Convolutional autoencoded echo state network for the prediction of extreme events in turbulence

By N. A. K. Doan,† A. Racca‡ AND L. Magri¶‖††

Extreme events are sudden changes of a turbulent state or observable. The time-accurate and statistical prediction of extreme events is difficult to achieve because their dynamics are chaotic, and hence unpredictable. The goal of this paper is to propose a methodology to learn the turbulent dynamics from data for the time-accurate prediction of unseen extreme events. We break down the spatiotemporal turbulent problem into spatial and temporal problems. First, we reduce the high-dimensional turbulent dynamics into a lower-dimensional latent space by using a nonlinear transformation based on convolutional neural networks. Second, we compute the temporal dynamics in the low-dimensional latent space by using a reservoir computer, which is a form of recurrent neural network. We seamlessly assemble the two architectures into the convolutional autoencoded echo state network (CAE-ESN). The test case is a three-dimensional (3D) turbulent flow (the minimal flow unit) in which quasi-relaminarization events with their bursts in kinetic energy and dissipation rate are the extreme events. We show that the CAE-ESN is (i) scalable to a 3D turbulent flow; (ii) efficient, requiring approximately one-tenth as many degrees of freedom as proper orthogonal decomposition for compressing the data; and (iii) accurate in time-accurately forecasting the occurrence of the quasi-relaminarization events and their statistics. The proposed architecture opens opportunities for reduced-order modeling and time-accurate prediction of turbulent flow from data.

1. Introduction

Many fluid dynamics systems exhibit extreme events, which are sudden and violent changes of the flow state. Examples are oceanic rogue waves (Dysthe et al. 2008), extreme weather patters in atmospheric science (Majda 2012), and intermittency in turbulence (Blonigan et al. 2019). The time-accurate prediction of such events remains a major challenge because of the chaotic nature of turbulent flows, in which infinitesimal perturbations in the initial conditions will exponentially amplify.

To tackle the time-accurate prediction of extreme events, recent techniques based on deep learning have been developed. For example, Wan et al. (2018) used an architecture based on long short-term memory (LSTM) networks combined with a reduced-order model based on proper orthogonal decomposition (POD) to predict the occurrence of extreme dissipation events in the two-dimensional Kolmogorov flow. Sapsis (2018) further combined a data-driven reduced-order model with large deviation theory to efficiently characterize heavy-tailed distribution in the Kolmogorov flow. Further development by Doan et al. (2021*b*) and Racca & Magri (2022) focused on echo state networks (ESNs) to

† Faculty of Aerospace Engineering, Delft University of Technology, Netherlands
‡ Department of Engineering, University of Cambridge, United Kingdom
¶ Department of Aeronautics, Imperial College London, United Kingdom
‖ The Alan Turing Institute, United Kingdom
†† Institute of Advanced Study, TU Munich, Germany (visiting)

time-accurately predict extreme events in a nine-dimensional model of turbulence. One of the main advantages of ESNs is that it is straightforward to train with just a matrix inversion, in comparison to LSTM, which requires a gradient-based optimization process.

The works discussed above highlight the potential of deep learning for the time-accurate prediction of extreme events in turbulent flows, but these studies were restricted to small models of turbulence. Therefore, the applicability of these deep-learning-based techniques, and specifically ESNs, to time-accurately predict extreme events in 3D flows remains unknown. In this paper, we propose the 3D convolutional autoencoder echo state network (CAE-ESN) to learn the dynamics of the 3D minimal flow unit (MFU) and time-accurately predict the occurrence of extreme events in the flow. We will explore whether the CAE-ESN can be scaled to a 3D flow, how it can represent the flow dynamics in a latent space with a reduced number of degrees of freedom, and whether it can time-accurately predict extreme events. Section 2 describes the MFU and its extreme events as well as the numerical setup used to simulate it. Section 3 presents in detail the CAE-ESN framework used to learn the dynamics of the MFU, and the accuracy of the CAE-ESN in predicting the statistics and the time-accurate prediction of extreme events of the MFU are discussed in Section 4. A summary of the main results and directions for future work are provided in Section 5.

