

it is a streak that is about to burst, and the CNN is able to localize and focus its attention on it

Need to put in all formulas of prod, dissipate, Reynolds stresses, etc.

And all formulas of all the operations in the CNN part.

$u = \cdot$  streamwise velocity fluctuations.

Energy: Production vs nonlinear turbulent transport.

Conditional PDFs. Check all email notes. Check txt notes. PDF of size distribution of v blobs and G blobs?

Say - can use in applications like single optimal sensor measurement used to get an idea of 3D structures within the flow field, even far from the wall?

Need to put in all formulas of prod, dissipate, Reynolds stresses, etc.

Need to put in the roller image from the NSF proposal.

And all formulas of all the operations in the CNN part.

# Identifying dynamically important regions in turbulent flows using 3D Convolutional Neural Networks

Eric Jagodinski<sup>1</sup>, Xingquan Zhu<sup>2</sup>, and Siddhartha Verma<sup>1,3†</sup>

<sup>1</sup>Department of Ocean and Mechanical Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA

<sup>2</sup>Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA

<sup>3</sup>Harbor Branch Oceanographic Institute, Florida Atlantic University, Fort Pierce, FL 34946, USA

(Received xx; revised xx; accepted xx)

Near-wall regions in wall-bounded turbulent flows experience intermittent ejection of slow-moving fluid parcels away from the wall and sweeps of faster moving fluid towards the wall. These extreme events play a central role in regulating the energy budget of the boundary layer, and are analyzed here with the help of three-dimensional Convolutional Neural Networks (CNNs). A CNN is trained on Direct Numerical Simulation data from a periodic channel flow to deduce the intensity of such extreme events, and more importantly, to reveal localized 3D regions in the flow where the network focuses its attention to make an accurate determination of burst intensity. These salient regions correlate well with coherent fluid parcels being ejected away from the wall and with regions involving high energy production. The results indicate that CNNs can identify three-dimensional spatial correlations in turbulent flow using a single scalar-valued metric provided as the quantity of interest, which in the present case is the burst intensity. While the current work presents an alternate means of analyzing nonlinear spatial correlations associated with near-wall bursts, the techniques explored here may be used in other scenarios where the underlying spatial dynamics are not known a-priori.

## 1. Introduction

The dynamics of wall-bounded turbulent flows are linked closely to processes that regulate flow behaviour close to the wall. One of the prominent characteristics of the near-wall region is the presence of slow moving wavy streaks of fluid, which intermittently and abruptly lift-up away from the wall, and eject slow moving fluid towards the faster core (Kline et al. 1967; Offen & Kline 1975). These bursts of slow moving streaks have been identified in various experiments using hydrogen bubble visualization (Kline et al. 1967), dye visualization (Kim et al. 1971), and by observing neutrally buoyant colloidal particles (Corino & Brodkey 1969). The ejections are usually followed by sweeps of faster moving fluid towards the wall (Corino & Brodkey 1969), completing the cycle of momentum exchange between the low speed near-wall region and the high speed core.

These intermittent bursts are involved in the generation and dissipation of turbulent kinetic energy within boundary layers. Moreover, they control important transport phenomena and contribute significantly to turbulent drag acting on the wall (Kline et al.

† Email address for correspondence: vermas@fau.edu

1967; Corino & Brodkey 1969; Kim *et al.* 1971; Wallace *et al.* 1972; Lumley & Blossey 1998; Jiménez 2012). Even though the existence of intermittent bursts in wall-bounded turbulence is widely accepted, there has been some ambivalence regarding the role they play in near-wall dynamics. Robinson (1991), Moin & Mahesh (1998), and Schoppa & Hussain (2002) suggested that bursts may not play as crucial a role in turbulence generation as previously thought. The main argument in favour of this viewpoint was that the intermittent events observed by Kline *et al.* (1967) may have been caused by the passage of streamwise vortices over static measurement locations. However, other studies have remarked that these strong intermittent events are not merely artefacts of vortices passing by, but should instead be viewed as intrinsic components of the near-wall dynamics (Jiménez 2012, 2013). Lumley & Blossey (1998) considered bursts to be integral to the formation and evolution of coherent structures, and proposed that the inhibition of bursts should be viewed as a critical component of flow-control strategies. Moreover, Jiménez (2013) found that frictional drag on the wall increased abruptly and substantially during bursting events. Other studies have proposed that instabilities and ejections associated with low-speed streaks may give rise to coherent hairpin vortices (Lozano-Durán & Jiménez 2014; Hack & Moin 2018). A different viewpoint by Schlatter *et al.* (2014) proposes that hairpin vortices may be artefacts of the relatively moderate Reynolds numbers that prior DNSs (Direct Numerical Simulations) had been restricted to due to computational limitations. It is evident that the exact nature of near-wall dynamics is the subject of some debate, which highlights the need for tools that can help better interpret the nonlinear spatial and temporal correlations inherent in turbulent flows.

Experimental diagnostics and simulation capabilities have witnessed steady progress since some of the early studies discussed above. However, a comprehensive understanding of fundamental processes in near-wall turbulence, and more importantly, effective means of influencing them are still being sought actively (Jiménez 2018). A variety of analytical techniques have been explored in this pursuit, with the goal of identifying reduced order phenomena that can accurately describe the dynamical behaviour of turbulent flows. Proper Orthogonal Decomposition (POD), also known as Principal Component Analysis (PCA) (Jolliffe & Cadima 2016) and various other names, is one such dimensionality-reduction technique used in the analysis of turbulent flows (Berkooz *et al.* 1993). Dynamic Mode Decomposition (DMD) (Rowley *et al.* 2009; Schmid 2010; Mezi 2013) is another technique, which is used to extract low order (albeit non-orthogonal) spatiotemporal modes primarily in oscillatory flows. Both these techniques have contributed significantly to our understanding of coherent structures in fluid flows, however, they pose certain limitations when analyzing nonlinear spatiotemporal correlations.

Artificial Neural Networks (ANNs) have helped address some of these limitations, such as by extending POD to identify nonlinear correlations, by using autoencoder networks embedded with nonlinear activation functions (Kramer 1991). Milano & Koumoutsakos (2002) followed this approach to compare the prediction and reconstruction capabilities of nonlinear autoencoders and standard POD in turbulent channel flow simulations. They determined that ANN-based nonlinear POD provided improved compression ability, as well as better reconstruction of near-wall velocity data that was not included in the training set. In a similar approach, Murata *et al.* (2019) employed CNN-based autoencoders to decompose 2D flow around a cylinder at low Reynolds number into its constituent modes. They demonstrated that the use of nonlinear activation functions, which allows neural networks to represent nonlinear functions, resulted in lower reconstruction error compared to the use of standard POD modes. Erichson *et al.* (2020) used a limited set of point measurements to reconstruct the velocity field around a shedding cylinder, the velocity field in homogeneous isotropic turbulence, and the distribution of sea surface

temperature around the globe with the help of an ANN-based decoder. The authors remarked that while the decoder was able to reconstruct the flow and temperature fields successfully for data samples that had been included in the training set, its performance deteriorated noticeably when using time snapshots that were sufficiently removed from the training data. The performance degradation was likely related to overfitting due to the use of very few hidden layers in the study, which highlights the need for deep networks when attempting to recover generalizable representations. One of the goals of the present work is to demonstrate that 3D CNNs with multiple hidden layers can extract salient 3D patterns from turbulent flow fields even when using snapshots that are temporally decorrelated from the training dataset.

Apart from flow reconstruction and low-order mode identification, ANNs have also found use in flow control and turbulence modeling. Early adoption by Fan et al. (1993) explored active control using ANNs to suppress artificial wave-like disturbances in laminar flow using wall-based actuators. Lee et al. (1997) used a turbulent channel flow to train ANNs that employed wall shear measurements to estimate the wall-normal velocity at a normalized distance of  $y^+ = 10$ . The predicted wall-normal velocity was then used to implement blowing-suction control for minimizing skin-friction drag. Upon successfully training the ANN to achieve close to 20% drag reduction, they examined the weight distribution within the trained network to deduce a simplified control scheme which performed comparably well at a lower computational cost. Lorang et al. (2008) followed a similar approach to Lee et al. (1997), but in the Fourier domain instead of physical space. In another extension of the study by Lee et al. (1997), Park & Choi (2020) trained a 2D CNN to predict the wall-normal velocity required to perform opposition control, using pressure and shear stresses at the wall as inputs to the CNN.

In other work related to subgrid scale modelling and predicting macroscale behaviour, Sarghini et al. (2003) explored the ability of ANNs to determine the value of the pointwise eddy viscosity coefficient in Large Eddy Simulation of a turbulent channel flow. Hack & Zaki (2016) employed ANNs to predict transition to turbulence in a spatially developing boundary layer by identifying near-wall streaks that were most likely to breakdown and generate turbulent spots. Ling et al. (2016) used an ANN with a special layer to ensure Galilean invariance of the predicted Reynolds stress anisotropy tensor, which improved its accuracy compared to certain conventional RANS models. Maulik & San (2017) trained a single layer feedforward ANN to deconvolve low-pass filtered turbulent datasets and reconstruct the subfilter scales. Gamahara & Hattori (2017) used ANNs to encode the non-linear functional relation between the resolved flow field and the subgrid stress tensor from filtered DNS data. Upon conducting a-priori tests using DNS data that were not part of the training set, they demonstrated that for relatively small filter widths the functional relation learned by the ANN closely resembled results produced by the gradient model for the subgrid stress.

We remark that a majority of the studies discussed here have relied on ANN architectures that ‘flatten’ the input data into 1D arrays. This poses a significant disadvantage when considering flow data that may contain 2D or 3D spatial features, such as near-wall coherent structures or contiguous regions of high energy production and dissipation. Any such spatial correlations inherent in the data are lost upon flattening the input vector. This suggests that ‘vanilla’ ANN architectures (also referred to as MLPs, i.e., multilayer perceptrons) may not be ideal for analyzing turbulent bursts, since these extreme events are localized in both space and in time (Adrian 2007; Encinar & Jimnez 2020). The velocity at any given point within a burst is strongly correlated to the flow state within a finite neighbourhood in 3D space, which makes 3D CNNs more suitable for analyzing its spatiotemporal evolution. As opposed to vanilla ANNs, the specialized architecture of

CNNs (Fukushima 1980) helps preserve spatial correlations inherent in the input data, and thus, they have been employed in studies of steady (Guo et al. 2016; Sekar et al. 2019) and unsteady 2D (Lee & You 2019) laminar flows. 2D CNNs have also been used to predict unsteady force coefficients associated with bluff bodies (Miyanawala & Jaiman 2018), the pressure distribution on a cylinder (Ye et al. 2020), and drag for arbitrary 2D shapes in laminar flows (Viquerat & Hachem 2020). Furthermore, Fukami et al. (2019), Liu et al. (2020), and various others have explored deconvolution to reconstruct subfilter scales using CNNs in 2D flow sections, based on the super-resolution technique of Dong et al. (2016).

Given the inherent ability of CNNs to identify nonlinear spatial correlations, and the central role of bursting events in regulating the turbulence generation cycle within the boundary layer, we employ a fully three-dimensional Convolutional Neural Network architecture to analyze the bursting process in a turbulent channel flow simulation. The CNN is trained using 3D sections of near-wall velocity data as input, to predict the intensity of intermittent ejection events that appear as the flow evolves in space and in time. We emphasize that the main aim of the present work is not simply to explore another approach for burst detection; a number of techniques developed specifically for identifying turbulent bursts have been surveyed extensively by various authors (Willmarth & Lu 1972; Wallace et al. 1972; Lu & Willmarth 1973; Blackwelder & Kaplan 1976; Kim 1985; Bogard & Tiederman 1986; Lozano-Durn et al. 2012). Instead, the central goal of the present study is to identify spatially localized regions of the flow that the trained CNN focuses its attention on in order to make accurate inferences. This allows us to look beyond the black-box nature of the neural network, and to examine physical processes that CNNs might help identify in wall-bounded turbulent flow fields.

