

Data-driven control of extreme events in turbulent flows through latent space clustering

Youssef Shehata ^{1,2,*}, Kevin Schuurman ², Pablo Domínguez Estévez ³,
and Nguyen Anh Khoa Doan ^{1,2}

¹*Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands*

²*Department of Aeronautics, Imperial College London, London SW7 2AZ, United Kingdom*

³*School of Aeronautics and Space, Institute of Physics and Aerospace Science (IFCAE), University of Vigo, 32004 Ourense, Spain*



(Received 25 November 2025; accepted 4 August 2026; published 10 September 2026)

Turbulent flows can exhibit extreme events, which are characterized by sudden bursts in the system observables. These events pose significant challenges for prediction and control owing to their intermittent, high-dimensional, and strongly nonlinear nature. In this work we present a predominantly data-driven control framework for the suppression of extreme events in turbulent flows, leveraging reduced-order modeling for nonlinear compression of high-dimensional flow fields and a data-driven clustering algorithm for the identification of precursors to extreme events. A control law is defined in the low-dimensional latent space and is optimized to efficiently mitigate extreme events through actuation values associated with the preidentified clusters. To address issues in the latent space structure that arise from invariance transformations present in many nonlinear systems, leading to an inflated latent space, we employ symmetry-aware autoencoders to establish a more structured and compact latent space. By analyzing high-fidelity simulations of a canonical chaotic flow (Kolmogorov flow), we demonstrate how the resulting closed-loop dynamics exhibit a substantial reduction (up to ca. 99.4%) in the frequency and intensity of extreme events. The framework exhibits scalability to a more chaotic flow regime characterized by higher Reynolds number without encountering numerical instabilities, achieving a 96.9% reduction in the occurrence of extreme events. The ability to incorporate controller limitations, including actuator latency relevant to practical applications of the framework, has also been demonstrated. This study highlights the efficacy of data-driven methods that require little to no prior knowledge of the underlying system dynamics to achieve effective flow control, thereby providing a pathway for general real-time suppression of extreme events in turbulent flows.

DOI: [10.1103/jbgr-wsnn](https://doi.org/10.1103/jbgr-wsnn)

I. INTRODUCTION

Turbulent flows, fundamental to understanding atmospheric dynamics, ocean circulation, and engineering applications, exhibit complex dynamics where small perturbations result in significant deviations over time [1,2]. Due to their sensitivity to initial conditions and inherently nonlinear

*Contact author: y.shehata26@imperial.ac.uk

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

dynamics, long-term prediction of fluid systems is fundamentally intractable [3]. These flows typically evolve on background attractors but can occasionally visit unstable regions of the phase space. Such excursions may trigger extreme events, manifesting as rare, high-amplitude deviations with potentially catastrophic consequences, such as rogue waves [4,5], sudden climate shifts [6–8], dissipation events in turbulent flows [9,10], and flashback in turbulent reacting flows [11–13].

The identification of precursors, which indicate the onset of extreme events, is key to facilitate timely interventions. Therefore, recent research has increasingly focused on identifying precursors. In fluid dynamics, various approaches have been devised to tackle this topic. Autoencoder-based approaches provide an effective framework for extracting dynamically relevant structures from high-dimensional turbulent flows by mapping the data into a low-dimensional space [14–21]. Within this low-dimensional space, temporal evolution can be modeled efficiently using either learned discrete- or continuous-time operators, after which the states are mapped back to the full physical space. This formulation is particularly appealing due to its efficient nonlinear dimensionality reduction and unsupervised feature extraction, which enable the discovery of precursors in complex, multiscale dynamics. Moreover, extreme events in turbulence have been reported to be preceded by characteristic spatiotemporal patterns that may serve as predictive signatures, identified by solving a PDE-constrained optimization problem [9,10]. However, the reliability of such precursors remains strongly system-dependent and thus, might not generalize to other systems and domains. Furthermore, this approach may not scale well with Reynolds number due to numerical instability issues as the adjoint grows exponentially in time at a rate close to the largest Lyapunov exponent of the flow [10]. Other approaches have leveraged data-driven techniques, including surrogate modeling through deep learning models [22,23] and clustering algorithms [24], to identify precursor states leading to extreme events, generalizing to wide flow regimes and with potential for application to other chaotic systems.