2. Minimal flow unit

The flow under consideration is the MFU (Jiménez & Moin 1991). This is a prototypical flow of near-wall turbulence in channel flows, which consists of a turbulent channel flow whose dimensions are smaller than conventional channel flow simulations. In this small domain, the flow accurately reproduces the near-wall turbulent statistics of turbulent channel flows. The MFU is governed by the incompressible Navier-Stokes equations, which are equipped with initial conditions

$$\begin{aligned} \nabla \cdot \boldsymbol{u} &= 0, \\ \partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} &= \frac{1}{\rho} \boldsymbol{f_0} - \frac{1}{\rho} \nabla p + \nu \Delta \boldsymbol{u}, \end{aligned} \tag{2.1}$$

where $\boldsymbol{u} = (u, v, w)$ is the 3D velocity field and $\boldsymbol{f}_0 = (f_0, 0, 0)$ is the constant forcing in the streamwise, x, direction. ρ, p, and ν are the density, pressure, and kinematic viscosity, respectively. The geometrical domain of the MFU is composed of two walls, in the wall-normal, y, direction with a no-slip boundary condition such that $\boldsymbol{u}(x, \pm\delta, z, t) = 0$, where δ is half the channel width. Periodic boundary conditions are considered for the streamwise, x, and spanwise, z, directions. For this study, a channel with dimension $\pi\delta \times 2\delta \times 0.34\pi\delta$ is considered, as done by Blonigan et al. (2019) with $\delta = 1.0$. The Reynolds number of the flow, based on the bulk velocity and the half-channel width, is set to $Re = 3000$, which corresponds to a friction Reynolds number $Re_\tau \approx 140$. We employ an in-house code based on work by Bernardini et al. (2014) to simulate the MFU. The flow is discretized on a Cartesian grid with staggered central second-order finite-difference approximations. Time marching is performed with a third-order low-storage Runge-Kutta algorithm coupled with a second-order Crank-Nicolson scheme, which are combined in a fractional-step procedure in which the convective and diffusive terms are treated explicitly and implicitly, respectively (Bernardini et al. 2014). The solution of the Poisson equation for the pressure is obtained with a direct solver based on the Fourier transform (Kim & Moin 1985).

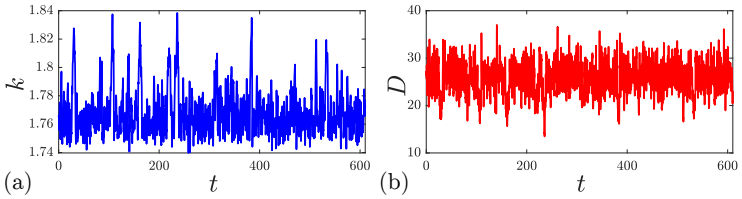

FIGURE 1. Time evolution of the kinetic energy (a) and dissipation rate (b) in the minimal flow unit. Extreme events correspond to the peaks in k and D.

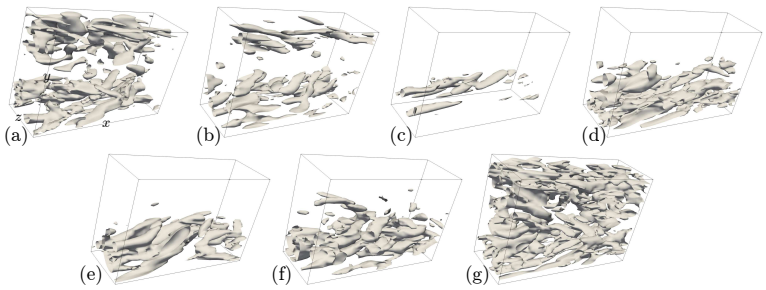

FIGURE 2. Snapshots of the Q-criterion isosurface (with value $Q = 0.1$) during an extreme event, where $Q = 0.5(||\boldsymbol{\omega}||^2 - ||\boldsymbol{S}||^2)$, $\boldsymbol{\omega}$ is the vorticity vector, and $\boldsymbol{S}$ is the strain-rate tensor.

2.1. *Extreme events*

The extreme events in the MFU are quasi-relaminarization events, which take place close to either wall, and are accompanied by bursts in the total kinetic energy, $k(t)$, and its dissipation rate, $D(t)$. These are computed as

$$k(t) = \int\int\int_{\Omega} \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}dxdydz, \quad D(t) = \int\int\int_{\Omega} \mathrm{tr}\left(\boldsymbol{\tau}\nabla\boldsymbol{u}\right)dxdydz, \tag{2.2}$$

where Ω is the computational domain and $\boldsymbol{\tau} = \mu\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\right)$ is the stress tensor, with μ being the dynamic viscosity.

Figure 1 shows the time evolution of k and D, in which the extreme events are related to the peaks in their time series which correspond to quasi-relaminarization events close to either or both walls, as shown in Figure 2. Time is normalized by the eddy turnover time. During an extreme event, (i) the flow at either wall (the upper wall in Figure 2) becomes laminar (Figure 2(a-c)); (ii) the flow remains laminar for some time (Figure 2(c-f)), which results in a larger axial velocity close to the centerline (and therefore an increase in $k(t)$); (iii) the higher velocity close to the centerline makes the effective Reynolds number of the flow larger, which in turn makes the flow prone to a turbulence burst on the quasi-laminar wall; (iv) the turbulence burst occurs on the quasi-laminar wall, which results in a large increase in the dissipation rate, $D(t)$; and (v) the flow close to that quasi-laminar wall becomes turbulent again, which leads to a decrease in the kinetic energy, $k(t)$ (Figure 2(g)).