The remainder of the paper is organized as follows. Details regarding the numerical methods and training procedure for the CNN are provided in §2. Results demonstrating the inference and flow-feature identification capabilities of the CNN are presented in §3, followed by detailed analysis and discussion in §4, and concluding remarks in §5.

## 2. Methods

### 2.1. Direct Numerical Simulation

The datasets used in this work were generated via Direct Numerical Simulation of a periodic turbulent channel flow. The incompressible Navier-Stokes equations were solved using a **2nd order central** finite difference scheme and the 2nd order semi-implicit Crank-Nicolson scheme (Desjardins et al. 2008). The flow was driven in the channel by imposing a pressure gradient in the streamwise direction to maintain a constant mass flow rate. The simulation domain and its dimensions are shown in figure 2 for friction Reynolds number  $Re_\tau = u_\tau(L_y/2)/\nu \approx 300$ , where  $u_\tau = \sqrt{\tau/\rho}$  is the friction velocity and  $\tau = \mu\delta u/\delta y$  is the surface shear stress. Periodic boundary conditions were used in the streamwise and spanwise directions, and the no-slip boundary condition was enforced at the top and bottom walls. A stretched cartesian grid was used in the wall-normal direction to resolve the viscous length scales close to the wall. The minimum grid cell height  $\Delta y$  was  $0.03\delta^+$  next to the wall, and the maximum  $\Delta y$  was  $2.4\delta^+$  in the core region, with the cell height stretched using a sin function. Here,  $\delta^+ = \nu/u_\tau$  is the viscous length scale,  $\nu = \mu/\rho$  is the kinematic viscosity, and  $\rho$  is the fluid density. The grid cell sizes were kept uniform in the streamwise and spanwise directions ( $\Delta x = \Delta z = 3.5\delta^+$ ). The mean velocity and rms velocity profiles for two distinct channel flow simulations at  $Re_\tau = 300$  and 670 are shown in figure 3, and compare well with benchmark results from Moser et al. (1999).

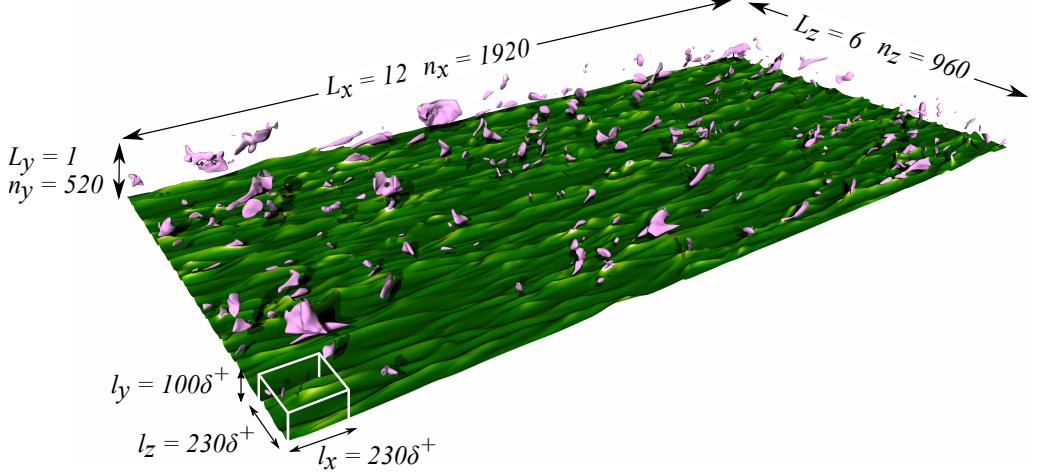


Figure 1: A snapshot of the flow field from a turbulent channel flow simulation at  $Re_\tau = 300$ . The horizontal plane shows an isocontour of the streamwise velocity  $u$ , coloured using the wall-normal velocity  $v$  with brighter shades indicating movement away from the wall and darker shades towards. Low-speed streaks manifest as sinuous ridges, and bright spots mark regions where the flow is being ejected away from the wall. The pink coherent structures denote high intensity ejection parcels where positive fluctuations for  $v$  exceed 2 standard deviations, i.e.,  $v > \bar{v} + 2\sigma_v$ . The white box in the bottom left corner depicts Minimal Flow Unit-sized sections that the snapshots were divided into for training the 3D CNN.

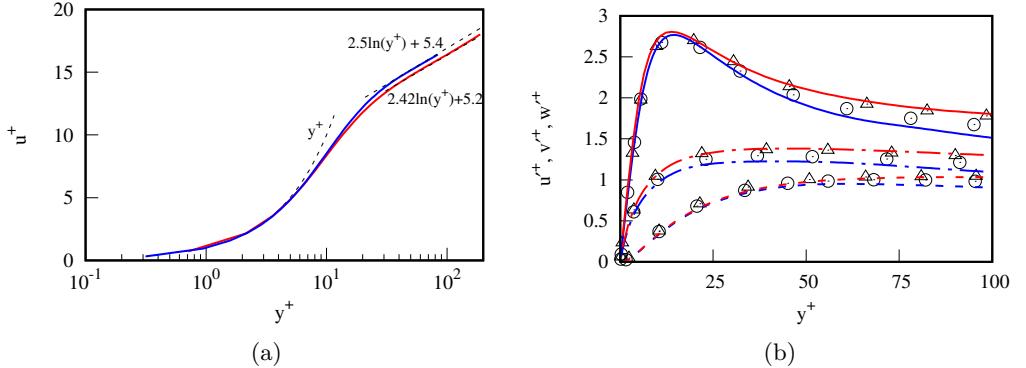


Figure 2: (a) Mean streamwise velocity profile, and (b) rms velocity profiles shown in wall units for  $Re_\tau = 300$  (blue) and  $Re_\tau = 670$  (red). The symbols in (b) correspond to data from Moser et al. (1999) for  $Re_\tau = 395$  and  $590$ .

Once the flow reached a statistically stationary state, time snapshots were recorded at regular intervals with sufficient separation ( $\approx 40t^+$ ) to ensure temporal decorrelation. Here,  $t^+ = \delta^+/u_\tau$  is the viscous time scale. Each full-channel snapshot was then divided into Minimal Flow Unit-sized sections (MFU - Jiménez & Moin (1991)), as indicated by the white box in the bottom left corner of figure 2. Similarly, MFU-sized samples were extracted from the upper channel wall after rotating the wall-normal and spanwise velocities appropriately, so as to maintain the same orientation as the lower wall samples.

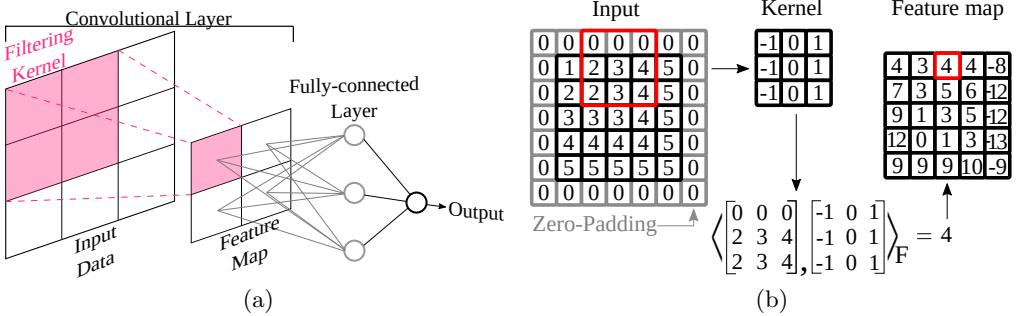


Figure 3: (a) A simplified schematic of a 2D CNN architecture showing a convolutional layer followed by a fully-connected layer that leads to a scalar-valued output. (b) Within a convolutional layer, the filter kernel is convolved across the padded 2D input array, and the Frobenius inner product of these two tensors determines the corresponding values in the ‘feature map’ (i.e. filtered data matrix) of the same size. The appropriate kernel matrix is learned iteratively during training of the CNN.

This procedure resulted in 450 three-dimensional sections (i.e., velocity samples) on each wall from every snapshot, for a total of 10,800 velocity samples from 12 time-decorrelated full-channel snapshots recorded from the DNS.

## 2.2. Convolutional Neural Network

We now provide a brief description of the operations involved in a basic CNN architecture. In a vanilla feedforward ANN with a single layer, the input vector  $I$  may be used to determine a scalar valued output  $\mathcal{O}$  as follows (Higham & Higham 2019):

$$\mathcal{O} = A(b + \sum_i w_i I_i) \quad (2.1)$$

Here,  $w$  is the weight vector,  $b$  is the bias and  $A$  is the activation function. Using nonlinear functions for  $A$ , e.g.,  $\tanh(\cdot)$ ,  $\frac{1}{1+e^{-x}}$  (sigmoid), etc., allows neural networks to encode nonlinear relationships that may exist between the input and output. The weights and biases are unknown at first, and are initialized to random values. To determine the appropriate values of  $w$  and  $b$  that correctly encode the functional relationship between  $I$  and  $\mathcal{O}$ , the ANN is trained via gradient descent:

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \quad (2.2)$$

Here,  $E$  is the defined network error function (e.g. the mean squared error between the true output and the ANN’s predicted output),  $\eta$  is the learning rate, and  $\Delta w$  is the weight adjustment. Every training iteration, values of  $\Delta w_i$  are determined for each neuron based on the current prediction error  $E$ , and using the chain rule to compute the values of  $\Delta w_i$  for each hidden layer. This process is usually referred to as ‘backpropogation’. As the ANN improves in its ability to correctly predict the output  $\mathcal{O}$  via iterative training, the network error decreases and the neuron weights asymptote to their ‘correct’ value. The corresponding bias values are determined in a similar fashion.

Differently from a vanilla ANN, a CNN (Fukushima 1980; Lecun et al. 1998) consists of a series of convolutional layers, pooling layers, and fully-connected or dense layers (figure ??(a)). Figure ??(b) shows a simplified 2D example of a convolutional layer, where 2D weight matrix, also known as a filter kernel is convolved with the input data,

resulting in a 2D filtered output referred to as a feature map:

$$F^k = (f^k * I) \quad (2.3)$$

Here, the filter  $f^k$  is convolved with the 2D input  $I$  to yield the feature map  $F^k$ , i.e., the filtered data corresponding to the ' $k$ 'th filtering kernel. Normally, several distinct filter kernels are associated with a single convolutional layer, and each of these kernels is determined during training using the backpropagation process described previously. As depicted in figure ??(b), zero-padding allows the feature map to retain the same dimensions as the layer's input.

In a CNN, pooling layers usually follow convolutional layers, and help downsample the information passed on to subsequent convolutional layers. These pooling operations have been shown to improve the CNN's ability to encode translational invariance of the most important spatial features present in the input data (Goodfellow et al. 2016). In a 2D CNN, a max pooling layer with a 2:1 ratio will only pass along the highest value from every  $2 \times 2$  square. This would reduce the dimensionality of the data by a factor of four after every pooling operation in two dimensions.

The 3D CNN architecture used in the current work is shown in figure 5, with four alternating convolutional and pooling layers, followed by flattening into two dense layers connected to the final output value. In our specific case, the primary function of the convolutional and pooling layers is to extract 3D features from the flow data, whereas the fully connected layers towards the end learn to associate the resulting assortment of feature maps with the corresponding ejection intensity value. In the first convolutional layer, the  $[30 \times 40 \times 30]$  input is convolved with 32 distinct  $[3 \times 3 \times 3]$  3D filter kernels. This is followed by a max pooling layer, resulting in 32 distinct three-dimensional feature maps, each of size  $[15 \times 20 \times 15]$ . The convolution-pool operation is repeated over the subsequent convolutional layers, leading to a final assortment of 256 feature maps of size  $[1 \times 2 \times 1]$ . These are flattened and directed into a dense layer of size 128, which connects to another dense layer of size 24, finally leading to the scalar valued inferred output.

