the TBL and non-TBL regions respectively. The inputs to the SOM algorithm are summarized in table I. It should be emphasized that the current method does not use any neighboring point information other than the gradients; only local data are used as input. Using neighborhood information may cause unwanted spatial filtering on the data which will smooth the TBLI [9].

A training using 120 million data points with 16 dimensions would take 1 hour with 100 GB memory. However, After the training, the SOM outputs the final position (weight) vectors of the two nodes in the space of input data. Those data points whose input vectors are closer to the weight vector of one node are classified as one group, while the other points are the other group.

Lastly, a post-processing step is undertaken to account for small-scale intermittency. Even within the TBL region there are many small regions (holes) that should be considered part of TBL but that could fall into the non-TBL group during the SOM classification. In order to count such points as TBL, any topologically closed region that is classified as non-TBL and fully surrounded by the TBL region (non-TBL holes) will be "filled" and re-classified as TBL. The TBLI will then be the surface separating both regions.

Input #	Description	Expression	Normalization (σ)
1-3	Instantaneous Velocity	$\|u\|_s$, $\|v\|_s$, $\|w\|_s$	0.1312, 0.0229, 0.0264
4-5	Velocity fluctuations	$\|u'\|_s$, $\|v'\|_s$	0.0400, 0.0229
6-14	Velocity gradients	$\|\partial u/\partial x\|_s$, $\|\partial v/\partial x\|_s$, $\|\partial w/\partial x\|_s$ $\|\partial u/\partial y\|_s$, $\|\partial v/\partial y\|_s$, $\|\partial w/\partial y\|_s$ $\|\partial u/\partial z\|_s$, $\|\partial v/\partial z\|_s$, $\|\partial w/\partial z\|_s$	0.0243, 0.0246, 0.0270 0.1595, 0.0296, 0.0536 0.0626, 0.0385, 0.0306
15-16	Coordinates	x_s, y_s	286.6778, 8.2739
17	Number of clusters, M	2	—

TABLE I. Inputs to the SOM algorithm. Inputs # 1-16 are standardized input features, and each of them has unit-variance. The normalization column shows the standard deviations σ of the input features in the entire 3D domain.

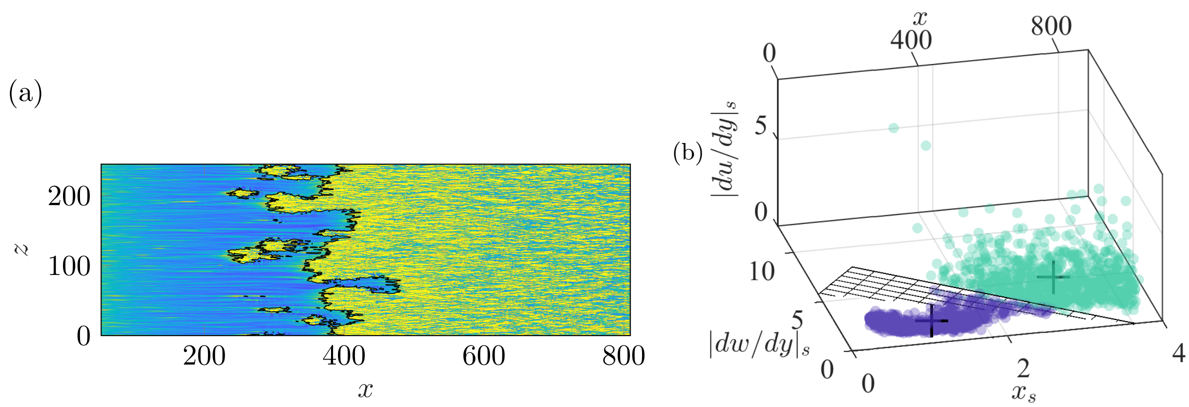

FIG. 4. (a) Wall contour of $|\partial u/\partial y|_s$ and identified TBLI using SOM. (b) Scatter plot of $|\partial u/\partial y|_s$, $|\partial u/\partial y|_s$ and x_s: blue circles, non-TBL data points; green circles, TBL data points; +, weight (position) vectors of two ~~neurons~~ nodes. The TBLI is a bisecting plane of the two weight vectors in the space constructed by the input features $|\partial u/\partial y|_s$, $|\partial u/\partial y|_s$ and x_s. For convenience, the original x is also shown at the top axis.

C. Illustrative application at the wall in two spatial dimensions

Considering that we are analyzing a 16-dimensional problem (see table I), it is useful first to consider a simpler, lower-dimensional example. A special region in a wall-bounded flow is the no-slip boundary. There, all three velocity components are zero, and therefore the velocity gradients in the wall-parallel directions ($\partial/\partial x$ and $\partial/\partial z$) are strictly zero. In addition, since $\partial u/\partial x = \partial w/\partial z = 0$ at the wall, $\partial v/\partial y = 0$ due to incompressibility. Therefore, only three out of the 16 input features are useful at the wall; they are $|\partial u/\partial y|_s$, $|\partial w/\partial y|_s$ and x_s. Therefore, in this case $\mathbf{X} = (X_1, X_2, X_3)$ with $X_1 = |\partial u/\partial y|_s$, $X_2 = |\partial w/\partial y|_s$ and $X_3 = x_s$.

Figure 4(a) shows the wall contours of $|\partial u/\partial y|_s$ and the TBLI (black line) obtained from SOM applied to the three-dimensional wall data only, clearly showing that the method distinguishes between TBL and non-TBL regions on the wall plane. As desired, laminar streaks are not catalogued as TBL. In this case the input data vectors may be visualized via scatter plot in the three-dimensional state space (figure 4(b)). The two nodes ~~(neurons)~~ onto which the SOM has converged (two black crosses) are located apart from each other. The non-TBL data (blue circles) are all closer to the left node and appear clustered around a curved cylindrical shape. The data points classified as TBL (green circles) appear more spread out and are all closer to the right node. A bisecting plane, equidistant from both nodes, separates both regions. If the distance were not measured with the Euclidean norm, that separating surface may not be a plane. Note that the plane is tilted in all three directions, i.e. the SOM finds that all three state variables are relevant in making the classification into TBL and non-TBL regions. Mapping the plane onto the physical domain, and excluding the holes inside the turbulent region, yields the TBLI (black line) shown in 4(a). Thus it is evident that the identified TBLI corresponds to a bisecting hyperplane in the input state space.