Because an extreme event is localized at either wall, the total kinetic energy computed over the whole domain makes the identification of specific extreme events difficult, and only very strong extreme events where the flow near both walls simultaneously becomes quasi-laminar are apparent in the time series of total kinetic energy. Those are the peaks observed in Figure 1. Therefore, we define a near-wall kinetic energy deficit to identify extreme events. This is defined as the deficit of kinetic energy in a layer at either wall,

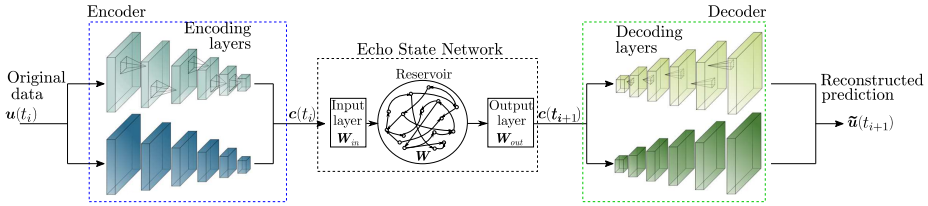

FIGURE 3. Schematic of the 3D convolutional autoencoded echo state network.

which occurs when it becomes quasi-laminar. Hence, we first define

$$k^u = \int_{0.75\delta}^{\delta} \int\int \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}dxdzdy, \quad k^l = \int_{-\delta}^{-0.75\delta} \int\int \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}dxdzdy \tag{2.3}$$

where k^u and k^l are the near-wall kinetic energy of the flow at the upper and lower wall, respectively, and compute the normalized kinetic energy deficit as $\widetilde{k}_w^{\cdot} = 1 - k_w^{\cdot}/\max(k_w^{\cdot})$ where $\cdot$ is l (u) for the lower (upper) wall. The symbol $\tilde{\cdot}$ indicates the normalization. We define an extreme event as the event peak that occurs when either k_w^u or k_w^l are greater than 0.55. This threshold was chosen to ensure that only clear peaks of kinetic energy are considered as extreme events. Such a value indeed indicates a large deficit in kinetic energy near that wall, and thus a quasi-laminar flow state. The dataset of the MFU contains 2000 eddy turnover time (i.e., 20000 snapshots) on a grid of $32 \times 256 \times 16$, which contains 50 extreme events. The first 200 eddy turnover times of the dataset (2000 snapshots) are employed for the training, which contains only 4 extreme events.

3. Methodology

We develop a hybrid architecture: the 3D convolutional autoencoded echo state network (CAE-ESN). This is composed of (i) a 3D convolutional autoencoder, which learns an appropriate reduced latent representation of the flow state, and (ii) an echo state network, which learns the flow dynamics in the latent space. A schematic of the proposed architecture is shown in Figure 3.

3.1. *Convolutional autoencoder*

The three dimensional convolution autoencoder (CAE) (boxes in blue and green in Figure 3) learns an efficient reduced-order representation of the original data, which represents the flow state $\boldsymbol{u} \in \mathbb{R}^{N_x \times N_y \times N_z \times N_u}$, where N_x, N_y and N_z are the number of gridpoints in the streamwise, wall-normal and spanwise directions, respectively. The number of velocity components is $N_u = 3$. On one hand, the encoder (blue box in Figure 3) reduces the dimension of the data down to a latent state, $\boldsymbol{c}$ of a smaller dimension N_c. This operation can be symbolically expressed as $\boldsymbol{c} = \mathcal{E}(\boldsymbol{u}; \boldsymbol{\phi}_E)$ where $\boldsymbol{\phi}_E$ represents the weights of the encoder. On the decoder side (green box in Figure 3), the CAE then reconstructs the data from the latent state back to the original full flow state. This operation is expressed as $\widetilde{\boldsymbol{u}} = \mathcal{D}(\boldsymbol{c}; \boldsymbol{\phi}_D)$ where $\boldsymbol{\phi}_D$ are the trainable weights of the decoder. We employ a multiscale autoencoder, which was originally developed for image-based super-resolution analysis (Hasegawa et al. 2020). It relies on the use of various scales of convolutional filters to analyse the input information and allowing the reconstruction (Du et al. 2018). In this work, two scales of filters ($3\times5\times3$) and ($5\times7\times5$) are employed (represented schematically by the two parallel streams of encoder/decoder in the blue and green boxes in Figure 3). This choice ensures a trade-off between the size of the 3D multiscale autoencoder