### *2.3. Preparation of 3D velocity samples*

For training the CNN to infer ejection intensity values, each of the 10,800 MFU-sized samples were 'labeled' by calculating their corresponding intensities a-priori. This was done by computing the percentage of cells within the sample where the wall-normal velocity  $v$  exceeded 2 standard deviations at the wall-normal distance corresponding to the cells' location, i.e.,  $v_{(x,y,z)} > \bar{v}(y) + 2\sigma_{v(y)}$ , where  $y$  is the wall-normal distance. This metric provides a useful indication of ejection activity within each velocity sample, and identifies areas associated with strong velocity fluctuations without having to rely on adjustable parameters. We note that while there are other widely used methods for classifying bursts and sweeps, for instance, the quadrant method introduced by Wallace et al. (1972) and Lu & Willmarth (1973), the aim of the current work is not merely to employ CNNs to identify bursts. Instead, the goal is to demonstrate that CNNs can identify coherent 3D regions that influence a specified quantity of interest, without any a-priori knowledge of the underlying dynamics. The approach proposed here considers ejections to be associated with large deviations in the wall normal velocity  $v$ , which conforms to the viewpoint by Kline et al. (1967) of associating bursts with strong intermittent events. After labeling, each velocity sample was interpolated down from its original size of  $64 \times 40 \times 64$  cells onto a grid of size  $30 \times 40 \times 30$  with uniformly spaced cells in the wall-normal direction, and converted to half-precision floating point numbers to conserve memory required for training.

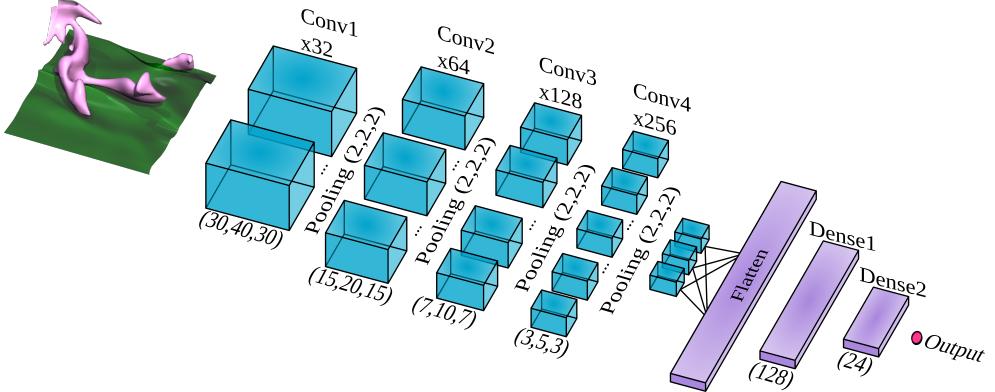


Figure 4: The Convolutional Neural Network takes a 3D sample of the turbulent flow field as input, and infers the ejection intensity as the output based on the wall-normal velocity component. The architecture consists of 4 convolution + pooling layers, which identify and extract the most important flow features from the data. The 3D data is then flattened, followed by two fully-connected layers terminating in the output node. The number of distinct filtering kernels used at each convolution layer ( $\times 32$ ,  $\times 64$ , ...), and the layer sizes ((30, 40, 30), ...), are shown in the figure. Altogether, there are approximately 1.2 million unknown parameters (weights and biases) that must be learned during training.

#### 2.4. Training the CNN

After labeling and interpolation, the 10,800 MFU-sized velocity samples were split randomly into 85% (9,180) training, 7.5% (810) validation, and 7.5% (810) test sets. The training samples were fed in batches of five to the CNN as input, along with their corresponding labels, i.e. the pre-calculated ejection intensities. We note that only the wall-normal velocity  $v$  was used as input for training, as it is the component most closely related to ejection events. The CNN architecture and training processes were implemented using the open source library Keras, with TensorFlow as its backend (Chollet et al. 2015; Abadi et al. 2015). The relevant source code is provided as part of the supplementary materials. The loss-function (i.e. network error function  $E$ ) was defined as the percentage error between the output value calculated by the CNN and the actual label for each sample. The weights and biases were updated using the Adam optimizer (?) to minimize this loss value during training.

To achieve the best possible accuracy for an ANN, it is often necessary to conduct multiple training runs to determine the ideal set of hyperparameter values which control the network's architecture and training process. In the present work, the CNN architecture and training procedure were optimized through a series of hyperparameter sweeps where several combinations of hyperparameter values were tested. The first sweep was conducted to determine the appropriate number of layers and filters as well as the layer-sizes and filter-sizes that resulted in the highest training accuracy. A second sweep was then conducted through the rest of the hyperparameters, i.e., the learning rate, number of epochs, batch size, and other parameters listed in Table 1. The training accuracy was determined by comparing the CNN-computed output to actual labels, i.e., the ground-truth, for velocity samples taken from a separate full-channel snapshot which was time-decorrelated from the training data. The resulting optimal combination of hyperparameters that yielded the highest accuracy is shown in Table 1.

Hyperparameters	Parameters/Values	Hyperparameters	Parameters/Values
Kernel Size	3x3x3	Batch Size	5
Pooling Size	2x2x2	Epochs	57
Weight Initialization	He uniform	Dropout	0.5
Bias Initialization	Zeros	Learning Rate	0.0001
Loss Function	Percent Error	Decay	0.0001
Optimization	Adam	Activation Function	ReLU

Table 1: Parameters and hyperparameters related to the CNN architecture and training.

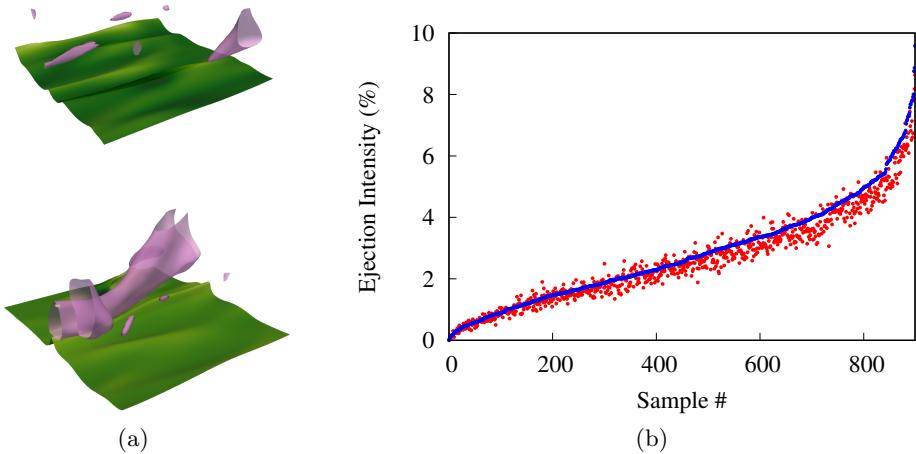


Figure 5: (a) Two velocity samples that were not included in training or validation. The pink cohesive structures correspond to high intensity ejection packets computed as  $v > 2\sigma_v$ , and the horizontal planes depict isocontours of  $u = 0.5 * U_0$ , where  $U_0$  is the centerline velocity. The isocontours have been coloured by  $v$ , with brighter shades indicating movement away from the wall and darker shades indicating movement towards it. The pre-computed labels (i.e., ground truth intensities) for the two data samples are 0.284% (top) and 4.751% (bottom), whereas the values inferred by the CNN are 0.282% and 4.749%, respectively. (b) Comparison of the ground-truth labels (blue dots) and the network-inferred ejection intensities (red dots). The data is shown for an out-of-sample snapshot which is time-decorrelated from the training dataset.

### 3. Results

#### 3.1. Inference accuracy of the trained CNN

With the optimal hyperparameter values provided in table 1, the loss error  $E$  computed on the validation set decreased from over 100% at the onset of training to under 10% within 60 epochs (i.e., iterations over the training set). This took approximately one hour on an Nvidia Titan RTX graphics card with 4608 CUDA cores and 24 GB of GDDR6 VRAM. The CNN was then provided with 3D velocity samples from a time-decorrelated snapshot as input (i.e., separated by at least  $40t^+$  from the datasets used during training). The resulting CNN-inferred ejection intensities for the time-decorrelated samples are shown in figure 6. Two of the velocity samples are visualized in Figure 6a,

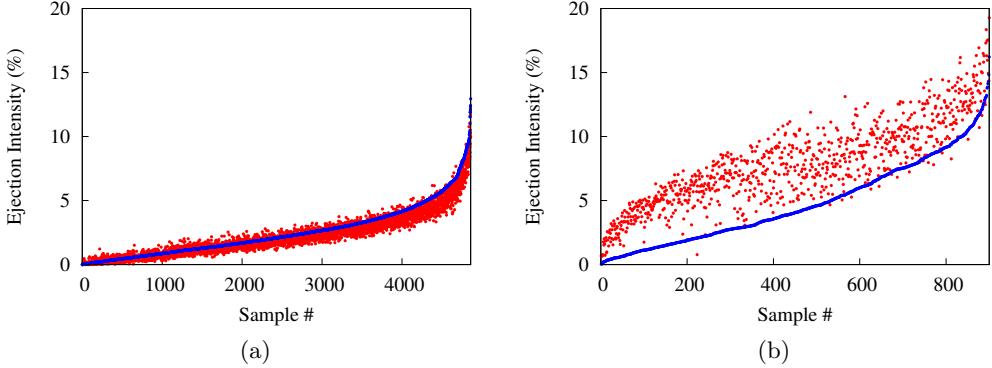


Figure 6: ?? Inference for  $Re_\tau = 670$  data using a CNN trained on the  $Re_\tau = 300$  database. The trained CNN is able to maintain a mean absolute percent error of less than 25% despite having been trained on a lower  $Re_\tau$  dataset. (b) Inference for artificially generated samples with the same first and second-order statistics as the DNS samples used in figure 6. The CNN yields a mean absolute percent error of over 130%, demonstrating that the network does not just rely on statistics for inferring the ejection intensity.

whose ejection intensities (i.e., labels) computed using the thresholding described in §2.3 were 0.284% (top) and 4.751% (bottom). When these two samples were fed in as inputs to the 3D CNN, the ejection intensity values inferred by the network as output were 0.282% and 4.749%, respectively, which correspond very well with the ground-truth labels. Overall, for 900 distinct time-decorrelated velocity samples tested (figure 6b), the mean relative error computed using the absolute difference between the labels and the network-inferred values was less than 10%. Figure 6b also indicates that the network-inferred outputs match the ground-truth labels quite well, regardless of whether the samples are associated with high intensity ejections or low-activity samples.

To determine how well the CNN trained with data from the simulation at  $Re_\tau = 300$  generalizes to a different flow condition, we examine the network's inference ability using velocity samples from a different channel flow simulation at  $Re_\tau = 670$ . The characteristics of near-wall coherent structures (e.g., their size) are known to scale in viscous units (?). Thus, the physical dimensions of the new velocity samples were kept identical to those of the  $Re_\tau = 300$  samples in wall units, and the velocity components were rescaled by multiplying with  $u_{\tau 300}/u_{\tau 670}$  which are the respective friction velocities. The resulting comparison between pre-computed labels and network-inferred values are shown in figure 7??, where we observe that despite a notable difference in the Reynolds number, the network is able to make inferences with a mean absolute percentage error of approximately 24%. While the agreement between the network's inferred output values and the ground-truth labels is not as close as that observed in figure 7(b), it still indicates the ability of CNNs to identify and extract important physical processes that may persist across dissimilar flow conditions.