Beyond precursor identification, the control of extreme events in turbulence and, generally, chaotic systems represents a significant challenge due to the inherent sensitivity of these systems to tiny perturbations. Few control strategies for mitigating extreme events have been explored. One approach, primarily introduced for systems outside the domain of turbulent flows, involves using reservoir computing to predict extreme events and apply minimal perturbations for their suppression [25]. This method has been demonstrated in systems like coupled FitzHugh-Nagumo neurons [26], where early detection allows for timely feedback control. For turbulent flows, Racca and Magri [23,27] integrate an echo state network, a type of reservoir computing, to suppress extreme events in a model of chaotic shear flows, leveraging data-driven methods that does not rely on the knowledge of the governing equations. Farazmand and Sapsis [28] utilize the probability of upcoming extreme events in the Kolmogorov flow, introduced by earlier work [9], as a precursor for extreme events, and define a control law to act on the Fourier mode that contributes to imminent extreme events. These predictive and control methodologies are vital for controlling complex systems prone to extreme events. However, they are either computationally expensive at runtime (e.g., [27]) or remain system-dependent with scalability issues for high Reynolds number flows (e.g., [28]).

To address the above issues, we propose a three-stage framework for suppressing extreme events in turbulent chaotic flows, involving dimensionality reduction, precursor identification via a clustering-based algorithm, and real-time control in low-dimensional space. The proposed workflow is predominantly data-driven, with minimal requirements of physical priors for constructing a structured latent space, in contrast to system-dependent approaches such as [28]. Further, during runtime/evaluation, only an additional forward pass through a lightweight autoencoder network (i.e., few thousand parameters) per control step is required to estimate the control signal. We examine the stability of our approach on a case of forced turbulence, i.e., the Kolmogorov flow, to demonstrate its ability in mitigating extreme events in real time with minimal knowledge about the system dynamics. In addition, the framework is validated for scalability to a more chaotic flow regime with higher Reynolds number without incurring numerical issues reported for other approaches (e.g., [9,10]).

The rest of the paper is structured as follows. Section II introduces the physical system under consideration and the data generation process. It also defines our criteria for extreme events for this particular system. In Sec. III we introduce the three stages of our methodology in detail, while in Sec. IV we present and discuss our results for the different stages and how they affect the overall framework. Concluding remarks are then provided in Sec. V.

II. PROBLEM DEFINITION

A. Kolmogorov flow setup

The flow considered for this study is the Kolmogorov flow, i.e., 2D forced turbulence case [29,30]. Using a pseudospectral solver [31], the incompressible Navier-Stokes equations with spatially varying sinusoidal forcing are simulated on a 2D square domain $\Omega = [0, 2\pi] \times [0, 2\pi]$ with spatial variables (x, y) , subject to doubly periodic boundary conditions on a Cartesian grid. The governing momentum and continuity equations are given by

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \Delta \mathbf{u} + \mathbf{f}, \quad (1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (1b)$$

with body force $\mathbf{f} = [\sin(k_f y) \ 0]^T$ and wave number $k_f = 4$. As a first test case, we chose the Reynolds number $\text{Re} = 40$ to ensure that the resulting flow exhibits intermittent bursts in energy dissipation rate [30], which is defined as

$$\epsilon(t) = \frac{1}{\text{Re} \cdot |\Omega|} \int_{\Omega} |\nabla \mathbf{u}|^2 d\Omega. \quad (2)$$