IV. RESULTS FOR THE THREE-DIMENSIONAL FLOW DOMAIN

Next, the SOM is applied to a snapshot of the transitional boundary layer introduced in section II. The entire 3D flow domain is considered by computing the 16 components of the state vector at all points in the flow. To sample physical space in an unbiased fashion, instead of using data on the simulation grid points that are clustered near the wall, we use a spatially uniform mesh consisting of $(n_x, n_y, n_z) = (831, 280, 512)$ points to cover the entire flow domain stored in the database. We use the fourth-order Lagrange polynomial spatial interpolation and fourth-order finite difference differentiation scheme implemented in the JHTDB web services [27]. The SOM is applied using $M = 2$ and it converges after about 500 iterations. The small non-TBL holes inside the TBL region are "filled" as described above. The results can be cast as visualizations of the interface separating the TBL and non-TBL regions, or mathematically as a hyperplane in the 16-dimensional state space.

A. Visualizations of the TBLI

Several visualizations of the TBLI identified by the SOM are shown in figure 5. Panel (a) shows the growth of the TBL region downstream from patches of turbulence. Some selected stream-wise and cross-stream planes (two planes at $z = 122.6$ and at $x = 613.1$) are shown in figures 5(b), (c) and (d) respectively, on which the SOM-identified TBLI is shown alongside $|v'|_s + |w'|_s$ color contours. The visualizations show how the boundary layer grows in the wall-normal direction with downstream distance, and that the free-stream turbulence and the laminar steaks can be distinguished from the TBL region. These visualizations confirm that the SOM can provide satisfactory TBLI detection without using a threshold when applied in the entire 3D flow domain.

We note that the ranges of the variables in the entire 3D domain are substantial. For example, the mean value of $|\partial u/\partial y|$ as function of y varies over two orders of magnitude within the thickness of the boundary layer. As a result, a single threshold set on the gradient, or any other variable, would have been unlikely to work across the entire height. Indeed previous researchers chose different thresholds at different wall-normal heights, and then reconstructed the 3D TBLI (e.g. Nolan and Zaki [17] who used y-dependent thresholds on $|v'| + |w'|$). There are also variations in the streamwise direction, that in the past have been addressed using x-dependent normalizations of vorticity (see the definition of ω^* by Borrell and Jiménez [6] and §V C below) for application to fully developed turbulent boundary layers. For transitional portions of the boundary layer, reformulation of the algorithm is required [19]. In the present method, a threshold based on a linear combination among all 16 input variables is determined by the SOM without additional user input.

Next, we show that the SOM obtained, or trained, on one snapshot of the transitional data set can be used to very efficiently classify another snapshot of the same flow. Figure 5(e) shows the result of applying the trained SOM to another instant of the flow separated from the first snapshot by a time interval $1175\delta_{99_0}/U_\infty$ (significantly larger than the advection time across the transition zone). The results demonstrate that the free-stream vortical perturbations, streaks, spots and the fully turbulent zone are properly identified for an independent realization of the same flow, even though naturally the TBLI is different in its details. The reason for this good performance is that even a single snapshot in the training set is quite large and includes sufficient data to construct an accurate descriptor of the TBLI.

In figure 6, we plot the average height of the SOM-determined TBLI, $\langle y_I(x)\rangle$, normalized by the boundary-layer thickness, $\delta_{99}(x)$. The average was evaluated by applying the SOM to 97 snapshots equi-spaced in time, spanning close to one flow-through time. Since the instantaneous interface undulates to capture the instantaneous edge of the turbulent region, its mean value bears a more physical interpretation, relative to the larger 99% thickness that is based on the mean-velocity profile. At $x = 1000$, $\langle y_I\rangle$ is approaching $0.7\delta_{99}$.

B. Hyperplane representation of the SOM classifier

The outputs of the SOM are the coordinates of the two nodes in the state space as well as the bisecting hyperplane. In our application, the resulting plane is represented according to

$$\boldsymbol{a} \cdot \boldsymbol{X} + 1 = 0, \tag{5}$$

where

$$\begin{aligned}\boldsymbol{a} =&[0.19, -0.15, -0.16, -0.16, -0.16, -0.17, -0.10, -0.15,\\ &-0.15, -0.17, -0.16, -0.17, -0.16, -0.17, -0.08, 0.15]\end{aligned}$$