We recall that the ground-truth labels used during training were determined by computing the first and second order statistics of the wall normal velocity, i.e.,  $\bar{v}_y$  &  $\sigma_v(y)$ . To determine whether the trained network relies on these specific statistical characteristics for inferring the ejection intensity, we now examine the network's performance using artificially generated velocity samples. These artificial samples were populated using

Gaussian noise with the same mean and variance ( $\bar{v}_y$ ,  $\sigma_v(y)$ ) at a given wall-normal distance as that of the DNS samples used in figure 6b. After processing the artificial samples using the CNN, the network-inferred output values were compared to the pre-computed ground-truth labels, and the resulting comparison is shown in figure 7(b). Overall, the mean absolute percent error for these artificially generated samples was over 130%, demonstrating that the network does not merely rely on the samples' statistics, but instead may be relating ejection intensities to cohesive spatial features present in the data. Spatial feature extraction is an important capability of CNNs, which makes them distinct from traditional statistical approaches, and provides an opportunity for identifying dynamically important coherent structures that may be present in the flow.

### 3.2. Examining the CNN's focus on cohesive 3D regions in the flow

Certain studies have determined that the flow structures that govern near-wall momentum transfer are transient in time and localized in space (Lozano-Durán & Jiménez 2014; Encinar & Jiménez 2020). It is often necessary to use some form of user-prescribed spatial filtering when studying the coupled energetics of such structures which may form at disparate length scales. CNNs provide an alternative means of extracting these spatial structures by virtue of their autonomous feature identification capability. Autonomous identification of the most critical flow regions and cohesive structures can prove to be especially useful when considering scenarios where the underlying dynamics are not known a-priori.

There are various techniques that have been developed for interpreting how trained networks correctly relate inputs to the corresponding outputs (Simonyan et al. 2013; Zhou et al. 2015; Selvaraju et al. 2016; ?). These techniques help overcome the black box nature of neural networks which on their own provide no indication of the underlying physics. Exploring these techniques can yield valuable insight into whether the CNN learns to correlate dynamically significant cohesive regions to the inferred burst intensity. Some common methods for interpreting a trained CNN involve inspecting the filter kernels associated with every convolutional layer, or examining the feature maps, i.e., the convolved data obtained after each filtering operation during forward propagation. We note that both these approaches can result in hundreds of outputs to analyze, and yet provide little insight into the trained CNN's input-output correlation. Saliency maps were created by Simonyan et al. (2013) to address this shortcoming, and provide a visual representation of the inferred output's sensitivity to slight changes in the input data. These maps are generated by measuring the change that results in the output value due to small perturbations introduced at each coordinate point in the input data. The normalized gradient of the output with respect to each input data point is calculated as follows:

$$S_{(x,y,z)} = \frac{\partial \mathcal{O} / \partial I_{(x,y,z)}}{\max(\partial \mathcal{O} / \partial I)} \quad (3.1)$$

Here,  $\mathcal{O}$  represents the inferred output which in our case is the burst intensity value,  $I$  is the input, i.e., a 3D velocity sample, and  $S$  is the saliency value computed at coordinates  $(x, y, z)$ . While the resulting saliency map indicates which input data points influence the inferred output the most, the resulting salient regions resemble a scattered set of points rather than a cohesive region for a given sample (e.g., figures 9(b) and 9(c)).

Class Activation Map (CAM) (Zhou et al. 2015) is another interpretation technique which was developed to improve localization compared to saliency maps. CAM averages the feature maps resulting from the last layer for each kernel followed by a single dense layer. The corresponding dense layer weight for each filter kernel of the final convolutional

layer acts as an ‘importance score’. The activation map is then generated by summing the feature maps of each filter weighted by this importance score. Two major drawbacks associated with using the CAM approach are that it requires the network architecture to include a Global Average Pooling layer at the end of the convolutional layers, and that it only allows for one dense layer in order to use its weights as the importance scores. These limitations allow only specific network architectures to be compatible with CAM, making it a less generalizable technique.

Gradient-weighted Class Activation Map, or GradCAM (Selvaraju et al. 2016), is another interpretation technique which attempts to address the issues encountered by both saliency maps and CAM. A schematic overview of the steps involved in computing the GradCAM output gradient,  $G_n$ , is shown in figure 9(a). GradCAM averages the gradients of the output with respect to each feature map of the last convolutional layer, to determine the corresponding filter’s importance score  $a^k$ :

$$a^k = \frac{1}{n_x n_y n_z} \sum_x \sum_y \sum_z \frac{\partial \mathcal{O}}{\partial F_{(x,y,z)}^k} \quad (3.2)$$

Here,  $F_{x,y,z}^k$  is the feature map of filter  $k$  in the final convolutional layer, and  $a^k$  is the neuron importance weight. This importance score is then used to weight and sum the feature maps of the final convolutional layer:

$$G_{(x,y,z)} = \text{ReLU} \left( \sum_k a^k F_{(x,y,z)}^k \right) \quad (3.3)$$

Here,  $G$  is the gradient importance score, i.e., the GradCAM gradient computed at coordinate  $(x, y, z)$ . The weighted sum is transformed using a Rectified Linear Unit (ReLU) which zeros out negative argument values, thereby selectively retaining information that positively influences the inferred output. In our case, this implies that a greater emphasis is placed on grid cells that contribute to higher bursting intensity for the sample under consideration.

As a simplified visual example of how some of the interpretation techniques compare to one another, figures 9(b), 9(c), and 9(d) show the input image, the saliency map, and the GradCAM for an image-classification network that has been trained to discern cats and dogs. Both the saliency map and GradCAM focus on the ears, eyes, and collar in order to identify the picture as that of a dog. But, as observed in figure 9(c), the saliency map resembles a scattered set of points, whereas the GradCAM in figure 9(d) clearly highlights cohesive regions of the input image. While the saliency map does pick out similar areas, the GradCAM is better at identifying the regions more precisely. This demonstrates the difference between the two interpretation methods, and serves as the reason for using GradCAM in the present work to identify salient structures in the near-wall region.

### 3.3. Multi-Layer GradCAM

One downside of GradCAM is that the resulting activation map is only the size of the feature map from the last convolutional layer. In most CNNs, there are pooling layers in between some or all convolutional layers to downsample the data meaning that the last feature map of the last convolutional layer as well as the resulting GradCAM are a significantly lower resolution than the input data and needs to be interpolated up to match the input. In order to address this and construct a higher resolution activation map, we summed the GradCAMs with respect to each convolutional layer. A similar procedure was used in ?. As shown in figure 9(a), the GradCAM gradient from each convolutional layer was summed after being interpolated to the resolution of the input

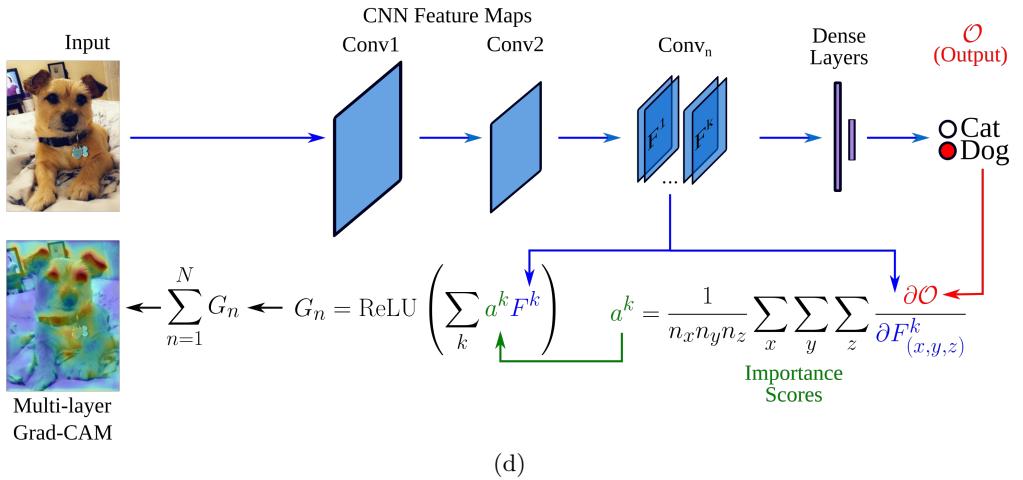
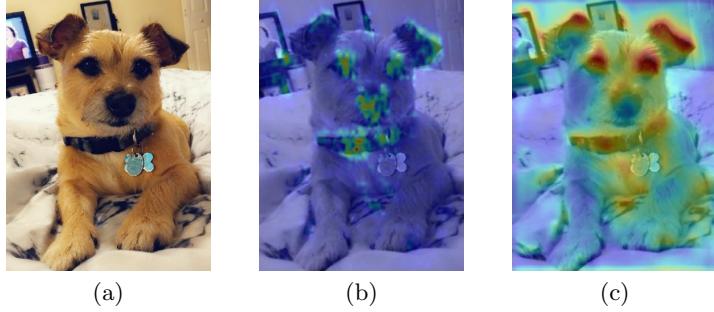


Figure 7: (b) Input image and corresponding (c) saliency map and (d) GradCAM gradients from a CNN trained to discern between cats and dogs. The green, yellow and red areas depict the salient regions which most influence the CNN’s inference, namely, the ears, the eyes and the collar. (a) A simplified 2D example of the multi-layer GradCAM interpretation technique for producing visual explanations of CNN decisions using a network with a similar arrangement to figure 5. The test image was first input into the network to obtain the output and the feature maps, the filtered data, of each layer for every filter. Each feature map was averaged by pixel values to produce a filter importance score. The product of these importance scores and their respective feature map was put through a final ReLU to focus on cells that increase the inference accuracy for the GradCAM of that layer. Finally, the GradCAM activation map is interpolated to match the dimensions of the input image for the output. For the multi-layer GradCAM method, the GradCAM gradients of each convolutional layer is summed.

data. This addresses some of the issues with resolution differences between input data and the traditional GradCAM gradient. As well this activation map now contains information from lower convolutional layers.

### 3.4. Spatially Localized Salient Regions

Using the modified GradCAM technique described in §??, we now examine the localized flow features that the CNN has learned to focus its attention on. Figure 8?? shows

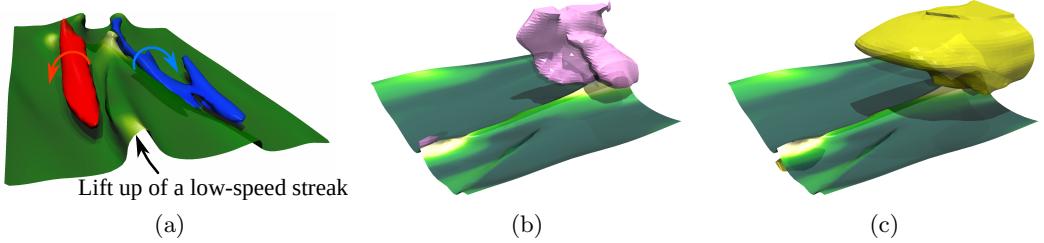


Figure 8: ?? A low-speed streak undergoing bursting with counter-rotating streamwise rollers on either side. The green horizontal plane represents an iso-contour of the streamwise velocity component colored by wall-normal velocity. The red and blue coherent structures are Q criterion iso-surfaces of vorticity. ?? Velocity sample showing high-intensity ejection packets in pink. A white bursting streak is visible in the rear part of the  $u$  isocontour. ?? The corresponding Grad-CAM image obtained from the trained CNN. The golden structures depict localized regions in the flow that have the most significant influence on the CNN’s inference of the sample’s ejection intensity.

a low-speed streak undergoing bursting. The green iso-contour is of streamwise velocity and coloured by wall-normal velocity while the red and blue coherent structures are Q criterion iso-surfaces of vorticity showing the rollers. On either side of the lifting streak are counter rotating streamwise rollers. The brighter shades of the isocontour indicate movement away from the wall which correspond to ejection events. describe roller image here and how they correspond to the lifting up burst, contour color, etc. Figure 8?? shows the visualization of a velocity sample that is provided as input to the trained CNN, and figure 8?? shows the corresponding GradCAM gradient obtained from the CNN for this particular sample. The cohesive pink structures in figure 8?? represent pre-computed regions of high ejection intensity, where positive fluctuation in  $v$  exceed two standard deviations (similar to figure 2). We also observe a bursting streak underneath the region of high ejection intensity, denoted by a brightly coloured ridge in the streamwise velocity contour plane. From the corresponding GradCAM image shown in figure 8??, we observe that the CNN focuses its attention on the ejection parcel and the bursting streak together in order to determine the sample’s ejection intensity. This manifests as the golden GradCAM structures occupying the same regions as the pink ejection packets, and reaching down to the bursting streak visible on the contour plane. Although qualitative in nature, this is a notable outcome, especially since the CNN was provided with no a-priori knowledge of the flow patterns related to near-wall ejection. Instead, this ability was gained autonomously by the CNN by training on velocity samples that were assigned a single scalar metric as the corresponding label, i.e., the ejection intensity.