and the reconstruction accuracy (Section 4). To reduce the dimension of the input, the convolution operation in each branch of the encoder is applied in a strided manner, which means that the convolutional neural network (CNN) kernel is progressively applied to the entire input by moving the CNN kernel by $(s_x, s_y, s_z) = (2, 4, 2)$ grid points. This results in an output of a smaller dimension than the input. After each convolution layer, to fulfill the boundary conditions of the MFU, periodic padding is applied in the x and z directions, while zero padding is applied in the y direction. Three successive layers of CNN/padding operations are applied to decrease the dimension of the original field from $(32, 256, 16, 3)$ to $(2, 4, 2, N_f)$, where N_f is the specified number of filters in the last encoding layer. As a result, the dimension of the latent space is $N_c = 16 \times N_f$. The decoder mirrors the architecture of the encoder, where transpose CNN layers (Zeiler et al. 2010) are used, which allows for an increase in the dimension of the information from the latent dimension up to the original flow dimension. The end-to-end autoencoder is trained by minimizing the mean squared error (MSE) between the reconstructed velocity field, $\widetilde{\boldsymbol{u}}$, and the original field, $\boldsymbol{u}$.

3.2. *Echo state network*

An ESN is composed of (i) an input matrix, $\boldsymbol{W}_{in}$, (ii) a reservoir characterized by a state matrix, $\boldsymbol{W}$, and (iii) an output matrix, $\boldsymbol{W}_{out}$ (Figure 3). At a time t_i, the input vector, $\boldsymbol{c}(t_i) \in \mathbb{R}^{N_c}$is mapped into the reservoir state by the input matrix. The reservoir state, $\boldsymbol{r}(t_i) \in \mathbb{R}^{N_r}$ with $N_r \gg N_c$, is updated at each time iteration as a function of the current input and its previous value as

$$\boldsymbol{r}(t_{i+1}) = \tanh\left(\boldsymbol{W}_{in}[\boldsymbol{c}(t_i); 0.1] + \boldsymbol{W}\boldsymbol{r}(t_i)\right), \tag{3.1}$$

where $[\cdot;\cdot]$ indicates a vertical concatenation to introduce a bias at the input. The predicted output, $\widetilde{\boldsymbol{c}}(t_{i+1}) \in \mathbb{R}^{N_c}$, is obtained from

$$\tilde{\boldsymbol{c}}(t_{i+1}) = \widehat{\boldsymbol{r}}(t_{i+1})^T \boldsymbol{W}_{out}, \tag{3.2}$$

where $\widehat{\boldsymbol{r}}(t_{i+1}) = g(\boldsymbol{r}(t_{i+1}))$ is the result of a nonlinear transformation of $\boldsymbol{r}(t_{i+1})$ through the nonlinear operator $g(\cdot)$. Here, g is the vertical concatenation of $\boldsymbol{r}(t_i)$ with the unit scalar, i.e., $\widehat{\boldsymbol{r}}(t_i) = [\boldsymbol{r}(t_i); 1]$, which is introduced following Racca & Magri (2021) to break possible unphysical symmetric solution from the ESN. The input matrix, $\boldsymbol{W}_{in}$, and state matrix, $\boldsymbol{W}$, are (pseudo)randomly generated and fixed. Only the weights of the output matrix, $\boldsymbol{W}_{out}$, are computed by training the network. The input matrix and state matrix are constructed as described by Racca & Magri (2021), which results in a tunable input scaling σ_{in}, which describes the range of the distribution from which the elements of $\boldsymbol{W}_{in}$ are sampled, and a tunable spectral radius ρ of $\boldsymbol{W}$. To train the output matrix, $\boldsymbol{W}_{out}$, first the ESN is run in an open-loop configuration, where the data is fed to the ESN as input at each time step and the reservoir state, $\boldsymbol{r}(t_i)$, is stored (except the initial transient part, which is called the washout part). Subsequently, we minimize the MSE between the output of the ESN, $\widetilde{\boldsymbol{c}}(t_i)$, and the data, $\boldsymbol{c}(t_i)$, over a training set of N_{tr} points. This can be done efficiently with ridge regression

$$\left(\boldsymbol{R}\boldsymbol{R}^T + \beta\boldsymbol{I}\right)\boldsymbol{W}_{out} = \boldsymbol{R}\boldsymbol{C}^T, \tag{3.3}$$

where $\boldsymbol{R} \in \mathbb{R}^{N_r \times N_{tr}}$ and $\boldsymbol{C} \in \mathbb{R}^{N_c \times N_{tr}}$ are the horizontal concatenation of the updated reservoir states, $\boldsymbol{r}(t_i)$, and the data, $\boldsymbol{c}(t_i)$, respectively; $\boldsymbol{I}$ is the identity matrix; and β is the user-defined Tikhonov regularization parameter. Because the cost functional is quadratic, the training finds a global minimum with a solution of a linear system.