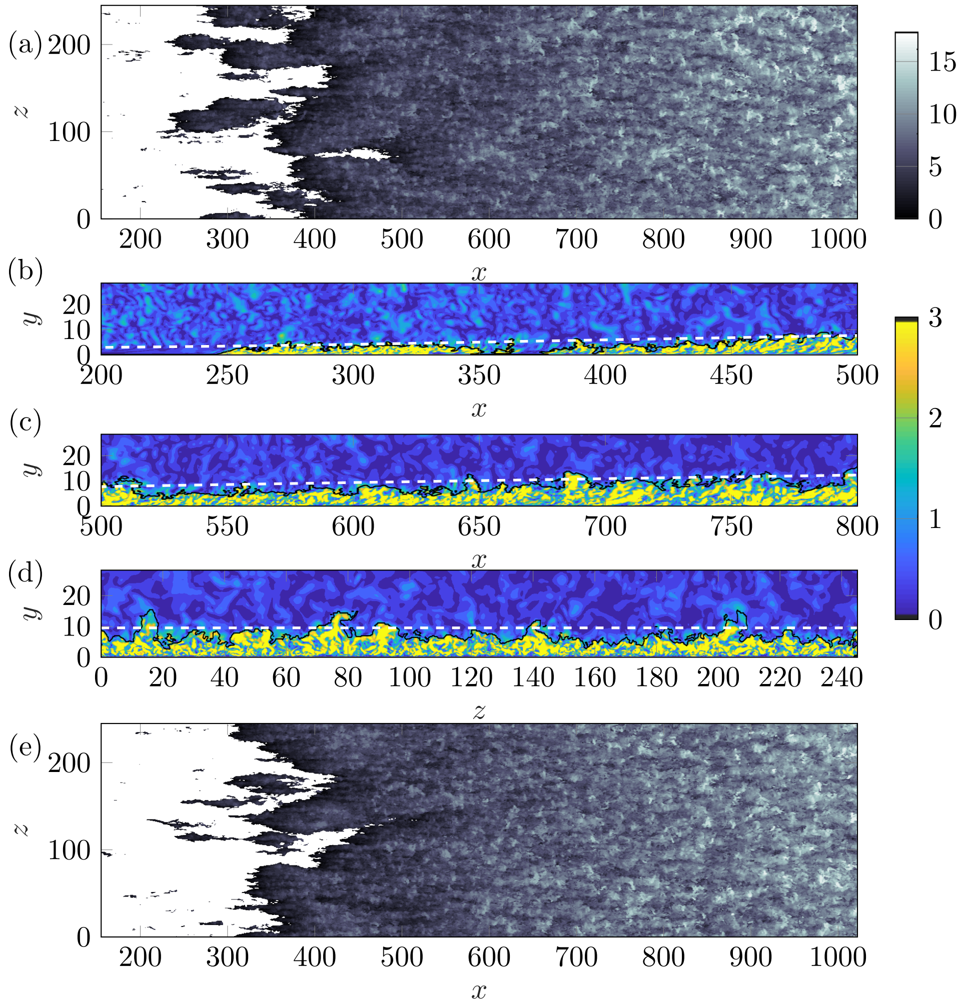

FIG. 5. (a) Top view of TBLI identified by the SOM algorithm. The surface is colored by its local wall-normal height. In (b,c,d), $|v'|_s + |w'|_s$ contours are shown together with the TBLI (black line) and the boundary layer thickness $\delta_{99}(x)$ (white dashed line). Three different cuts are shown: streamwise at $z = 122.6$ between $x = 200$ and 500 (b); between $x = 500$ and 800 (c), and a spanwise plane at $x = 613.1$ (d). Panel (e) shows the TBLI obtained by applying the SOM trained from (a) to another temporal snapshot, separated by a time $1175\delta_{99_0}/U_\infty$. The colormap in (e) is the same as in (a).

are the coefficients on each of the state input variables (flow features) within the vector

$$\begin{aligned}\boldsymbol{X} =&[|u|_s, |v|_s, |w|_s, |u'|_s, |v'|_s, |\partial u/\partial x|_s, |\partial u/\partial y|_s, |\partial u/\partial z|_s,\\ &|\partial v/\partial x|_s, |\partial v/\partial y|_s, |\partial v/\partial z|_s, |\partial w/\partial x|_s, |\partial w/\partial y|_s, |\partial w/\partial z|_s, x_s, y_s].\end{aligned}$$

If $\boldsymbol{a} \cdot \boldsymbol{X} + 1 < 0$ the point is classified as turbulent, while if $\boldsymbol{a} \cdot \boldsymbol{X} + 1 > 0$ it is non-turbulent. The coefficients of x and y have different signs, indicating that these two inputs have opposite effects on the classification: the TBL region becomes dominant as x increases, i.e. farther downstream, and y decreases, i.e. nearer to the wall; conversely, non-TBL region is found at smaller x and higher values of y. This intuitive difference in the sign of x and y is but an example of how the weights of the SOM encode information about the flow. We observe that the coefficients of all 16 flow variables are of the same order of magnitude , which indicates that determination of the TBLI relies on all the input data. Often such analysis can be used to argue that some parameters are irrelevant. ~~Here we find that the method relies on all the input data.~~ We performed additional training (not shown here) which includes z coordinate, as well as those variables in Table I. As anticipated, the resulting coefficient of z was two orders of magnitude lower than other coefficients, which is consistent with z not providing useful information since the flow is homogeneous in that direction. This also demonstrates that the SOM has the ability to discover and disregard irrelevant inputs.

Next, we inquire how different the coefficients would be if we trained the SOM on another snapshot of data, taken at a different time separated by $1175\delta_{99_0}/U_\infty$ (significant larger than the advection time across the transition zone).

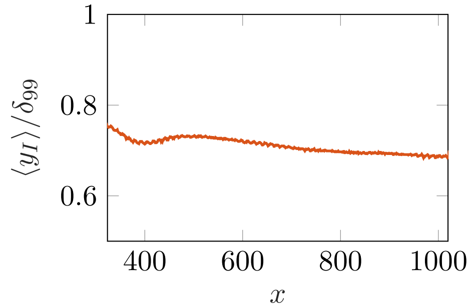

FIG. 6. Average height of TBLI $\langle y_I \rangle/\delta_{99}$.

We find that the coefficients of the hyperplane function (5) calculated from two independent snapshots differ by less than 2%, and only 4 grid points are classified into different region in the whole 3D domain using the two different hyperplane functions. The reason for this insensitivity is that there are sufficient data points in a single snapshot so that the training sample provides a (nearly) complete state-space representation.

Results suggest that this hyperplane representation can be used as a general tool to separate the TBL/non-TBL regions in a transitional boundary layer, at least for the present ranges of free-stream turbulence intensity and Reynolds number.

V. COMPARISON WITH OTHER DETECTION METHODS

Borrell and Jiménez [6] proposed a dimensionless vorticity $\omega^* = \omega(\delta_{99}^+)^{1/2}(\nu/u_\tau^2)$ as the detector variable in a fully turbulent boundary layer. This non-dimensional vorticity becomes independent of the streamwise location, or Reynolds number, and therefore a single threshold can in principle be applied in the entire three-dimensional flow if there is no transitional region in the domain considered for the analysis. For a transitional boundary layer, Nolan and Zaki [17] used $|v'|+|w'|$ as the detector variable, which successfully separated the TBL region from the laminar streaks. To select the threshold, Otsu's method [18], which was first applied to transitional flows by Nolan and Zaki [17] and subsequently adopted by others [32–34], identifies an optimum threshold that minimizes the intraclass variance, or maximizes the interclass variance. In this section, the TBLI from SOM is compared with the two previously proposed approaches: the $|v'| + |w'|$ and ω^* methods, the former also with Otsu's method to chose a threshold.

A. Comparison with the cross-stream fluctuation method

Figure 7(a) shows contours of $|v'|_s + |w'|_s$ on the plane $y = 0.50$, which visually display two distinct regions: the TBL region with high velocity fluctuations and the non-TBL region with small amplitudes. Results from the SOM and from $|v'| + |w'|$ thresholded using Otsu's method are compared, when applied to a single plane, as was done by Nolan and Zaki [17]. The SOM now uses 15 variables because y is fixed, but otherwise proceeds as described above. Figure 7(b) shows the PDF of $|v'|_s + |w'|_s$ from various regions in the flow. The blue filled function shows the overall PDF on the entire plane, while the black dashed and solid lines show the PDFs of $|v'|_s + |w'|_s$ within each of the TBL and non-TBL regions as classified by the SOM. The classification is visualized in figure 7(c). As desired, the SOM does not classify the streaks in the laminar region as TBL (see figure 2), although they contain significant vorticity.