We now examine the ability of the CNN to track the salient regions as the flow evolves in time. Figure 11 shows successive snapshots at a fixed spatial location within the channel, with ejection parcels and bursting streaks superimposed with the GradCAM structures. At  $t_0$ , the GradCAM structures are able to identify three distinct ejection packets, as well two bursting streaks towards the left (upstream) and right (downstream) edges of the selected region. One viscous time unit later, i.e. at  $t_0 + t^+$ , the CNN considers the larger ejection packet entering the field of view from the center-left to be more important to its inference of ejection intensity, and focuses less on the outgoing ejection packet that has started dissipating near the lower right edge. At this instant, the ejection packet towards the top right and the bursting streak near the bottom right edge are still influential in

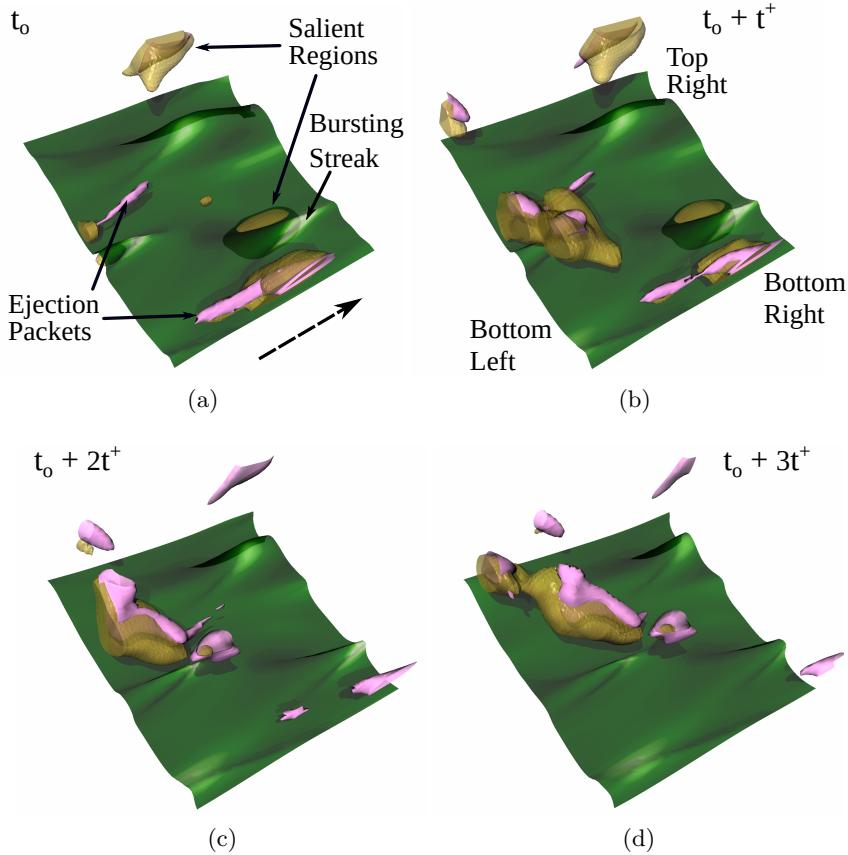


Figure 9: Four successive time instances showing an overlay of the GradCAM gradient over the corresponding flow field (animation available in Supplementary Movie 1). In (a) the CNN focuses its attention on ejection parcels that are already well formed, as well as on the streak that is undergoing bursting in the bottom right corner. (b) As a new ejection parcel enters the field of view from the left, the CNN includes it as part of the salient regions. (c), (d) As the bursting streak and ejection parcels move out of the field of view, the CNN switches its attention to the strong ejection parcel developing on the left side.

the CNN’s inference. At  $t_0 + 2t^+$  and  $t_0 + 3t^+$ , the large ejection packet that has entered the field of view is considered by the CNN to be the dominant structure for inferring the sample’s ejection intensity. These results indicate that the CNN is able to adaptively modulate its focus on the most critical features as the flow evolves in time.

### 3.5. Identifying Geometric Centers of Cohesive Structures

In order to quantify how well the GradCAM based salient regions correspond to ejection packets, we first identify cohesive geometric structures that represent localized regions of interest for  $v$  and  $G$ . As can be observed in figures 8 and 11, the spatial distribution and shape of such geometric structures tend to be very irregular. Thus a two-step process based on the use of Radial Basis Functions (RBF) was devised. This involved first identifying the potential centers of such structures as the ‘center of mass’ based on the

values of  $v$  and  $G$ , then determining the structures' spatial extent in three dimensions. We note that such geometric structures were analyzed specifically in the log-law region of the flow, i.e., between  $y^+ = 30$  and 100. In the first step, all cells that exceeded  $2\sigma_p$ , where  $\sigma_p$  represents the standard deviation of the parameter of interest, which may be  $v$ ,  $G$ , or any other quantity of interest  $p$ , were identified as potential geometric centers. The 'center of mass' for the parameter was then calculated within a spherical region of radius  $15^+$  units, and the geometric center was shifted to this point. The sphere's radius was then increased to  $45^+$  units, and the new center of mass was calculated. If the center of mass changed between the two calculations then the process was continued iteratively using the two radii until either there was no further change, or if the same point had been visited at least three times. After the potential geometric centers were established, the next step involved determining the structures' spatial extent, both to distinguish between neighboring structures that may be close together, and to determine whether the structure identified was large enough to be considered significant. Starting with a radius of  $15^+$  units, a Gaussian weighted average based on radial distance from the sphere's center was calculated as follows:

$$w_p = e^{-r^2/2\sigma^2} \quad (3.4)$$

Here,  $r$  is the radial distance from the center of mass,  $\sigma = R/3$  where  $R$  is the radius of the spherical region (e.g.,  $15^+$  units for the initial sphere), and  $w_p$  is the resulting weight used to calculate the Gaussian weighted average. The radius was then increased by 50% and the Gaussian weighted average was calculated again. This process was continued until the weighted average decreased by over 25% between consecutive iterations, indicating that the expanded region had likely exceeded the extent of the cohesive structure. If the radius increased at least three times during the iterative expansion, then the region being examined was considered to be large enough to potentially constitute a geometric feature of interest. After one pass of this procedure was conducted, the 'mass' density for each geometric structure was computed and their average density used for an additional requirement to be a geometric feature of interest. A second pass of the same procedure was conducted using 75% of the average density as an additional cutoff within the expansion step to ensure that only geometric features of a relevant density were kept.

The procedure outlined above was used to identify geometric centers of prominent structures associated with wall-normal velocity fluctuations,  $v$ , and the GradCAM gradient,  $G$ , as well as other important quantities such as energy production and dissipation. Figure 10 shows a visual example of the geometric center identified for a prominent  $v$ -structure and a neighboring  $G$ -structure, with the RBF spheres overlayed. The Euclidean distance between these geometric centers can be used to quantify how well the three dimensional structures for  $v$ ,  $G$ , and any other relevant parameters  $p$ , correlate with each other.

### 3.6. Salient Structures, Ejection Regions, and the Energy Budget

To examine how well the salient regions identified by the CNN correlate to bursting and the evolution of turbulent kinetic energy (TKE), we identified the relevant geometric structures using the procedure outlined in §???. The three main components of the energy budget were selected for this purpose, namely, energy production, dissipation and transfer. These terms are part of the average TKE equation, which can be obtained from the momentum equation as follows. Using Reynolds decomposition, the velocity can be expressed as  $u = \bar{u} + u'$ , where  $\bar{u}$  involves a generic averaging operation in space or time and  $u'$  is the fluctuating part. The momentum equation for the mean velocity can

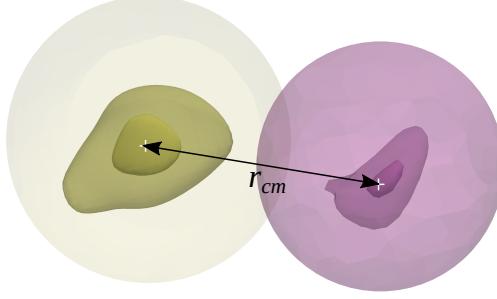


Figure 10: Examples of cohesive structures associated with high wall-normal velocity (pink) and the GradCAM gradient (gold). The outermost contour shows the spherical RBF bounds determined by the procedure described in §?? whereas the two inner contours correspond to  $2\sigma_p$  and  $3\sigma_p$  for the respective parameters,  $p$ . The centers of mass computed for the two structures are shown using the + symbol and the Euclidean distance between them is denoted as  $r_{cm}$ .

then be obtained by averaging the momentum equation:

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} + u'_j \frac{\partial u'_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{P}}{\partial x_i} + 2\nu \bar{S}_{ij,j} \quad (3.5)$$

Here,  $S_{ij}$  is the strain rate tensor,  $P$  is the pressure and  $\rho$  is the density. Subtracting equation ?? from the momentum equation gives:

$$\frac{\partial u'_i}{\partial t} + u'_j \frac{\partial u'_i}{\partial x_j} + \bar{u}_j \frac{\partial u'_i}{\partial x_j} + u'_j \frac{\partial \bar{u}_i}{\partial x_j} - \frac{\partial(u'_i u'_j)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P'}{\partial x_i} + 2\nu S'_{ij,j} \quad (3.6)$$

and multiplying equation ?? by  $u'_i$  yields the TKE equation:

$$\frac{\partial u'_i u'_i / 2}{\partial t} + u'_j \frac{\partial u'_i u'_i / 2}{\partial x_j} + \bar{u}_j \frac{\partial u'_i u'_i / 2}{\partial x_j} + u'_i u'_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial(P' u'_i)}{\partial x_i} + 2\nu(u'_i S'_{ij})_{,j} - 2\nu S'_{ij} S'_{ij} + u'_i \frac{\partial u'_i u'_j}{\partial x_j} \quad (3.7)$$

The transport equation for the average TKE can be obtained by taking the average of equation ??:

$$\frac{\partial k}{\partial t} + \bar{u}_j \frac{\partial k}{\partial x_j} + \frac{\partial}{\partial x_j} \left[ \frac{1}{2} (\overline{u'_i u'_i u'_j}) - 2\nu \overline{u'_i S'_{ij}} \right] + \frac{1}{\rho} \frac{\partial P' u'_i}{\partial x_i} = -\overline{u'_i u'_j} \frac{\partial \bar{u}_i}{\partial x_j} - 2\nu \overline{S'_{ij} S'_{ij}} \quad (3.8)$$