Using a Fourier-Galerkin discretization approach [32], the solver is initialized with 128 collocation points in each spatial direction and integrated in time using a fourth-order Runge-Kutta (RK4) scheme with a constant time step $\Delta t = 0.0025$, similar to the setup in [9,28]. This configuration yields high-fidelity spectral data by well-resolved spatial and temporal dissipative scales. Simulations are initialized from a random initial condition, followed by a sufficiently long transient period of 200 time units before data collection begins. A total of N velocity field samples are recorded at a time interval of $\Delta t' = 0.1$ for 10 000 time units from a single trajectory and used as the training dataset. This results in a set of velocity states, $\mathcal{U} = \{\mathbf{u}^{(i)}\}_{i=1}^N$, ordered in time along with their corresponding energy dissipation rate values, $\epsilon = \{\epsilon^{(i)}\}_{i=1}^N$. A test or evaluation dataset of the same size is obtained from a trajectory with a different initial condition. The velocity fields are projected onto a Cartesian grid with resolution $N_x \times N_y$, such that the final shape of a single state is $\mathbb{R}^{2 \times N_x \times N_y}$. To balance accuracy with computational effort, we chose $N_x = N_y = 32$ to downsample the spatial information of the Kolmogorov flow, as it was shown that the mechanisms underlying extreme events in this flow are dominated by low-frequency modes [9]. While further downsampling may still be feasible, it could discard dynamically relevant structures for the extreme events of the Kolmogorov flow. Consequently, this may degrade the quality of the learned latent representation, clustering, and ultimately the control performance. We discuss the significance of this downsampling step by comparing directly against a full resolution case (i.e., $N_x = N_y = 128$) in terms of accuracy in flow reconstruction and state identification using the clustering algorithm in Appendix C.

B. Extreme events in the Kolmogorov flow

Extreme events are here defined as states whose observables deviate from the mean value by a certain number of standard deviations n_σ . These sudden, large-amplitude changes in the flow statistics arise when a trajectory reaches an instability region on the background attractor, steering the system toward extreme event regions. In the Kolmogorov flow, prior to an extreme event, energy is transferred from the Fourier mode $a(1, 0)$ to the mean flow [i.e., the mode $a(0, k_f)$] through triad

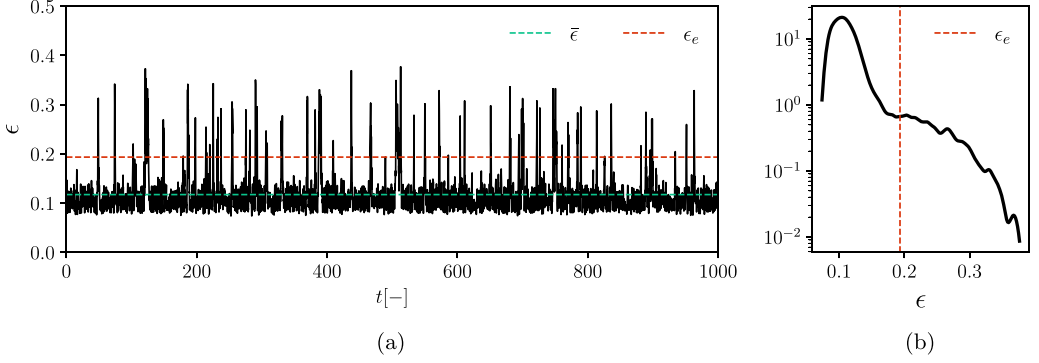


FIG. 1. (a) Energy dissipation time series of a sample trajectory highlighting mean and extreme events threshold. (b) Corresponding heavy-tailed distribution (probability density function) of the observable ϵ with intermittent bursts.

interactions of the modes $a(0, k_f)$, $a(1, 0)$, and $a(1, k_f)$ [9], where $a(k_x, k_y)$ denotes the Fourier coefficient corresponding to the wave number vector $\mathbf{k} = (k_x, k_y)$. The increase in the mean-flow energy consequently leads to an increase in the energy input rate followed by a burst in the energy dissipation rate. These findings are not only key to understanding the mechanisms underlying the emergence of extreme events but also provide insight into possible control strategies to avoid such undesirable phenomena.

In this study, the observable exhibiting intermittent bursts is the energy dissipation rate, and we consider an extreme event threshold of $\epsilon_e = \bar{\epsilon} + n_\sigma \sigma(\epsilon)$, where $\bar{\epsilon}$ and $\sigma(\epsilon)$ denote the temporal mean and standard deviation, respectively, of ϵ computed over a sufficiently long time series with converged statistics. A typical threshold is defined using $n_\sigma = 2$ [9,28], leading to $\epsilon_e = 0.193$. However, for clustering and control optimization, we employ a more conservative threshold $\tilde{\epsilon}_e$ using $n_\sigma = 1.5$, such that $\tilde{\epsilon}_e = 0.174$ to account for accumulating errors arising from both the clustering and optimization processes. Nonetheless, in evaluation mode, we use ϵ_e . Figure 1 shows a sample time series for energy dissipation and the corresponding distribution, which is characterized by a heavy tail distribution due to the presence of rare extreme events.