The results from the SOM are compared to the two approaches to identify the TBL regions, one based on the PDF of $|v'| + |w'|$ and the other using Otsu's method to chose the threshold. Considering the former approach, a plateau is seen in the PDF profile between 0.8 and 1.5 (the orange region in figure 7(b)). Here we choose 1.0 as the threshold and the result is shown in figure 7(d). Otsu's method [18] that identifies an optimum threshold that minimizes the intraclass variance (or maximizes the variance among classes) yields a threshold of 2.2 as marked by an orange line in figure 7(b). The resulting TBL and non-TBL regions on the data plane are shown in figure 7(e). By comparing the three methods, while the threshold from Otsu's method is relatively high in this case, it seems that the SOM result is quite similar to the plateau method, on this plane. The PDFs of $|v'|_s + |w'|_s$ in the TBL/non-TBL regions detected by the SOM are shown in figure 7(b), demonstrating again different behavior than a single threshold that would separate the two PDFs into two non-overlapping regions. However, the plateau in the total PDF lays between the two peaks of PDF profiles of the TBL/non-TBL regions found by the SOM method.

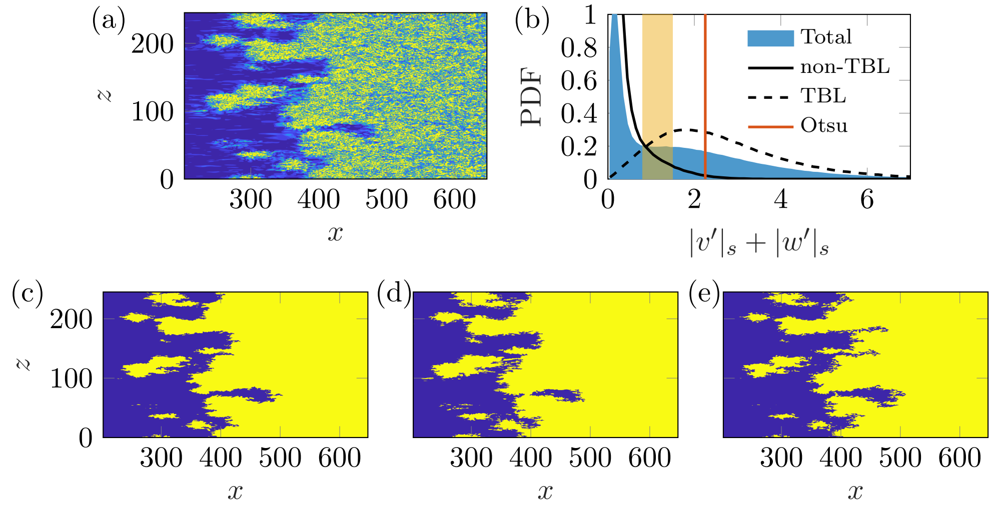

FIG. 7. (a) Contour map of $|v'|_s + |w'|_s$ at $y = 0.50$. (b) The PDF of $|v'|_s + |w'|_s$. The blue filled profile shows the overall PDF on the entire plane and a plateau is seen in the range of values indicated by the orange band. The black dashed and solid lines are the PDFs in the SOM-determined TBL/non-TBL regions. The threshold picked by Otsu's method is shown as the orange line. Panels (c)-(d) show the TBL/non-TBL regions (blue, non-TBL region, yellow, TBL region) identified using: (c) the SOM algorithm, (d) the threshold on $|v'|_s + |w'|_s$ chosen within the PDF plateau, and (e) the threshold identified by Otsu's method.

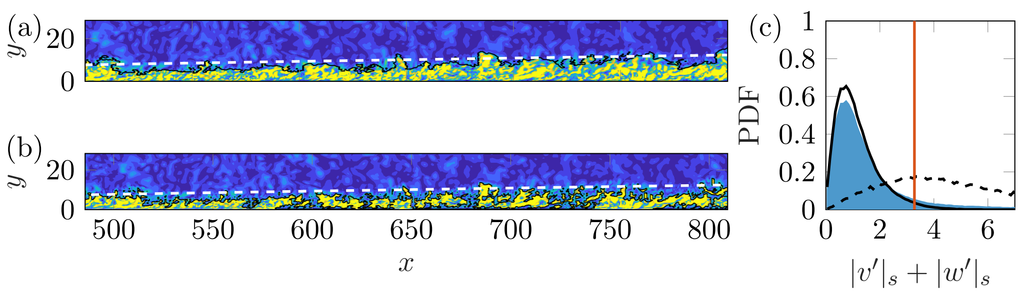

FIG. 8. Results when applying various methods on a streamwise vertical plane at $z = 122.6$ in the transitional boundary layer data set. Panels (a)-(b) show TBL/non-TBL regions identified by the SOM algorithm and Otsu's method applied to $|v'|_s + |w'|_s$, respectively. The background shows $|v'|_s + |w'|_s$ contours, the black line is the TBLI, and the white dashed line is the $\delta_{99}(x)$. (c) shows the PDF of $|v'|_s + |w'|_s$ on the entire plane. See figure 7(b) for legend.

The $|v'| + |w'|$ thresholding method has the drawback that the proper threshold depends on y. To illustrate this known issue [17], we now apply the method in vertical $x - y$ planes, i.e. including variations in y in the data, but attempting to use a single threshold. Figure 8(a) shows the contour of $|v'|_s + |w'|_s$ and the SOM-determined TBL/non-TBL regions (black line) at plane $z = 122.6$. Now the SOM includes the full 16 variables since y is also relevant. Here the free-stream turbulence is clearly seen in the contour plot, but the SOM is able to distinguish it from the near-wall turbulent boundary layer. The PDF of $|v'|_s + |w'|_s$ at plane $z = 122.6$ is shown in figure 8(c). As is evident, there is no plateau region in the PDF and thus the plateau method is not applicable in this case, while the SOM algorithm is not affected. Otsu's method picked the threshold equal to 3.3, leading to results shown in figure 8(b). The SOM provides visibly more appropriate output than Otsu's method which should instead be applied to separate y planes. The PDFs of $|v'|_s + |w'|_s$ in the TBL/non-TBL regions detected by the SOM algorithm are presented in figure 8(c) as well, again confirming that the SOM does not separate the TBL/non-TBL regions based on a single threshold of a single parameter.