Where,  $k = \frac{\overline{u'_i u'_i}}{2}$ . Here, the  $-\overline{u'_i u'_j} \frac{\partial \bar{u}_i}{\partial x_j}$  term contributes to the production of average TKE due to the contribution of ??, whereas  $-2\nu \overline{S'_{ij} S'_{ij}}$  contributes to viscous dissipation. The divergence terms on the left hand side are responsible for the transfer of energy, with no overall contribution increase or decrease. After determining the geometric centers for cohesive structures related to ejections, energy production, dissipation, transfer and the GradCAM-based salient regions, the Euclidean distance ( $r_{cm}$ ) between these structures were computed to assess their spatial proximity to each other. The Probability Density Functions (PDF) of  $r_{cm}$  for the selected quantities of interest are shown in figures 12 and 13. Figure 12 provides an indication of the spatial proximity between cohesive structures related to ejection regions, production, and dissipation. We observe a tendency for the energy related structures to be found in close proximity to the ejection related structures, as indicated by high PDF values for smaller  $r_{cm}$  values. The most notable

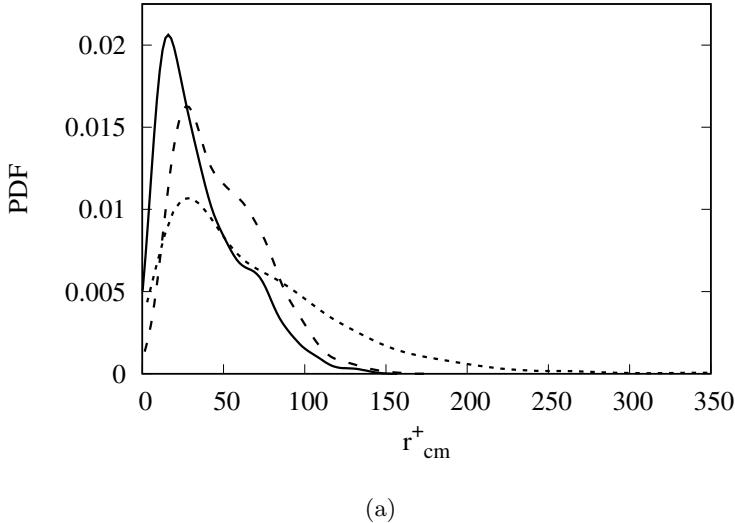


Figure 11: Probability Density Function (PDF) of the Euclidean distance  $r_{cm}^+$  between the nearest cohesive structures for two selected quantities of interest. Production to velocity (—), dissipation to velocity (- - -), and production to dissipation (.....). In the velocity proximities, there is a notable relationship in the parameters with production-velocity having a peak at  $17^+$  while the dissipation-velocity peak is at  $27^+$

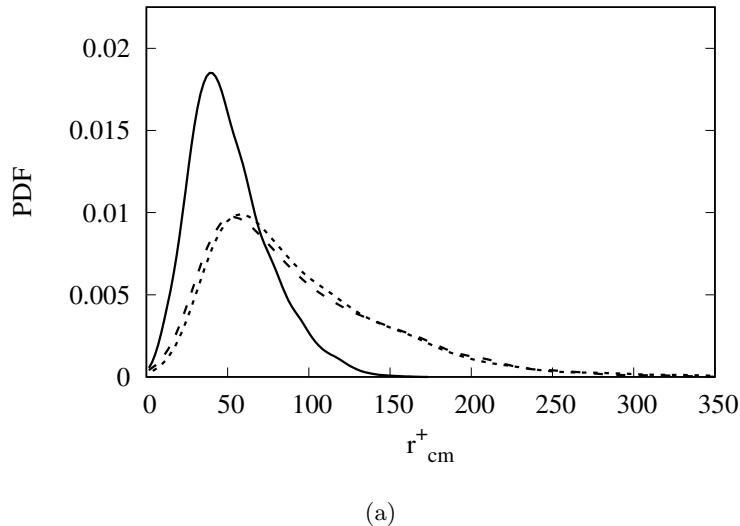


Figure 12: PDF of the Euclidean distance  $r_{cm}^+$  between the nearest cohesive structures for GradCAM gradient and a quantity of interest. Gradient to velocity (—), gradient to production (- - -), and gradient to dissipation (.....). The Gradient to velocity PDF has the most notable correlation with a peak corresponding to  $37\delta^+$ .

correspondence is that between clusters of high energy production and high wall-normal velocity fluctuations, for which the PDF peaks at an  $r_{cm}$  value of  $17\delta^+$ , which is indicative of the most probable Euclidean distance between cohesive structures for the two quantities. The observed correspondence is expected, since the energy production equation has a direct dependence on wall-normal velocity fluctuations, when simplified for the channel flow scenario it follows equation ???. Here, we used the fact that because of the periodic boundary conditions in the streamwise and spanwise directions, only the mean streamwise velocity will vary in the wall-normal direction therefore  $\frac{\partial \bar{u}}{\partial y} \neq 0$  in the production term, thus energy production  $\mathcal{P} \approx -u'v'\frac{\partial \bar{u}}{\partial y}$ . The PDFs of  $r_{cm}$  computed between dissipation-ejection and production-dissipation structures also show a similar but lower peak at a Euclidean distance of approximately  $27\delta^+$ .

Figure 13 compares the spatial proximity of these ejection regions, and production and dissipation cohesive structures to the salient regions of GradCAM gradient. The strongest spatial relationship as anticipated is between the GradCAM gradient concentrations and high wall-normal velocity fluctuation, with a peak at a Euclidean distance of approximately  $37\delta^+$ . Since the CNN was trained using only bursting intensity as the metric, it is expected that the GradCAM gradient resulting from the trained network would have a high correlation to the wall-normal velocity fluctuations. The authors see this as confirmation of the significant correlation between the two as similar the relationship in proximity between physically derived parameters such high wall-normal velocity and production.

### 3.7. Cross-Correlation of Parameters of Interest

**New-subsection to be filled.** In order to further investigate the relationship between GradCAM gradient and other parameters of interest, the cross-correlation was calculated by radial distances . Figure ?? shows the auto-correlation and cross-correlation of several physical parameters of interest for comparison. On both, the correlation tends to zero by  $100^+$ . Figure ???? shows the cross-correlation of velocity fluctuations  $u'$  and  $v'$  which have a high negative correlation at close distances which can be attributed to their relationship in bursting events. Figure ???? shows the cross-correlation of velocity fluctuations with kinetic energy production. Production and  $u'$  &  $v'$  have a notable correlation at close distances as well with higher values of production being associated with higher wall-normal and lower streamwise velocity fluctuations like during ejection events. Figure ???? is the cross-correlation of GradCAM gradient and velocity fluctuations and it shows a similar relationship. Higher values of GradCAM gradient correlating at close distances to the higher wall-normal and lower streamwise velocity fluctuations associated with ejection events.

### 3.8. Spatial Distribution of the GradCAM Value and Parameters of Interest

To provide a better idea of the distribution of the GradCAM value  $G$  within the sample volumes, we examine the Complementary Cumulative Distribution Function (CCDF) of  $G$  in Fig 14a. The CCDF has been computed by determining the volume fraction occupied by values of  $G$  greater than or equal to a given value, averaged across 900 samples, and depends on the Cumulative Distribution Function (CDF) as follows:

$$CCDF(G) = Pr(\mathcal{G} \geq G) = 1 - CDF(G) \quad (3.9)$$

Here,  $\mathcal{G}$  represents the random variable for  $G$ . Fig 14a indicates that approximately 30% of a sample's volume on average is occupied by values of  $G$  greater than or equal to 1, and there is a steep drop for larger values; only about 4% of the sample volume is occupied

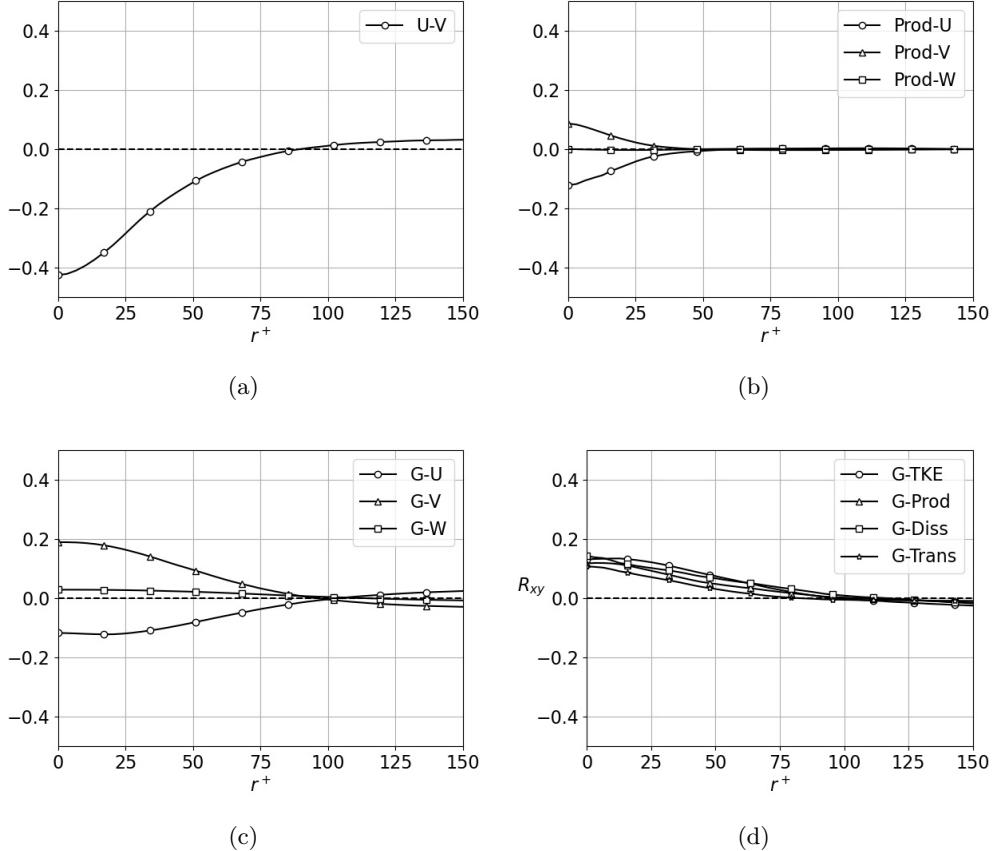


Figure 13: Temp images. Auto-correlation and cross-correlation plots of various parameters of interest with a radial offset. ?? and ?? are the cross-correlation of velocity fluctuations  $u$  and  $v$  ?? and production to each velocity fluctuation ???. ?? is the cross-correlation of GradCAM gradient and the velocity fluctuations  $u$ ,  $v$ ,  $w$ . GradCAM gradient and  $v$  show a particularly good correlation. ?? is the GradCAM gradient with various physical parameters of interest. All have about the same middling correlation.

with values of  $G \geq 2$  and 0.5% for  $G \geq 3$ . This suggests that the GradCAM is sparsely distributed within the sample volumes, which in turn implies that the intensity of the CNN's focus is highest in very localized spatial regions for making an inference of the burst intensity. We note that the magnitude of  $G$  can vary with ??? and that the values presented here have not been normalized in any way.