Figure 2(a) presents the time evolution of ϵ for a sample extreme event, comparing the system dynamics with and without control for the same initial condition. The corresponding vorticity fields ($\boldsymbol{\omega} = \nabla \times \mathbf{u}$) at different time instances for both cases are shown in Fig. 2(b). In the uncontrolled case, the magnitude of the maximum vorticity increases over time, leading to pronounced gradients

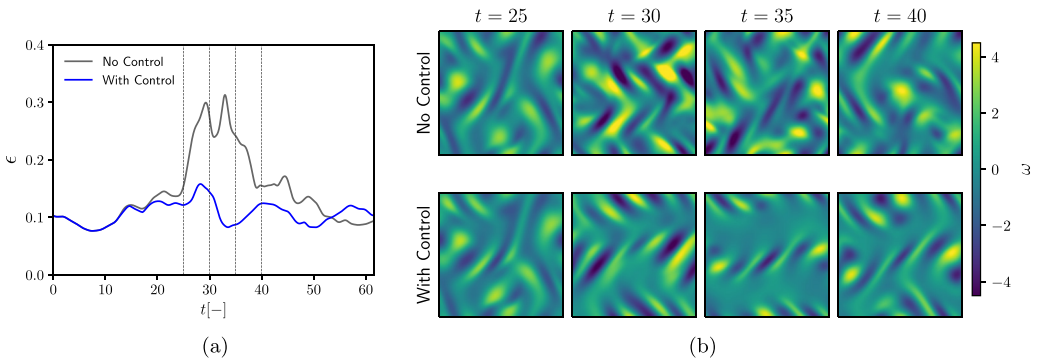


FIG. 2. (a) Time evolution of energy dissipation during an extreme event with and without control. (b) Snapshots of the vorticity field at select time steps for both cases.

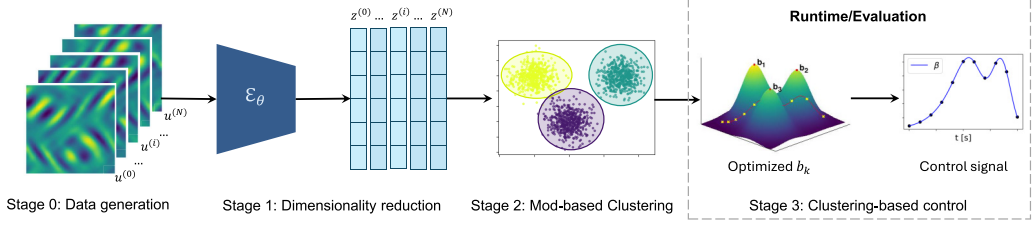


FIG. 3. Overview of the main stages of our proposed methodology.

that contribute to elevated energy dissipation values (e.g., at $t = 30$). In contrast, the controlled case does not exhibit such behavior and, therefore, maintains lower energy dissipation rate. The approach employed to achieve this control will be discussed in the next few sections.

III. METHODOLOGY

We present a predominantly data-driven approach for suppressing extreme events in turbulent chaotic flows consisting of three main stages. An overview of these stages is illustrated in Fig. 3. In the first stage, high-dimensional data in state space are compressed to a low-dimensional phase space, thereby rendering subsequent steps computationally feasible. The second stage is a clustering-based algorithm for the identification of precursors to extreme events, enabling real-time identification of state types. In the final stage, a parametrized control law, directly influencing the dynamics of the system through an additional body force, is defined and optimized in the low-dimensional phase space by assigning actuation values to clusters identified in the previous step. The components pertaining to the different stages are optimized once for the system under consideration and can then be utilized in evaluation mode without further retraining. Each of these components is described in detail in the following sections.