B. Vorticity and cross-stream fluctuation methods applied to 3D data

The vorticity magnitude is often used as a detector function for identification of the TBLI (or TNTI in the literature's terminology) — see e.g. Bisset, Hunt, and Rogers [4], Borrell and Jiménez [6], Lee, Sung, and Zaki [20]. Figure 9(a)

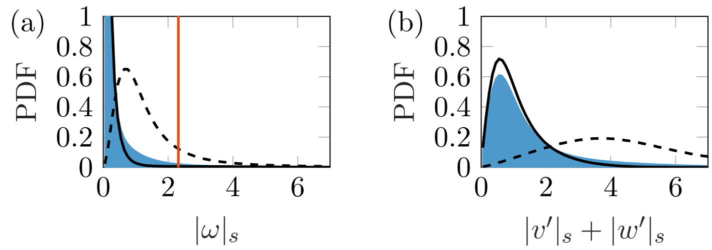

FIG. 9. PDFs of (a) $|\omega|_s$ and (b) $|v'|_s + |w'|_s$ in the entire 3D domain. See figure 7(b) for legend.

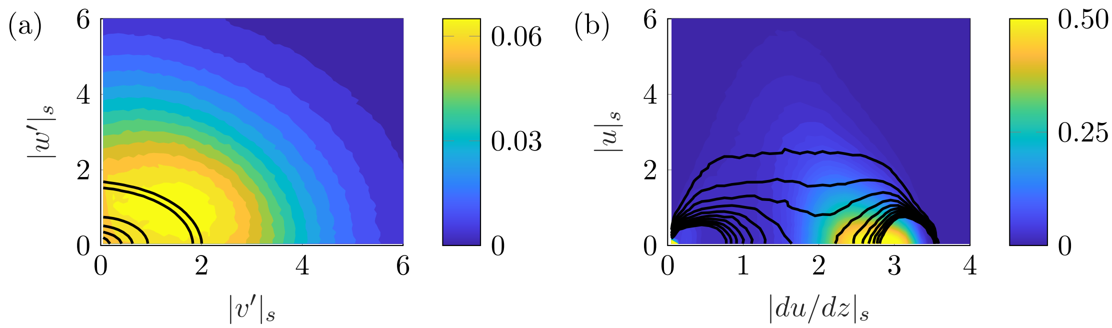

FIG. 10. Joint PDFs of (a) $|v'|_s$ vs. $|w'|_s$ and (b) $|\partial u/\partial z|_s$ vs. $|u|_s$ in the TBL (color contour) and the non-TBL (black line) regions in the entire 3D domain. The lines from top right to bottom left in (a) are isolines with PDFs equal to 0.05, 0.07, 0.5, 1, 1.5 and 2 respectively. The lines from top to bottom in (b) are isolines with PDFs between 0.01 and 0.1 with a constant step of 0.01.

shows the PDF of $|\omega|_s$ in the whole 3D domain (blue region), and in the two regions identified by the SOM (non-TBL as dashed line and TBL as solid line). The figure shows that the vorticity PDF profiles in the TBL and non-TBL regions overlap significantly. The PDF in the non-TBL zone extends to high vorticity values, which may be indicative of streaks in the boundary layer. On the other hand, the SOM-identified TBL region has small vorticity amplitudes near the edge of the boundary layer relative to the near-wall levels. In this way, the vorticity PDF in the TBL region extends to the low vorticity values. Thus, again due to the overlap of the TBL/non-TBL PDF profiles, there should not exist a single threshold to easily separate the TBL/non-TBL regions in the 3D domain. In addition, if one insists on using a single threshold in this case, the threshold should probably be picked between the peaks of TBL/non-TBL region PDF profiles as determined by the SOM; the threshold picked by the Otsu's method (orange line in figure 9(a)) seems too high.

Figure 9(b) shows the PDF of $|v'|_s + |w'|_s$, similar to the analysis in §V A but now in the entire 3D domain. Again the total PDF does not display a plateau hence it is challenging to select a single threshold. This difficulty led [17] to use a threshold that is a function of distance from the wall. Our SOM obviates this step, and is applied directly to the 3D data. In addition, the resulting peaks of $|v'|_s + |w'|_s$ PDF in the SOM determined TBL/non-TBL regions are clearly separated from each other.

Figure 10 shows two selected joint PDF plots obtained in the entire 3D domain within either the TBL (color contour) or the non-TBL (solid lines) regions, as determined by the SOM. The peaks of joint PDFs in TBL/non-TBL regions are overlapped, similar to the PDFs in figure 9, showing that it would appear difficult to choose a single threshold on combinations of these two variables in the entire 3D domain.

C. Comparison with the ω^* method in fully turbulent boundary layer

To compare the SOM with the ω^* method of Borrell and Jiménez [6] which was developed for a fully turbulent boundary layer (i.e. not including transition), we now apply the SOM to a different data set than that considered in §IV, namely on a sub-domain of a fully turbulent boundary layer DNS data set [20].

Figure 11(a) shows the PDF of $\log_{10}(\omega^*)$ at different y/θ_{in} heights in the sub-domain, where θ_{in} is the momentum thickness at the simulation inlet. The PDF has two well-defined regions: the bottom-right corner shows the high vorticity within the near-wall turbulent boundary layer and the top-left region represents the non-turbulent free

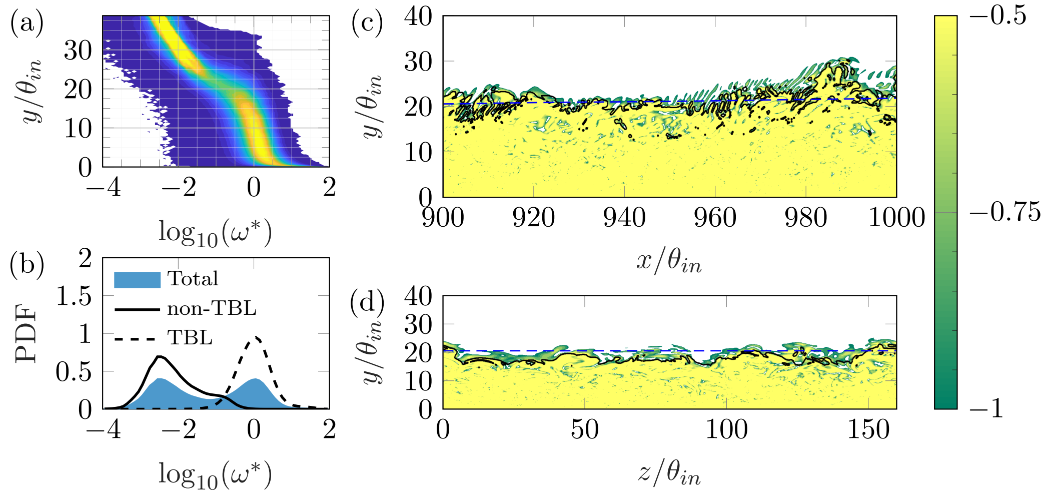

FIG. 11. Fully turbulent boundary layer without free stream turbulence. (a) PDF of $\log_{10}(\omega^*)$ at different y/θ_{in}. (b) PDF of $\log_{10}(\omega^*)$ in the whole 3D domain. (c,d) $\log_{10}(\omega^*)$ contour, TBLIs identified using SOM (black solid line) and δ_{99} (blue dashed line) at $z/\theta_{in} = 1.875$ and $x/\theta_{in} = 900$, respectively