To determine what fraction of a specific quantity of interest  $\mathcal{P}$ , e.g., energy production or dissipation, is found in high- $G$  regions, we define a conditional fraction as follows:

$$CF(\mathcal{P}, G) = \frac{\Sigma \mathcal{P}(G \geq g_n)}{\Sigma \mathcal{P}} \quad (3.10)$$

Figure 14?? shows a comparison of the conditional fraction computed for the TKE, energy dissipation, and positive and negative production, computed using equation ???. The positive and negative contribution to the average TKE production were computed separately to distinguish between the direction of energy transfer between the mean flow

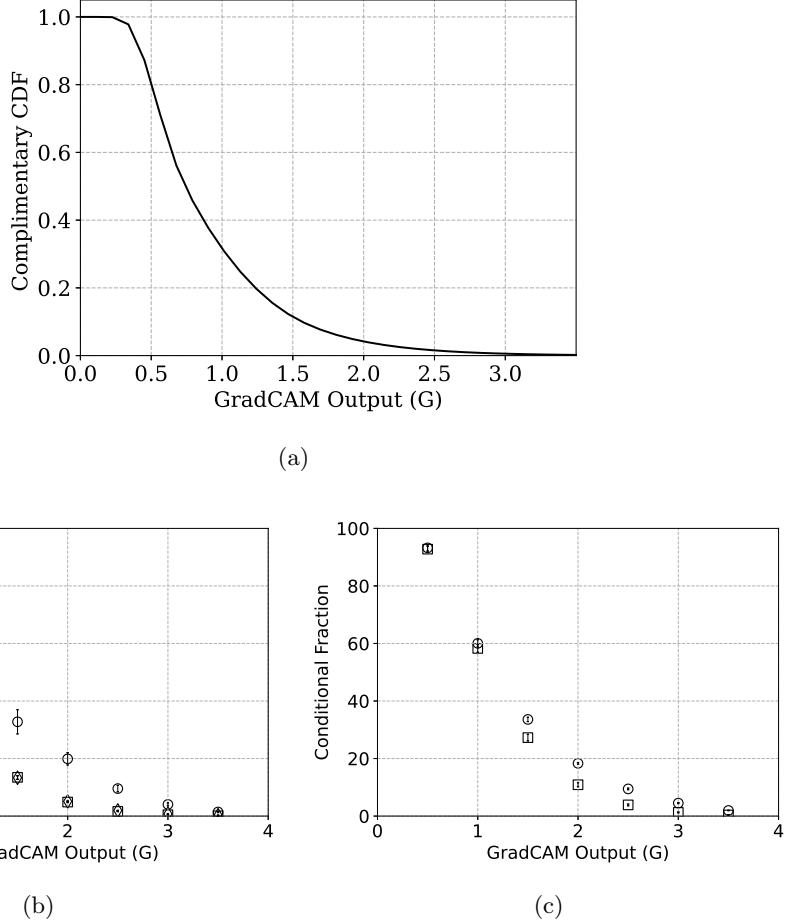


Figure 14: (a) Complimentary Cumulative Distribution Function (CCDF) of GradCAM gradient  $G$ , averaged across multiple time-decorrelated snapshots. ?? and ?? Conditional summation fractions of various parameters of interest. Namely, turbulent kinetic energy ( $\diamond$ ), energy dissipation ( $\square$ ), and energy transfer ( $\circ$ ) in ?? and positive ( $\circ$ ) and negative ( $\square$ ) production in ?? as described in equation ???. The positive and negative contributions to turbulence have been plotted separately to distinguish between the transfer of energy to and from the mean flow.

and fluctuations; positive local production corresponds to the generation of turbulent fluctuations, whereas negative local production acts as a source for the mean flow at the expense of the fluctuating field (B & Sarkar 2011; Cimarelli et al. 2019). Notably, we also observe from figure 14(a) that while regions with  $G \geq 1$  constitute less than 30% of the domain volume, they contain about 60% of the positive production and nearly 60% of the negative production. Similarly,  $G \geq 2$  constitutes less than 5% the domain volume, but these regions account for nearly 20% of the total positive production within a sample and account for about 10% of the negative production. The almost equal contribution from  $P+$  and  $P-$  in regions with low threshold values of  $G$ , i.e., for  $G \leq 1$  indicates that the regions which are the most consequential for the CNN’s inference capabilities, i.e., those containing extreme values of  $G$ , display a higher tendency for generating turbulent

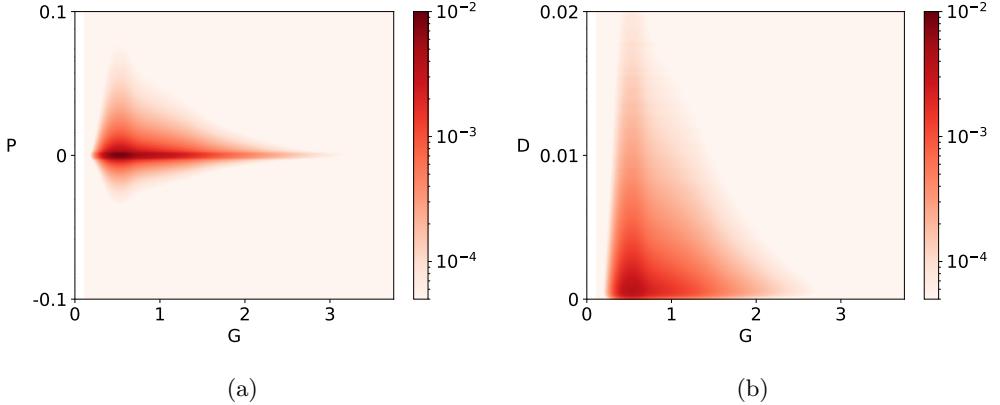


Figure 15: Joint probability density functions (JPDF) of GradCAM gradient with production (a) and dissipation (b). By focusing in on the data with a probability density of at least  $5 \times 10^{-4}$ , we can see the dominant concentrations of both relationships. The majority of GradCAM gradient is around 0.5 as also shown in figure 14(a). In (a) we can observe more positive production (transfer of energy from mean flow to fluctuations) and in (b), the distribution of dissipation is consolidated at lower values.

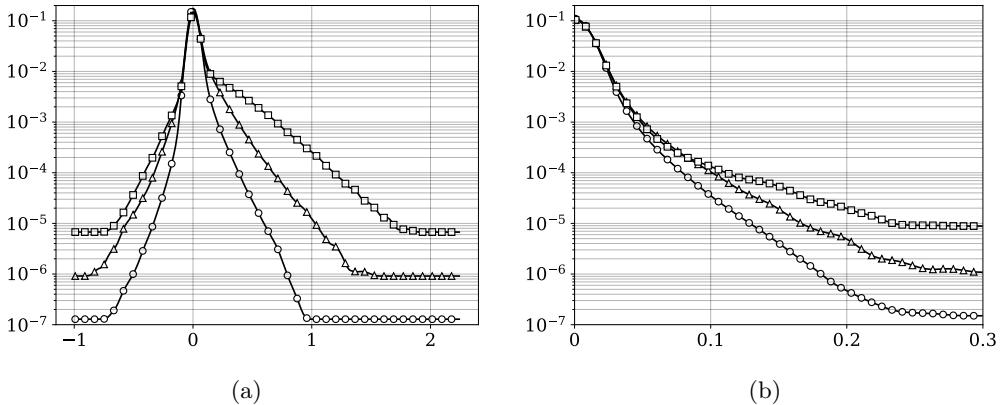


Figure 16: Conditional probability density functions (CPDF) for (a) production and (b) dissipation conditioned by GradCAM gradient, ( $G = 1 \circlearrowleft, 2 \triangle, 3 \square$ ). In both, the y axis is probability density and logarithmically scaled. In (a), the heavy tails of positive production with higher GradCAM gradient values demonstrate the relationship between the two. As well in (b), one can observe the increase likelihood of dissipation where there is higher values of GradCAM gradient.

fluctuations. This is a notable result that implies not only a strong relationship between positive production and the CNN's training metric, i.e., bursting intensity, but also shows a link between ejections and the transfer of energy from the mean flow to the fluctuations.

### 3.9. Joint PDFs and Conditional PDFs

To continue to investigate these relationships statistically, joint (figure 15) and conditional (figure 16) probability density functions (PDF) were calculated comparing the GradCAM gradient to two calculated physical parameters of interest, production and dissipation. In both plots, the focus was kept to data with a probability density over  $5 \times 10^{-4}$  in order to examine the prominent trends in the JPDF. As also shown in figure 14(a), GradCAM gradient is clustered about  $G = 0.5$ . Within figure 15(a), there was a greater concentration in positive production, or the transfer of energy from the mean flow to fluctuations following the energy cascade. Within figure 15(b), the spread of dissipation

This was explored further using conditional PDF in figure 16. In both plots, the y axis is logarithmically scaled to show the full range of probability density and the parameters of interest are conditioned by GradCAM gradients of 1.0, 2.0 and 3.0. 16a is the conditional PDF of production shown at the selected GradCAM gradients. Comparing the PDF at  $G = 1$  to the PDF at  $G = 2\&3$  shows the increasing probability density within the positive production, highlighting the correlation between the two parameters. ?? is the conditional PDF of dissipation at selected GradCAM gradients. Again, comparing the PDF at  $G = 1$  to the PDF at  $G = 2\&3$  shows shows the increasing probability density within higher values of dissipation.

## 4. Conclusion

In this work, a three dimensional Convolutional Neural Network (CNN) has been trained to infer the intensity of intermittent ejection events that occur in the near-wall layer of a turbulent channel flow. The CNN is able to accurately infer ejection intensities in velocity samples which were sufficiently removed from the training dataset so as to be temporally decorrelated. To understand which specific flow characteristics influence the network's ability to make accurate inferences, the CNN's attention on localized spatial regions was examined using the Gradient-weighted Class Activation Mapping (GradCAM) technique. These salient regions correlate well with high intensity ejection parcels, as well as with low-speed streaks undergoing bursting. This indicates that the CNN is able to autonomously identify dynamically important cohesive regions within the turbulent flow field, without a-priori knowledge of the intrinsic dynamics. We also demonstrate that the CNN trained on data at  $Re_\tau = 300$  is able to infer ejection intensities for velocity samples from a different simulation at  $Re_\tau = 670$ . This suggests that the inference ability of the trained CNN is generalizable to some extent, especially with regard to physical processes (i.e., near-wall ejections) that may persist across varying flow conditions. Further analysis indicates that salient regions with the most extreme GradCAM values display a higher tendency for positive energy production, suggesting that the CNN is able to isolate cohesive regions where turbulent fluctuations are generated. **This is further evident from the heavy tailed PDFs....** We note that while the current work presents an alternate means of analyzing nonlinear spatial correlations associated with near-wall bursts, the techniques explored here may be used in other scenarios where the underlying spatial dynamics are not known a-priori. Overall, the results indicate that with the specific modifications presented here, three dimensional Convolutional Neural Networks (CNN), which were originally developed for image recognition and classification, can prove to be immensely useful for uncovering and analyzing nonlinear correlations and salient spatial features present in turbulent flow fields.

## Acknowledgements

This work was supported by the Department of Ocean and Mechanical Engineering at the Florida Atlantic University, and by the National Science Foundation through Grant No. CBET-2103536. The authors thank Prof. Petros Koumoutsakos for helpful discussions, and for providing access to computational resources. Computational resources were provided by the Swiss National Supercomputing Centre (CSCS) under project ID ‘s929’, and by the National Science Foundation under grant CNS-1828181.