A. Dimensionality reduction

Dimensionality reduction methods are widely employed in fluid dynamics to extract low-dimensional representations of inherently high-dimensional flow fields. Classical linear techniques, such as Proper Orthogonal Decomposition (POD) [33,34] and Dynamic Mode Decomposition (DMD) [35], are commonly used to isolate dominant modes from data, thereby facilitating reduced-order modeling and flow reconstruction. These approaches are particularly effective when flow dynamics can be captured by a linear superposition of modes representing the most energetic or dynamically relevant features.

Recent advances have focused on nonlinear and machine learning–based dimensionality reduction techniques to address the limitations of linear approaches such as POD and DMD. While linear approaches may fail to represent strongly nonlinear or transient dynamics accurately, nonlinear methods, including deep learning approaches such as autoencoders (AEs), have demonstrated the ability to learn compact latent variables that preserve spatial coherence of complex flows [36]. In this study, we consider both a linear (POD) and a nonlinear model (AE) to compress high-dimensional input states to a compact latent representation and compare their performance in terms of information encoding and reconstruction accuracy. We adopt the implementation of POD as a linear AE (i.e., without nonlinear activation functions or biases), since the latent space spanned by POD coincides with that of a linear AE, up to the orthonormality properties of the POD basis, which can be recovered in linear AEs [37].

An AE is an artificial neural network composed of an encoder, $\mathcal{E}_{\theta_E}$, and a decoder, $\mathcal{D}_{\theta_D}$, network, parameterized by the learnable parameters θ_E and θ_D . The encoder network compresses the high-dimensional flow field $\mathbf{u} \in \mathbb{R}^{2 \times N_x \times N_y}$ to a low-dimensional latent space variable $\mathbf{z} \in \mathbb{R}^{N_d}$, where

TABLE I. Encoder and decoder architectures of the CAE. The hyperbolic tangent activation function is used between layers for nCAE, except for the final layer.

Encoder			Decoder		
Layer	Input Size	Output Size	Layer	Input Size	Output Size
Conv 3×3	$2 \times 32 \times 32$	$32 \times 16 \times 16$	Linear	N_d	2048
Conv 3×3	$32 \times 16 \times 16$	$64 \times 8 \times 8$	Unflatten	2048	$128 \times 4 \times 4$
Conv 3×3	$64 \times 8 \times 8$	$128 \times 4 \times 4$	ConvTranspose 3×3	$128 \times 4 \times 4$	$64 \times 8 \times 8$
Flatten	$128 \times 4 \times 4$	2048	ConvTranspose 3×3	$64 \times 8 \times 8$	$32 \times 16 \times 16$
Linear	2048	N_d	ConvTranspose 3×3	$32 \times 16 \times 16$	$32 \times 32 \times 32$
			Conv 3×3	$32 \times 32 \times 32$	$2 \times 32 \times 32$

$N_d \ll 2N_x N_y$. Conversely, the decoder network reconstructs a flow field $\hat{\mathbf{u}} \in \mathbb{R}^{2 \times N_x \times N_y}$ from an input latent space variable, $\mathbf{z}$. The forward pass of an AE is then described by

$$\mathbf{z}(t) = \mathcal{E}_{\theta_E}(\mathbf{u}(t)), \quad (3a)$$

$$\hat{\mathbf{u}}(t) = \mathcal{D}_{\theta_D}(\mathbf{z}(t)). \quad (3b)$$

Training an AE has the objective of optimizing the accuracy of the reconstructed field, $\hat{\mathbf{u}}$, by minimizing the reconstruction loss

$$\mathcal{L}_{AE} = \mathbb{E}_{\mathbf{u} \sim \mathcal{U}}[\|\hat{\mathbf{u}} - \mathbf{u}\|_2^2] \approx \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{u}}^{(i)} - \mathbf{u}^{(i)}\|_2^2. \quad (4)$$

The same loss function and dataset are used to train both the linear and nonlinear AEs. As previously outlined, the only difference between the two networks is in the network structure; nonlinear AE uses nonlinear activation functions (hyperbolic tangent in our case) between the different layers as well as biases, whereas the linear AE does not. Both networks utilize convolutional layers and are henceforth referred to as linear convolution AE (lCAE) and nonlinear convolution AE (nCAE). Details regarding the network structures are provided in Table I. For $N_x = N_y = 32$, a CAE based on this architecture has ca. 220 000 parameters for both θ_E and θ_D .