stream. The non-zero vorticity in the ideally irrotational outer flow is owing to the finite accuracy of the numerical scheme. The two regions can be easily distinguished as their vorticity values differ by about two orders of magnitude. The near-wall turbulent region and the free stream are connected by a band which spans over $-1 \leq \log_{10}(\omega^*) \leq -0.5$. This is also seen in the PDF profile evaluated over the entire three-dimensional domain and shown in figure 11(b), filled region: a plateau connects the near-wall turbulent region at right and the free stream with residual low vorticity at left. Previous researchers (e.g. Borrell and Jiménez [6], Lee, Sung, and Zaki [20]) selected vorticity thresholds within this plateau to detect the TBLI.

Figures 11(c,d) show contours of ω^* on two planes (streamwise and cross-stream, respectively), with contours in a range corresponding only to the interval suggested for thresholding from the PDF in figure 11(a) (namely $-1 \leq \log_{10}(\omega^*) \leq -0.5$). The SOM using the same 16 input variables as in §IV is applied to this snapshot of data. The classification into TBL and non-TBL regions yields the interface marked by the black line in figures 11(c,d). The TBLI detected by the SOM falls within the range of $-1 \leq \log_{10}(\omega^*) \leq -0.5$ (figures 11(c,d)). However, it does not correspond to a single scaled vorticity threshold, as demonstrated by the PDFs of ω^* in the SOM's TBL and non-TBL regions shown in figure 11(b). Clearly the SOM can classify the two peaks of the total PDF profile into TBL and non-TBL regions respectively, without using a single threshold for the TBLI detection. We conclude that in this case, the SOM provides results that are similar, but not precisely the same, to those from previously proposed thresholding method using ω^*.

It is important to recall that when using the SOM machine-learning approach, users do not have to normalize the vorticity in the very particular way that ω^* is defined, plot the PDF in a logarithmic scale, and choose a threshold within the plateau if one exists, or check whether the threshold appears (subjectively) acceptable; the SOM algorithm only requires sufficient data input values, normalized by their standard deviations over the domain of interest.

D. Robustness

We have seen that the current SOM method can separate the free-stream turbulence and the near-wall turbulent region (c.f. figures 5 and 8), which is recognized as a challenge for TBLI identification. The question is how robust is the current identification method to varying levels of free-stream turbulence. It would be surprising if the SOM proposed here would work for cases in which the free-stream turbulence levels approach those in the turbulent boundary layer. To explore this question, we have applied the SOM to cross-flow planes at various downstream locations in another fully turbulent boundary-layer data set [35]. It includes higher free stream turbulence intensity. Specifically, the free-stream turbulent intensity in the three selected cross-flow planes (figure 12) are 6%, 3% and 2% respectively. In all planes, the traditional TBLI detection methods should not work: there are no distinct, or well-defined, regions as seen in the case of fully turbulent boundary layer without free-stream turbulence (c.f. figure 11(a)), and the contours of ω_s and $|v'|_s + |w'|_s$ show it would be hard to use a single threshold to find the TBLI. In the plane with 6% FST intensity, the SOM provides somewhat satisfactory TBLI identification. However, some free-stream turbulence is also

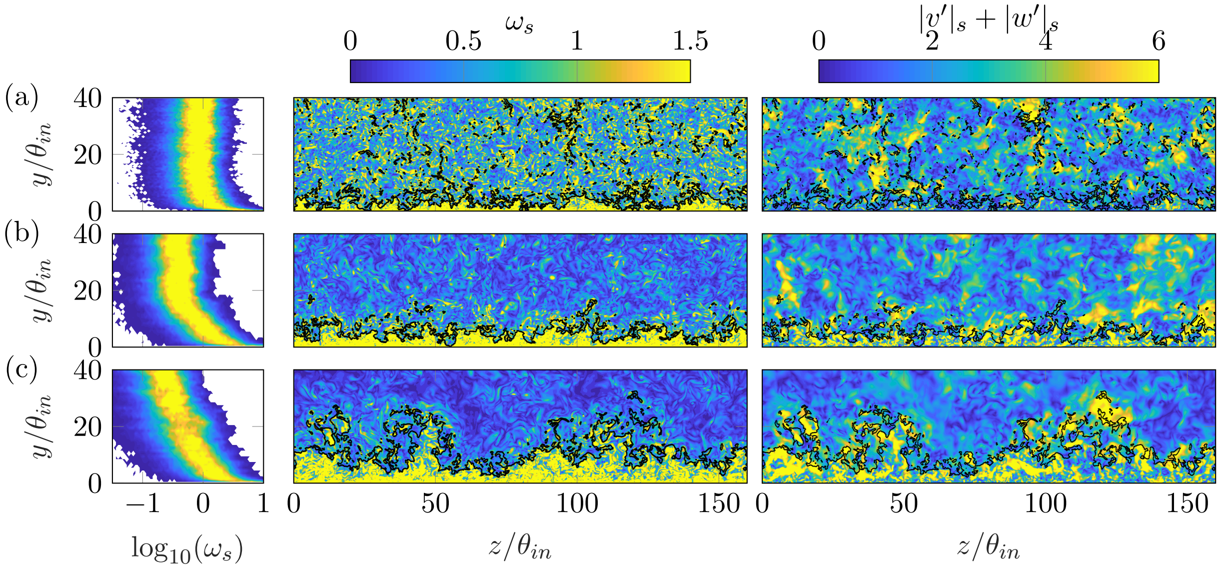

FIG. 12. Three cross-flow planes at different streamwise locations in a fully turbulent boundary layer with free stream turbulence: (a) $x/\theta_{in} = 125$, (b) $x/\theta_{in} = 500$ and (c) $x/\theta_{in} = 875$, where θ_{in} is the momentum thickness at $x/\theta_{in} = 0$. Left column shows PDF of vorticity magnitude ω_s at different wall normal heights, middle column shows the vorticity magnitude ω_s contours and right column shows $|v'|_s + |w'|_s$ contours. The TBLI identified using SOM are shown as black solid lines.