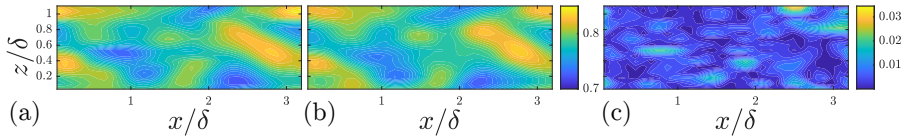

FIGURE 4. Comparison of (a) the actual velocity magnitude (ground truth), (b) the CAE-reconstructed velocity magnitude, (c) the root-squared difference between (a) and (b) in the mid-y plane for a typical snapshot in the testing dataset.

For a given reservoir size, N_r, to identify appropriate hyperparameters of the ESN, i.e., the input scaling, σ_{in}, Tikhonov regularization parameter, β, and spectral radius ρ, a Bayesian optimization approach based on the recycle validation in employed (Racca & Magri 2022). The role of the ESN in the architecture we propose is to learn the dynamics of the MFU in the latent space discovered by the autoencoder, thereby enabling the forecast of the MFU in this latent space. Given a state of the MFU defined in the latent space at a time t_i, $\boldsymbol{c}(t_i)$, the ESN time advances it to obtain $\boldsymbol{c}(t_{i+1}) = \mathcal{F}(\boldsymbol{c}(t_i); \boldsymbol{\phi}_F)$, where $\mathcal{F}$ represents the operation performed by the ESN and $\boldsymbol{\phi}_F$ its tunable (hyper)parameters.

3.3. *Convolutional autoencoded echo state network*

We propose the CAE-ESN to time-accurately predict turbulent dynamics and extreme events from data. First, the CAE is trained separately on the dataset of the MFU, using a 60/20/20% split for training/validation/testing, selected randomly from the dataset described in Section 2, $\{\boldsymbol{u}(t_i)\}_{i=1,\dots,N_t}$, where N_t is the number of snapshots used (here 2000). The ADAM optimizer is used to train the CAE (Kingma & Ba 2015). Second, once the CAE has been trained, the encoder is employed to convert the original dataset from the full-flow state to the latent space. Therefore, the dataset $\{\boldsymbol{c}(t_i)\}_{i=1,\dots,N_t}$ is obtained as $\{\boldsymbol{c}(t_i)\}_{i=1,\dots,N_t} = \{\mathcal{E}(\boldsymbol{u}(t_i))\}_{i=1,\dots,N_t}$. The ESN is then trained separately using the dataset $\{\boldsymbol{c}(t_i)\}_{i=1,\dots,N_t}$ with a Bayesian optimization approach, as described in Section 3.2. Once this is done, the entire CAE-ESN is trained. Finally, to forecast the MFU dynamics, all the parts of the CAE-ESN are seamlessly assembled as follows: (i) A given input flow state $\boldsymbol{u}(t_i)$ is fed to the encoder part of the autoencoder, (ii) the encoder transforms the input, $\boldsymbol{u}(t_i)$, in the latent space, $\boldsymbol{c}(t_i) = \mathcal{E}(\boldsymbol{u}(t_i); \boldsymbol{\phi}_E)$, (iii) the ESN time advances the flow state in the latent space, $\boldsymbol{c}(t_{i+1}) = \mathcal{F}(\boldsymbol{c}(t_i); \boldsymbol{\phi}_F)$, and (iv) the decoder reconstructs the predicted time-advanced flow state as $\widetilde{\boldsymbol{u}}(t_{i+1}) = \mathcal{D}(\boldsymbol{c}(t_{i+1}); \boldsymbol{\phi}_D)$. If the CAE-ESN is used to perform autonomous prediction (and therefore is in a "close-loop" format), this last prediction can then be fed back as the input of the encoder to obtain multistep prediction of the CAE-ESN.

4. Results

We assess the proposed CAE-ESN with three tests.

4.1. *Reconstruction error*

We analyze the ability of the CAE to learn a latent space in which to encode the flow state accurately. We train three CAEs with dimensions of latent space $N_c = 384$, 768, and 1536. We compute their reconstruction errors based on the MSE. A typical comparison between a reconstructed velocity field obtained from the CAE with $N_c = 1536$ is shown in Figure 4. The CAE is able to reconstruct accurately the features of the velocity field.