## REFERENCES

- ABADI, MARTÍN, AGARWAL, ASHISH, BARHAM, PAUL, BREVDO, EUGENE, CHEN, ZHIFENG, CITRO, CRAIG, CORRADO, GREG S., DAVIS, ANDY, DEAN, JEFFREY, DEVIN, MATTHIEU, GHEMAWAT, SANJAY, GOODFELLOW, IAN, HARP, ANDREW, IRVING, GEOFFREY, ISARD, MICHAEL, JIA, YANGQING, JOZEFOWICZ, RAFAL, KAISER, LUKASZ, KUDLUR, MANJUNATH, LEVENBERG, JOSH, MANÉ, DANDELION, MONGA, RAJAT, MOORE, SHERRY, MURRAY, DEREK, OLAH, CHRIS, SCHUSTER, MIKE, SHLENS, JONATHON, STEINER, BENOIT, SUTSKEVER, ILYA, TALWAR, KUNAL, TUCKER, PAUL, VANHOUCKE, VINCENT, VASUDEVAN, VIJAY, VIÉGAS, FERNANDA, VINYALS, ORIOL, WARDEN, PETE, WATTENBERG, MARTIN, WICKE, MARTIN, YU, YUAN & ZHENG, XIAOQIANG 2015 TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- ADRIAN, R. J. 2007 Hairpin vortex organization in wall turbulence. *Phys. Fluids* **19** (4), 041301.
- B, GAYEN & SARKAR, S. 2011 Negative turbulent production during flow reversal in a stratified oscillating boundary layer on a sloping bottom. *Phys. Fluids* **23** (10), 101703.
- BERKOOZ, G., HOLMES, P & LUMLEY, J L 1993 The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.* **25** (1), 539–575.
- BLACKWELDER, R. F. & KAPLAN, R. E. 1976 On the wall structure of the turbulent boundary layer. *J. Fluid Mech.* **76** (1), 89112.
- BOGARD, D. G. & TIEDERMAN, W. G. 1986 Burst detection with single-point velocity measurements. *J. Fluid Mech.* **162**, 389413.
- CHOLLET, FRANÇOIS & OTHERS 2015 Keras.
- CIMARELLI, A., LEONFORTE, A., ANGELIS, E. DE, CRIVELLINI, A. & ANGELI, D. 2019 On negative turbulence production phenomena in the shear layer of separating and reattaching flows. *Phys. Lett. A* **383** (10), 1019–1026.
- CORINO, E. R. & BRODKEY, R. S. 1969 A visual investigation of the wall region in turbulent flow. *J. Fluid Mech.* **37** (1), 1–30.
- DESJARDINS, O., BLANQUART, G., BALARAC, G. & PITSCHE, H. 2008 High order conservative finite difference scheme for variable density low mach number turbulent flows. *J. Comput. Phys.* **227** (15), 7125–7159.
- DONG, C., LOY, C. C., HE, K. & TANG, X. 2016 Image super-resolution using deep convolutional networks. *IEEE T. Pattern Anal.* **38** (2), 295–307.
- ENCINAR, M. P. & JIMNEZ, J. 2020 Momentum transfer by linearised eddies in turbulent channel flows. *J. Fluid Mech.* **895**, A23.
- ERICHSON, N. B., MATHELIN, L., YAO, Z., BRUNTON, S. L., MAHONEY, M. W. & KUTZ, J. N. 2020 Shallow neural networks for fluid flow reconstruction with limited sensors. *P. R. Soc. A-Math. Phy.* **476** (2238), 20200097.
- FAN, X., HOFMANN, L. & HERBERT, T. 1993 Active flow control with neural networks. In 3rd Shear Flow Conference, pp. 1–10.
- FUKAMI, K., FUKAGATA, K. & TAIRA, K. 2019 Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* **870**, 106–120.
- FUKUSHIMA, K. 1980 Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **36**, 193–202.
- GAMAHARA, M. & HATTORI, Y. 2017 Searching for turbulence models by artificial neural network. *Phys. Rev. Fluids* **2**, 054604.
- GOODFELLOW, IAN, BENGIO, YOSHUA & COURVILLE, AARON 2016 Deep Learning. MIT Press, <http://www.deeplearningbook.org>.

- GUO, X., LI, W. & IORIO, F. 2016 Convolutional neural networks for steady flow approximation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM.
- HACK, M. J. P. & MOIN, P. 2018 Coherent instability in wall-bounded shear. *J. Fluid Mech.* **844**, 917–955.
- HACK, M. J. P. & ZAKI, T. A. 2016 Data-enabled prediction of streak breakdown in pressure-gradient boundary layers. *J. Fluid Mech.* **801**, 43–64.
- HIGHAM, C. F. & HIGHAM, D. J. 2019 Deep learning: An introduction for applied mathematicians. *SIAM Rev.* **61** (3), 860–891.
- JIMÉNEZ, J. 2012 Cascades in wall-bounded turbulence. *Ann. Rev. Fluid Mech.* **44** (1), 27–45.
- JIMÉNEZ, J. 2013 Near-wall turbulence. *Phys. Fluids* **25** (10), 101302.
- JIMÉNEZ, J. 2018 Coherent structures in wall-bounded turbulence. *J. Fluid Mech.* **842**, P1.
- JIMÉNEZ, J. & MOIN, P. 1991 The minimal flow unit in near-wall turbulence. *J. Fluid Mech.* **225**, 213–240.
- JOLLIFFE, I. T. & CADIMA, J. 2016 Principal component analysis: a review and recent developments. *Philos. T. R. Soc. A* **374** (2065), 20150202.
- KIM, H. T., KLINE, S. J. & REYNOLDS, W. C. 1971 The production of turbulence near a smooth wall in a turbulent boundary layer. *J. Fluid Mech.* **50** (1), 133–160.
- KIM, J. 1985 Turbulence structures associated with the bursting event. *Phys. Fluids* **28** (1), 52–58.
- KLINE, S. J., REYNOLDS, W. C., SCHRAUB, F. A. & RUNSTADLER, P. W. 1967 The structure of turbulent boundary layers. *J. Fluid Mech.* **30** (4), 741–773.
- KRAMER, M. A. 1991 Nonlinear principal component analysis using autoassociative neural networks. *AICHE J.* **37** (2), 233–243.
- LECUN, Y., BOTTOU, L., BENGIO, Y. & HAFFNER, P. 1998 Gradient-based learning applied to document recognition. *P. IEEE* **86** (11), 2278–2324.
- LEE, C., KIM, J., BABCOCK, D. & GOODMAN, R. 1997 Application of neural networks to turbulence control for drag reduction. *Phys. Fluids* **9** (6), 1740–1747.
- LEE, S. & YOU, D. 2019 Data-driven prediction of unsteady flow over a circular cylinder using deep learning. *J. Fluid Mech.* **879**, 217–254.
- LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155166.
- LIU, B., TANG, J., HUANG, H. & LU, X.-Y. 2020 Deep learning methods for super-resolution reconstruction of turbulent flows. *Phys. Fluids* **32** (2), 025105.
- LORANG, L. V., PODVIN, B. & LE QUR, P. 2008 Application of compact neural network for drag reduction in a turbulent channel flow at low reynolds numbers. *Phys. Fluids* **20** (4), 045104.
- LOZANO-DURÁN, J. & JIMÉNEZ, J. 2014 Time-resolved evolution of coherent structures in turbulent channels: characterization of eddies and cascades. *J. Fluid Mech.* **759**, 432–471.
- LOZANO-DURN, A., FLORES, O. & JIMNEZ, J. 2012 The three-dimensional structure of momentum transfer in turbulent channels. *J. Fluid Mech.* **694**, 100130.
- LU, S. S. & WILLMARTH, W. W. 1973 Measurements of the structure of the reynolds stress in a turbulent boundary layer. *J. Fluid Mech.* **60** (3), 481511.
- LUMLEY, J. & BLOSSEY, P. 1998 Control of turbulence. *Ann. Rev. Fluid Mech.* **30** (1), 311–327.
- MAULIK, R. & SAN, O. 2017 A neural network approach for the blind deconvolution of turbulent flows. *J. Fluid Mech.* **831**, 151–181.
- MEZI, I. 2013 Analysis of fluid flows via spectral properties of the koopman operator. *Ann. Rev. Fluid Mech.* **45** (1), 357–378.
- MILANO, M. & KOUMOUTSAKOS, P. 2002 Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **182** (1), 1 – 26.
- MIYANAWALA, T. P. & JAUMAN, RAJEEV K. 2018 A novel deep learning method for the predictions of current forces on bluff bodies. In Volume 2: CFD and FSI. American Society of Mechanical Engineers.
- MOIN, P. & MAHESH, K. 1998 Direct numerical simulation: A tool in turbulence research. *Ann. Rev. Fluid Mech.* **30** (1), 539–578.

- MOSER, R. D., KIM, J. & MANSOUR, N. N. 1999 Direct numerical simulation of turbulent channel flow up to  $re_\tau = 590$ . *Phys. Fluids* **11** (4), 943–945.
- MURATA, T., FUKAMI, K. & FUKAGATA, K. 2019 Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J. Fluid Mech.* **882**.
- OFFEN, G. R. & KLINE, S. J. 1975 A proposed model of the bursting process in turbulent boundary layers. *J. Fluid Mech.* **70** (2), 209–228.
- PARK, J. & CHOI, H. 2020 Machine-learning-based feedback control for drag reduction in a turbulent channel flow. *J. Fluid Mech.* **904**, A24.
- POPE, S. B 2000 *Turbulent flows*. Cambridge: Cambridge Univ. Press.
- ROBINSON, S. K. 1991 Coherent motions in the turbulent boundary layer. *Ann. Rev. Fluid Mech.* **23** (1), 601–639.
- ROWLEY, C. W., MEZI, I., BAGHERI, S., SCHLATTER, P. & HENNINSON, D. S. 2009 Spectral analysis of nonlinear flows. *J. Fluid Mech.* **641**, 115127.
- SARGHINI, F., DE FELICE, G. & SANTINI, S. 2003 Neural networks based subgrid scale modeling in large eddy simulations. *Comput. Fluids* **32** (1), 97–108.
- SCHLATTER, P., LI, Q., ÖRLÜ, R., HUSSAIN, F. & HENNINSON, D. S. 2014 On the near-wall vortical structures at moderate reynolds numbers. *Eur. J. Mech. B-Fluid* **48**, 75 – 93.
- SCHMID, P. J. 2010 Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28.
- SCHOPPA, W. & HUSSAIN, F. 2002 Coherent structure generation in near-wall turbulence. *J. Fluid Mech.* **453**, 57–108.
- SEKAR, V., JIANG, Q., SHU, C. & KHOO, B. C. 2019 Fast flow field prediction over airfoils using deep learning approach. *Phys. Fluids* **31** (5), 057103.
- SELVARAJU, R. R., DAS, A., VEDANTAM, R., COGSWELL, M., PARIKH, D. & BATRA, D. 2016 Grad-CAM: Why did you say that? visual explanations from deep networks via gradient-based localization, arXiv: 1610.02391.
- SIMONYAN, K., VEDALDI, A. & ZISSERMAN, A. 2013 Deep inside convolutional networks: Visualising image classification models and saliency maps., arXiv: 1312.6034.
- VIQUERAT, J. & HACHEM, E. 2020 A supervised neural network for drag prediction of arbitrary 2D shapes in laminar flows at low Reynolds number. *Comp. & Fluids* **210**, 104645.
- WALLACE, J. M., ECKELMANN, H. & BRODKEY, R. S. 1972 The wall region in turbulent shear flow. *J. Fluid Mech.* **54** (1), 39–48.
- WILLMARTH, W. W. & LU, S. S. 1972 Structure of the reynolds stress near the wall. *J. Fluid Mech.* **55** (1), 6592.
- YE, S., ZHANG, Z., SONG, X., WANG, Y., CHEN, Y. & HUANG, C. 2020 A flow feature detection method for modeling pressure distribution around a cylinder in non-uniform flows by using a convolutional neural network. *Sci. Rep.-UK* **10** (1), 4459.
- ZHOU, B., KHOSLA, A., LAPEDRIZA, A., OLIVA, A. & TORRALBA, A. 2015 Learning deep features for discriminative localization, arXiv: 1512.04150.