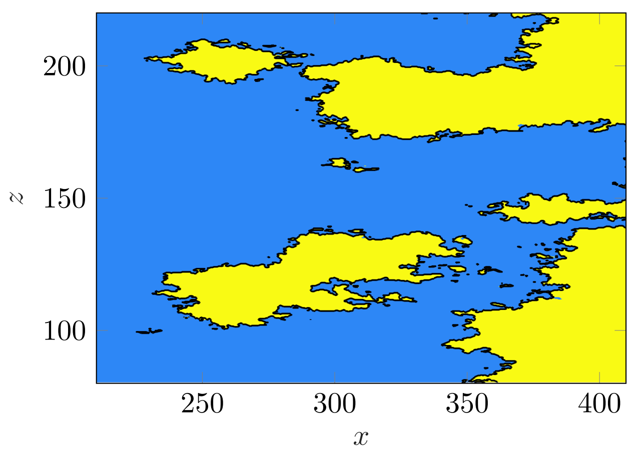

FIG. 13. TBLI detection results using SOM with noise. The input data is at $y = 0.50$ (same as figure 7), but 10% white Gaussian noise has been added to all input variables in the whole domain. The yellow and blue colors are the TBL and non-TBL regions with the original input, while the black line is the TBLI with the noisy data.

detected. The results are cleaner further downstream as the free-stream turbulence decays and becomes closer to the levels of the data set considered in §IV. This shows that while powerful in distinguishing nearly laminar or very weakly turbulent regions from the boundary-layer turbulence, the SOM method as applied here could not clearly distinguish between the high free-stream turbulence and the near-wall turbulence in the boundary layer when the two turbulence levels are comparable. We also note that initial attempts to use 3 classes ($M = 3$) for the entire data to attempt to distinguish possible further classes in the flow did not yield meaningful results. The unsupervised learning was effective only in distinguishing between two classes.

Finally, the question of whether the SOM method is robust to noise is addressed by evaluating the SOM-determined TBLI in data to which noise has been added. Specifically, we add white Gaussian noise to each of the 16 components of the data inputs to mimic the measurement errors: the standard deviation of the noise is 10% of the original values and the mean is zero. The SOM is then applied to this noisy data set, and the obtained TBLI is compared to the results without noise. As shown in figure 13, the identified TBLI is indistinguishable in the two cases.

VI. CONCLUSIONS

In the present study we have proposed to use a SOM, a class of unsupervised machine learning, to classify points in a flow as either belonging to the boundary layer turbulent region or not, and use the classification as a means to detect the TBLI. As input state variables for the SOM algorithm, the magnitudes of velocity, velocity fluctuations and velocity gradients normalized by their standard deviations, were chosen. The hope was that when applied to a transitional boundary layer flow, the algorithm would automatically distinguish between these two types of flow regimes without the need for user-specified thresholds.

The SOM was first tested on a two-dimensional subdomain of the flow, the wall surface. There, only three input variables were used, proportional to the two components of the wall stress and downstream distance. It was confirmed that application of the SOM to this input data yielded a clustering into two unlabelled categories, of which one was clearly the laminar region on the wall including streak signatures, and the other was the fully turbulent region.

The SOM was then applied to a full 3D domain that included weak outer turbulence, streaky laminar regions near the wall before transition to turbulence, patches of turbulence and then the fully turbulent boundary layer. Input variables consisting of magnitudes of velocity, fluctuations, velocity gradients and point position were assembled as 16-dimensional input vectors. When applied to classify the 16-dimensional data into two groups, the SOM yielded two node positions. Each point in the flow could then be compared to these two positions and classified depending on its (Euclidean) distance in the state space of normalized variables. A final post-processing step consisted in filling the typically small laminar holes (topologically closed) that are often found deep in the turbulent region and classifying them also as TBL. Visualizations of the resulting two regions and of the TBLI between them confirmed that the classification results are consistent with the visual appearance of the flow. The classification could be cast as a hyperplane in 16-dimensional state space and the respective coefficients were all non-negligible, i.e. none of the input variables used could be discarded as unimportant. We verified that when SOM was applied to another snapshot, very similar hyperplane coefficients were obtained, and when applied to an entirely different snapshot, the trained SOM also yielded excellent identification of the TBLI.

A more detailed analysis was performed, comparing the approach to vorticity and cross-flow velocity magnitude thresholds. In all cases, examinations of the probability density functions in the identified TBL and non-TBL regions highlighted the difficulties in using single thresholds. Moreover, tests with synthetic noise added to the data yielded nearly identical results.

Certain limitations of the SOM method were identified. User input is required in selecting a list of input flow variables. In particular, the choice of normalization was found to have an effect. For example, we found that when normalizing with the min-max span of each of the input data instead of the root-mean-square, rather poor results were obtained. Also, when applied to a data set in which the free-stream turbulence intensity approached the intensity of the boundary-layer region, not surprisingly the method was not able to uniquely identify only the turbulence in the boundary layer and began to include some of the turbulence from the free stream.

Nonetheless, the overall conclusion is that the SOM-based data clustering approach could successfully distinguish the weakly turbulent outer flow and the strong turbulent boundary layer region, and the interface separating the two regions, in a transitional boundary layer. More work is needed to explore and document applications of SOMs to other flows, with different levels of free-stream turbulence, and also classifying more than two types of flow regions.

ACKNOWLEDGMENTS

The authors acknowledge funding from the Office of Naval Research (grant N00014-17-1-2937) and the National Science Foundation (grants OCE-1633124 and CBET-1605404). Computations were made possible by the Maryland Advanced Research Computing Center (MARCC). Development of FileDB is also supported by NSF grant OAC-1261715. The authors would like to thank J. You for providing the fully turbulent boundary layer data..