Symmetry-aware autoencoders

Several dynamical systems, such as the Kolmogorov flow, that are governed by physical equations can exhibit equivariance under symmetry transformations. In other words, if $\mathbf{u}(t)$ is a solution for Eq. (1) and $\mathcal{G}$ denotes a symmetry operation (e.g., translation or reflection), equivariance implies that $\mathcal{G}\mathbf{u}(t)$ is also a solution to the same governing equations. Hence, the set of solutions is invariant under $\mathcal{G}$. The aforementioned dimensionality reduction methods are inherently unaware of the invariant solutions present in a dataset, and, therefore, encode them into distinct regions of the latent space [38,39]. Eventually, this results in an inflated latent space and challenges for subsequent clustering and control due to the distribution of equivariant states in different regions of the latent space. To that end, it is crucial to ensure that the latent space is properly structured by incorporating this information during the training of the AEs.

The Kolmogorov flow under consideration exhibits multiple symmetries, as discussed by Kneer *et al.* [38]. It is equivariant under the group of continuous translations $[\mathbf{T}(s) : \mathbf{u}(x, y) \rightarrow \mathbf{u}(x + s, y)]$ and discrete symmetry groups [shift-and-reflect, $\mathbf{S} : \mathbf{u}(x, y) \rightarrow \mathbf{u}(-x, y + \frac{\pi}{4})$, and rotation, $\mathbf{P} : \mathbf{u}(x, y) \rightarrow \mathbf{u}(-x, -y)]$. A symmetry group, denoted $\mathbf{H}$, derived from unique combinations of these symmetry groups has a total of 16 distinct symmetries for this dynamical system.

To train an AE that incorporates knowledge of invariant solutions, we leverage the symmetry-aware autoencoder architecture proposed by Kneer *et al.* [38]. The network consists of several

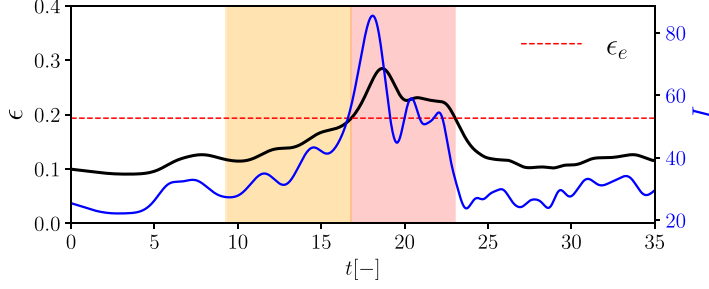


FIG. 4. Sample extreme event (red background) and a corresponding precursor (orange background) with an arbitrarily long lead time. The plot shows a burst in energy dissipation ϵ preceded by a burst in input energy I .

stages designed to map invariant solutions to a single point in the latent space, thereby enhancing their structural representation and separability. First, we utilize a Siamese architecture comprising 16 branches, each receiving a copy of the input. For each branch j , a discrete symmetry $\mathbf{H}_j$ is applied. Second, the transformed states are passed through a spatial transformer network (STN) [40] to extract a continuous transformation parameter s which is used for the translation function $\mathbf{T}(s)$. The same AE is then applied to each of the branches to reconstruct the transformed states. The branch j^* , which results in the lowest reconstruction error, is selected for the final step of recovering the initial input $\mathbf{u}$ through the inverse transformation $\mathbf{T}(s)^{-1}$ followed by $\mathbf{H}_{j^*}^{-1}$ to obtain the reconstructed solution $\hat{\mathbf{u}}$. More details regarding this multistage architecture and its application to various dynamical systems can be found in [38].

To benefit from the desirable properties of the latent space obtained from symmetry-aware AEs, we also consider symmetry-aware nonlinear CAE (SA-nCAE) in addition to lCAE and nCAE for dimensionality reduction.