To provide a comparison with the CAE, we also compute the reconstruction error

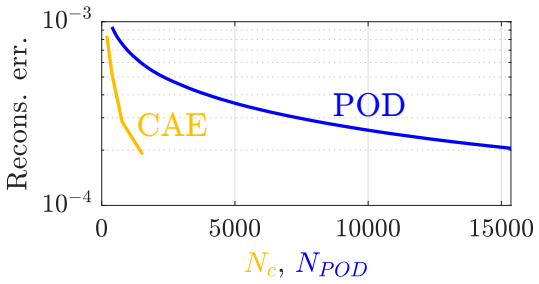

FIGURE 5. Reconstruction error with POD-based method (blue) and autoencoder (yellow) for different dimensions of the latent space, N_c, or number of retained POD modes, N_{POD}. The reconstruction error is computed as the mean squared error between the reconstructed velocity field and the exact one, averaged over the testing dataset.

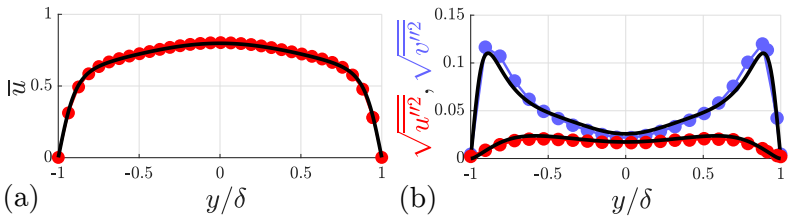

FIGURE 6. Profile of (a) mean streamwise and (b) fluctuating streamwise and spanwise velocity profiles. Black line: true statistics. Red/blue dots: statistics from CAE-ESN.

obtained when using modes from POD with a given number of retained POD modes. The POD modes are obtained with the method of snapshots on the same dataset. The reconstruction error is calculated for a varying number of POD modes. Figure 5 shows this comparison. POD decomposition requires more than 15000 POD modes to reach the same level of accuracy as the CAE with a latent space of dimension 1536. This highlights the advantage of learning a latent representation with nonlinear operations, as in the autoencoder, compared with relying on a linear combination of modes, as in POD.

4.2. *Turbulent statistics*

We assess the ability of the CAE-ESN to accurately reproduce the dynamics of the MFU in a statistical sense. In this section and the next, a CAE-ESN with $N_c = 1536$ and $N_r = 200000$ is used. First, the statistics of the velocity are analyzed here. To do so, the CAE-ESN is allowed to evolve autonomously; i.e., after a short washout time, the prediction from the CAE-ESN is fed back as its input so that the CAE-ESN makes a long-term prediction of the flow evolution. This has been performed for a duration of 500 eddy turnover times, over which the velocity statistics are collected. Figure 6 shows the velocity statistics averaged in time and over the periodic directions (x and z directions), which results in profiles along the wall-normal direction. The CAE-ESN reproduces accurately the statistics of velocity, which means that the CAE-ESN is able to make long-term predictions of the MFU while exhibiting the correct dynamics.

4.3. *Extreme events forecasting*

The ability of the CAE-ESN to forecast extreme events of the MFU is assessed. To do so, from the full MFU dataset of 2000 eddy turnover times, all the instants at which an extreme event occurs are extracted by identifying the time instants when the near-wall

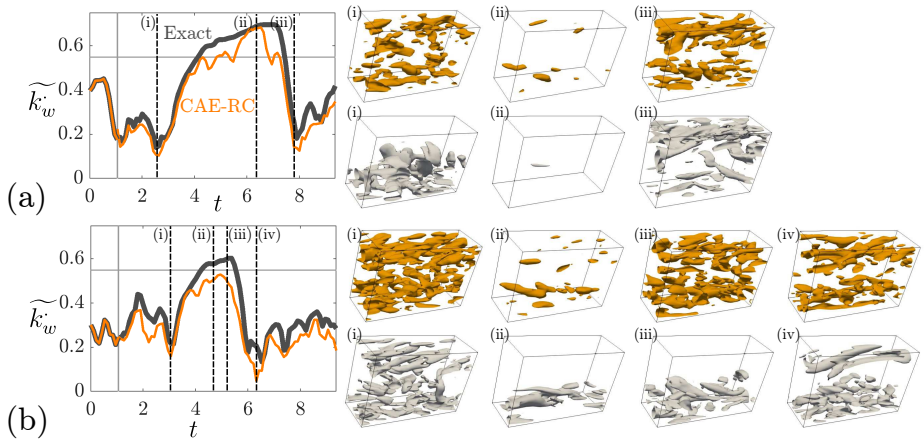

FIGURE 7. Typical evolution during an extreme event of the near-wall kinetic energy deficit during (a) a true positive and (b) a false negative predicted by the CAE-ESN (orange) and the true evolution (dark gray). The insets show the Q-criterion for the time instants indicated by the dashed vertical lines. The vertical full line indicates the instant when the CAE-ESN starts its autonomous prediction and the horizontal full line indicates the threshold for extreme events.