[1] S. Corrsin and A. L. Kistler, "Free-stream boundaries of turbulent flows," (1955).
[2] L. S. Kovasznay, V. Kibens, and R. F. Blackwelder, "Large-scale motion in the intermittent region of a turbulent boundary layer," Journal of Fluid Mechanics **41**, 283–325 (1970).
[3] K. R. Sreenivasan and C. Meneveau, "The fractal facets of turbulence," Journal of Fluid Mechanics **173**, 357–386 (1986).
[4] D. K. Bisset, J. C. R. Hunt, and M. M. Rogers, "The turbulent/non-turbulent interface bounding a far wake," Journal of Fluid Mechanics **451**, 383–410 (2002).
[5] J. Westerweel, C. Fukushima, J. M. Pedersen, and J. C. Hunt, "Momentum and scalar transport at the turbulent/non-turbulent interface of a jet," Journal of Fluid Mechanics **631**, 199–230 (2009).

[6] G. Borrell and J. Jiménez, "Properties of the turbulent/non-turbulent interface in boundary layers," Journal of Fluid Mechanics **801**, 554–596 (2016).
[7] R. Jahanbakhshi and C. K. Madnia, "Entrainment in a compressible turbulent shear layer," Journal of Fluid Mechanics **797**, 564–603 (2016).
[8] K. Chauhan, J. Philip, C. M. De Silva, N. Hutchins, and I. Marusic, "The turbulent/non-turbulent interface and entrainment in a boundary layer," Journal of Fluid Mechanics **742**, 119–151 (2014).
[9] C. M. de Silva, J. Philip, K. Chauhan, C. Meneveau, and I. Marusic, "Multiscale geometry and scaling of the turbulent-nonturbulent interface in high reynolds number boundary layers," Physical Review Letters **111**, 044501 (2013).
[10] R. K. Anand, B. J. Boersma, and A. Agrawal, "Detection of turbulent/non-turbulent interface for an axisymmetric turbulent jet: Evaluation of known criteria and proposal of a new criterion," Experiments in Fluids **47**, 995–1007 (2009).
[11] R. R. Prasad and K. R. Sreenivasan, "Scalar interfaces in digital images of turbulent flows," Experiments in Fluids **7**, 259–264 (1989).
[12] C. B. da Silva, J. C. Hunt, I. Eames, and J. Westerweel, "Interfacial Layers Between Regions of Different Turbulence Intensity," Annual Review of Fluid Mechanics **46**, 567–590 (2014).
[13] J. Philip, C. Meneveau, C. M. de Silva, and I. Marusic, "Multiscale analysis of fluxes at the turbulent/non-turbulent interface in high Reynolds number boundary layers," Physics of Fluids **26**, 015105 (2014).
[14] T. Watanabe, R. Jaulino, R. R. Taveira, C. B. da Silva, K. Nagata, and Y. Sakai, "Role of an isolated eddy near the turbulent/non-turbulent interface layer," Physical Review Fluids **2**, 094607 (2017).
[15] X. Wu, P. Moin, J. M. Wallace, J. Skarda, A. Lozano-Durán, and J.-P. Hickey, "Transitionalturbulent spots and turbulentturbulent spots in boundary layers," Proceedings of the National Academy of Sciences **114**, E5292–E5299 (2017).
[16] Y. Zhou and J. C. Vassilicos, "Related self-similar statistics of the turbulent/non-turbulent interface and the turbulence dissipation," Journal of Fluid Mechanics **821**, 440–457 (2017).
[17] K. P. Nolan and T. A. Zaki, "Conditional sampling of transitional boundary layers in pressure gradients," Journal of Fluid Mechanics **728**, 306–339 (2013).
[18] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics **9**, 62–66 (1979), arXiv:arXiv:1011.1669v3.
[19] J. Lee and T. A. Zaki, "Detection algorithm for large-scale structures in turbulent / non-turbulent intermittent flow," (2018).
[20] J. Lee, H. J. Sung, and T. A. Zaki, "Signature of large-scale motions on turbulent/non-turbulent interface in boundary layers," Journal of Fluid Mechanics **819**, 165–187 (2017).
[21] M. J. Hack and T. A. Zaki, "Data-enabled prediction of streak breakdown in pressure-gradient boundary layers," Journal of Fluid Mechanics **801**, 43–64 (2016).
[22] T. Kohonen, *Self-Organizing Maps*, Springer Series in Information Sciences, Vol. 30 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2001).
[23] B. MacQueen, J., "Some Methods for classification and Analysis of Multivariate Observations," Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press **1**, 281–297 (1967).
[24] E. de Bodt, M. Verleysen, and M. Cottrell, "Kohonen maps versus vector quantization for data analysis," in *Proc. ESANN*, Vol. 97 (1997) pp. 211–218.
[25] E. Perlman, R. Burns, Y. Li, and C. Meneveau, "Data exploration of turbulence simulations using a database cluster," Proceedings of the 2007 ACM/IEEE Conference on Supercomputing (SC '07) , 1 (2007).
[26] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, and G. Eyink, "A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence," Journal of Turbulence **9**, N31 (2008), arXiv:0804.1703.
[27] J. Graham, K. Kanov, X. I. Yang, M. Lee, N. Malaya, C. C. Lalescu, R. Burns, G. Eyink, A. Szalay, R. D. Moser, and C. Meneveau, "A web services accessible database of turbulent channel flow and its use for testing a new integral wall model for LES," Journal of Turbulence **17**, 181–215 (2016).
[28] H. Schlichting, *Boundary-Layer Theory*, 7th ed. (McGraw-Hill, Berlin, Heidelberg, 1979).
[29] T. A. Zaki, "From streaks to spots and on to turbulence: Exploring the dynamics of boundary layer transition," in *Flow, Turbulence and Combustion*, Vol. 91 (Springer Netherlands, 2013) pp. 451–473.
[30] F. Bação, V. Lobo, and M. Painho, "Self-organizing maps as substitutes for k-means clustering," Computational Science-ICCS 2005 **3516**, 476 – 483 (2005).
[31] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering **21**, 1263–1284 (2009), arXiv:arXiv:1011.1669v3.
[32] M. J. P. Hack and T. A. Zaki, "Streak instabilities in boundary layers beneath free-stream turbulence," Journal of Fluid Mechanics **741**, 280–315 (2014).
[33] M. A. André and P. M. Bardet, "Velocity field, surface profile and curvature resolution of steep and short free-surface waves," Experiments in Fluids **55**, 1709 (2014).
[34] A. Kushwaha, J. S. Park, and M. D. Graham, "Temporal and spatial intermittencies within channel flow turbulence near transition," Physical Review Fluids **2**, 024603 (2017).
[35] J. You and T. A. Zaki, "The influence of free-stream perturbations on turbulent boundary layers," in *The 12th International ERCOFTAC symposium on engineering, turbulence, modelling and measurements, ETMM12* (2018).