B. Clustering-based identification of precursors

In the latent space obtained from the dimensionality reduction techniques discussed in the previous section, we then employ a data-driven approach for identifying precursors of extreme events in arbitrary chaotic systems based on maximizing a modularity parameter [24,41,42]. A sample “ideal” outcome of this stage, where normal, precursor, and extreme states are identified, is demonstrated in Fig. 4. This approach operates in a phase space with reduced dimensionality (latent space with dimension N_d obtained through an AE in our study) to group states of different types together, enabling the identification of normal, precursors, and extreme states in the reduced phase space. The continuous phase space is first tessellated into high-dimensional volumes, referred to as *hypercubes*. Each state in a trajectory is then assigned to a hypercube. A hypercube is labeled as extreme if it contains at least a single point that crosses the extreme event threshold (i.e., $\epsilon_t \geq \epsilon_e$). Precursor clusters contain states that transition into extreme ones, whereas normal clusters represent the stable regions of the background attractor.

The tessellated phase space is then used to construct a transition probability matrix $\mathbf{P}$ where each entry, denoted P_{ij} , is the probability of transitioning from hypercube B_i to B_j and is defined as

$$P_{ij} = \frac{m(B_i \cap \mathcal{F}^{-1}(B_j))}{m(B_i)} \quad i, j = 1, \dots, N_B, \quad (5)$$

where $\mathcal{F}^{-1}$ is the preimage, $m(B_i)$ is the number of discretized trajectory points in hypercube B_i , and N_B is the number of hypercubes on the system trajectory. The tessellation resolution depends on the number of hypercubes per dimension M and results in a total of $\hat{M} = M^{N_d}$ hypercubes. In practice, however, the dimensionality of the latent space is expanded by an additional dimension, ϵ , which is used by the clustering algorithm to define extreme events; thus $\hat{M} = M^{N_d+1}$. Although this transition

matrix grows exponentially as N_d increases and therefore requires massive space in memory, it is usually a sparse matrix that is diagonally dominant, enabling a memory- and compute-efficient sparse representation. For the following steps, this sparse transition matrix is reinterpreted as a directed graph, with hypercubes as nodes and transition probabilities as edge weights.

The clustering approach, called modularity-based clustering, aims to partition the network into communities by maximizing the modularity parameter, which measures deviations from a random baseline distribution of edges. In other words, it quantifies how well the network is partitioned such that nodes within the same community are more densely connected to each other than would be expected from a random distribution. The graph representation of the transition probability matrix $\mathbf{P}$ is iteratively divided into communities until no further modularity improvement is possible or until user-specified thresholds are exceeded. It should be noted that modularity-based clustering automatically identifies the final number of clusters and does not require specifying it *a priori* like in k -means ++. The resulting clustered graph can then be reinterpreted as a graph and it can then be reclustered with modularity-based clustering. We repeat this process until either the number of identified clusters falls below 100 or the number of iterations reaches 4. These values were chosen as they provide a balance between keeping a large enough number of clusters to accurately describe the flow transitions while not having too many clusters to keep the dimensionality of the control optimization problem small. Similarly, the maximum number of iterations is chosen to be four as the clustering algorithm usually converges to a desirable number of clusters in around three iterations. For further details regarding this approach, the reader is referred to Refs. [24,41].

The outcome of the algorithm is a set of communities/clusters, which are labeled normal, precursor, or extreme, that are utilized for real-time state identification. For any latent space point $\mathbf{z} \in \mathbb{R}^{N_d}$, a labeling function $f_\ell : \mathbb{R}^{N_d} \rightarrow \{\text{normal, precursor, extreme}\}$ assigns the state type based on these precomputed communities in the latent space of the pretrained AE. Our implementation of this function is based on the Euclidean distance (i.e., ℓ_2 norm) to the nearest cluster centroid $\mathbf{c}_k$ such that $f_\ell(\mathbf{z}) = \ell(k^*)$, where $k^* = \operatorname{argmin}_k \|\mathbf{c}_k - \mathbf{z}\|_2$ and $\ell(k)$ is the type of cluster k .

The main objectives of effective clustering, which subsequently facilitate efficient control, are: (1) achieving high accuracy in identifying precursors with few false negatives, and (2) maximizing the time spent in a precursor cluster before a subsequent extreme event, which we refer to as the *lead time*. These conditions ensure that we are able to adequately predict extreme events in advance, thereby providing the control law with sufficient time to smoothly steer the system away from an extreme event and toward a stable region of the background attractor. Therefore, we consider these criteria in our selection of the optimum clustering, as will be discussed in Sec. IV.