kinetic energy deficit crosses the specified threshold. From these data, 50 extreme events are identified and the associated time instants are denoted $t_{ex,n}$, where n is the index of the extreme event. For each extreme event, the CAE-ESN is initialized with the flow state at time $t = t_{ex,n} - t_{pred}$, where t_{pred} is the prediction time. Subsequently, the CAE-ESN is employed to perform a flow prediction. We determine whether CAE-ESN has predicted the event by assessing whether the predicted normalized near-wall kinetic energy deficit, $\widetilde{k_w^{\cdot}}$, reaches the threshold value defined in Section 2 in the time window $[t_{ex,n} - 4\tau_e, t_{ex,n} + 4\tau_e]$, where τ_e is the eddy turnover time. The prediction is considered a true positive when such a prediction is made correctly. If the CAE-ESN has not predicted the extreme event, i.e., the predicted value of the normalized near-wall kinetic energy deficit does not cross the defined threshold, then the prediction is considered a false negative.

Typical evolutions of the flow field and near-wall kinetic energy deficit during a true positive and a false negative are shown in Figure 7, in which the time is normalized by the eddy turnover time. For the true positive case, the CAE-ESN is able to predict the occurrence of the relaminarization event that occurs on both walls. This is further evidenced by the near-wall kinetic energy, which evolves as the true flow. For the false negative, the CAE-ESN partially predicts a quasi-relaminarization event. However, the duration during which the flow close to the upper wall exhibits a quasi-laminar state is underestimated, and the flow returns to a fully turbulent state faster than the true flow state. This suggests that the ESN may not have accurately predicted the flow in the latent space. This could potentially be improved by combining the training of the CAE with the ESN (Doan et al. 2021*a*) or by enforcing physical constraints within the latent space (Doan et al. 2021*b*). Such explorations are left for future work.

The performance of the CAE-ESN in predicting the extreme events is shown in Figure 8. Figure 8(a) depicts the recall value, defined as $TP/(TP + FN)$, where TP and FN are the number of true positives and false negatives, respectively, for different values of the threshold used to define an extreme event and different prediction times. For a prediction time of one eddy turnover time, a recall value of 85% is found, which shows

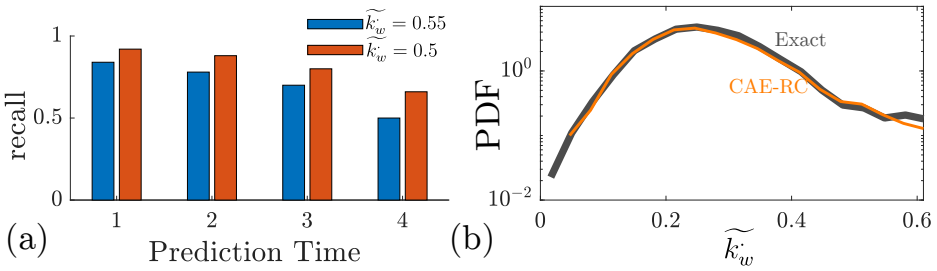

FIGURE 8. (a) Recall of the CAE-ESN for different thresholds for the definition of an extreme event and different prediction times (expressed in eddy turnover time). (b) PDF of the near-wall kinetic energy deficit.

that the CAE-ESN is able to forecast the short-term occurrence of extreme events. As expected, the performance decreases when the prediction time increases. A decrease in the threshold used to define an extreme event leads to an increase in the recall. This is in agreement with the observations made in Figure 7, in which the CAE-ESN is able to predict the main features of the MFU during an extreme event but where a lower peak value of $\widetilde{k_w}$ was observed for the CAE-ESN. The latter was due to some inaccuracies in how the CAE-ESN predicts the flow features and its dynamics. Finally, Figure 8(b) shows the probability density function (PDF) of the near-wall kinetic energy deficit obtained from performing a long-term prediction of 1000 eddy turnover times. The CAE-ESN reproduces accurately the long-tailed PDF of the near-wall kinetic energy deficit up to values larger than the extreme events threshold, indicating that the CAE-ESN can reproduce the occurrence of extreme events in a statistical sense.

5. Conclusions and outlook

In this work, we develop a machine-learning method to time-accurately predict the occurrence of extreme events in a turbulent flow. We propose the 3D CAE-ESN framework to learn the dynamics of the MFU, which exhibits extreme events in the form of near-wall quasi-relaminarization events. This framework is composed of a CAE, which learns an efficient reduced latent representation of the flow state, and of an ESN, which is able to learn the dynamics of the MFU in this latent space. We apply this framework to the 3D MFU. First, we show that the CAE-ESN is able to compress the flow state by three orders of magnitude to a lower-dimensional latent space of a few thousand variables to accurately reconstruct the flow state from this latent space. This constitutes a significant improvement over a POD-based approach, which requires at least one order of magnitude more POD modes to achieve a similar accuracy as the CAE. This is a particularly important advantage when attempting to identify the dynamics in this latent space, given that this latent space with CAE has a reduced number of degrees of freedom to consider. Second, the combined CAE-ESN is used to learn and predict the dynamics of the MFU. The CAE-ESN can reproduce autonomously the statistics of the velocity field of the MFU, which demonstrates the ability of the proposed framework to learn and make long-term predictions of the dynamics of the MFU. Finally, the ability of the CAE-ESN to time-accurately forecast the occurrence of extreme events is analyzed. The CAE-ESN has a high accuracy in predicting extreme events and in reproducing the tail-end statistics of the wall kinetic energy deficit magnitude, which is associated with the extreme events.

The proposed method and results open up possibilities for using deep learning to learn the dynamics of turbulent flows and predict extreme events. Future work will be devoted to physically interpreting the latent space discovered by the CAE, and to exploiting the latent representation to impose physical constraints such as the Navier-Stokes equations.

Acknowledgments

The authors thank Dr. Modesti for providing the flow solver, and Dr. Maeda, Dr. Mirjalili, and Dr. C. Laurent for acting as CTRSP hosts.

REFERENCES

Bernardini, M., Pirozzoli, S. & Orlandi, P. 2014 Velocity statistics in turbulent channel flow up to $Re_\tau = 4000$. *J. Fluid Mech.* **742**, 171–191.

Blonigan, P. J., Farazmand, M. & Sapsis, T. P. 2019 Are extreme dissipation events predictable in turbulent fluid flows? *Phys. Rev. Fluids* **4**, 044606.

Doan, N. A. K., Polifke, W. & Magri, L. 2021*a* Auto-encoded reservoir computing for turbulence learning. *Lect. Notes Comput. Sc.* **12746**, 344–351.

Doan, N. A. K., Polifke, W. & Magri, L. 2021*b* Short- and long-term prediction of a chaotic flow: a physics-constrained reservoir computing approach. *P. R. Soc. Lond. A* **477**, 20210135.

Du, X., Qu, X., He, Y. & Guo, D. 2018 Single image super-resolution based on multi-scale competitive convolutional neural network. *Sensors* **18**, 789.

Dysthe, K., Krogstad, H. E. & Müller, P. 2008 Oceanic rogue waves. *Annu. Rev. Fluid Mech.* **40**, 287–310.

Hasegawa, K., Fukami, K., Murata, T. & Fukagata, K. 2020 Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. *Theor. Comput. Fluid Dyn.* **34**, 367–383.

Jiménez, J. & Moin, P. 1991 The minimal flow unit in near-wall turbulence. *J. Fluid Mech.* **225**, 213–240.

Kim, J. & Moin, P. 1985 Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comput. Phys.* **59**, 308–323.

Kingma, D. P. & Ba, J. L. 2015 Adam: a method for stochastic optimization. *3rd Int. Conf. Learn. Represent. (ICLR 2015) Conf. Track Proc.* pp. 1–15.

Majda, A. J. 2012 Challenges in climate science and contemporary applied mathematics. *Commun. Pure Appl. Math.* **65**, 920–948.

Racca, A. & Magri, L. 2021 Robust optimization and validation of echo state networks for learning chaotic dynamics. *Neural Networks* **142**, 252–268.

Racca, A. & Magri, L. 2022 Data-driven prediction and control of extreme events in a chaotic flow. *Phys. Rev. Fluids* **7**, 104402.

Sapsis, T. P. 2018 New perspectives for the prediction and statistical quantification of extreme events in high-dimensional dynamical systems. *Philos. T. R. Soc. A* **376**, 21270133.

Wan, Z. Y., Vlachas, P., Koumoutsakos, P. & Sapsis, T. P. 2018 Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PLoS One* **13**, e0197704.

Zeiler, M. D., Krishnan, D., Taylor, G. W. & Fergus, R. 2010 Deconvolutional networks. *Proc. CVPR IEEE*, pp. 2528–2535.