C. Clustering-based control

1. Control law

We adopt the cluster-based control approach by Nair *et al.* [43] to define a control law parameterized by actuation values associated with each cluster. After obtaining N_{cl} clusters from the previous step, a constant actuation value b_k per cluster is defined. The final control law interpolates these constant actuation values over the phase space through normalized radial basis kernel, resulting in a scalar parameter β which varies in time according to the evolution of $\mathbf{z}(t)$ in phase space, and is defined as

$$\beta(b_k, \mathbf{z}(t)) = \frac{\sum_k b_k e^{-\|\mathbf{z}(t) - \mathbf{c}_k\|^2 / \sigma_l^2}}{\sum_k e^{-\|\mathbf{z}(t) - \mathbf{c}_k\|^2 / \sigma_l^2}}, \quad \forall k = 1, \dots, N_{cl}, \quad (6)$$

with intercluster variance defined as $\sigma_l^2 = 1/N_{cl} \sum_k N_k \|\mathbf{c}_k - \bar{\mathbf{c}}\|_2^2$, where $\bar{\mathbf{c}}$ is the entire trajectory centroid, and N_k being the number of points in cluster k . The control signal β is used to define the control force $\mathbf{f}_{CT} = -[\beta \ 0]^T \sin(k_f y)$, which is an additional term to be added to the momentum equation [Eq. (1a)]. We chose $\mathbf{f}_{CT}$ to be of the same functional form as $\mathbf{f}$ [see Eq. (1a)], but parameterized by β and with a negative sign. Accordingly, $\mathbf{f}_{CT}$ directly reduces the energy input

rate, defined as $I(t) = \frac{1}{|\Omega|} \int_{\Omega} \mathbf{u} \cdot \tilde{\mathbf{f}} d\Omega$, where $\tilde{\mathbf{f}} = \mathbf{f} + \mathbf{f}_{CT}$. This enables the control law to avoid bursts in energy dissipation which are preceded by an increase in the energy input rate; e.g., see Fig. 4.

The clustering algorithm provides N_{cl} clusters, including normal, precursor, and extreme ones. While it follows naturally that the actuation values for the normal clusters should be disregarded by setting $b_k = 0$ for all $\ell(k) = \{\text{normal}\}$, we do not impose this constraint. This choice is primarily motivated by anticipated inaccuracies from the clustering algorithm, resulting in a non-negligible percentage of false positives and false negatives for the identification of precursors and extreme events. In the presence of false positives, some of the actuation values for the normal clusters might generate nonzero control signals to compensate for such inaccuracies. Another advantage of this choice is the greater flexibility in shaping the optimum control signal within the latent space by considering all the actuation values for the optimization process. Moreover, there are no constraints imposed for b_k . However, β is limited to the domain $[0, \beta_{\max}]$ to avoid a positive control force $\mathbf{f}_{CT}$ as well as excessive force magnitudes, which encourages late and abrupt control signals. However, during evaluation this range can be manipulated without further optimization. In our study we use $\beta_{\max} = 0.8$ during optimization.

The control law in Eq. (6) exhibits smooth transitions for β in the latent space. However, this smoothness depends on the encoding from the state space to the latent space to yield temporally smooth trajectories over time in the latter. Since the outcome depends on the method employed for dimensionality reduction, there is no guarantee that the control signal will remain smooth at all times. Therefore, in cases where it is necessary to mitigate abrupt changes in the control signal, which might have adverse effects on the flow dynamics, we employ a smoothing function inspired by cosine annealing [44], with a response time t_R , defined as

$$\beta(t, t_R, \beta_{\text{start}}, \beta_{\text{end}}) = \beta_{\text{end}} + \frac{1}{2}(\beta_{\text{start}} - \beta_{\text{end}}) \left[1 + \cos \left(\frac{\pi t}{t_R} \right) \right], \quad \text{for } t \in [t_0, t_0 + t_R], \quad (7)$$

and an arbitrary starting time t_0 . The use of such a smoothing function not only enhances stability by reducing the likelihood of introducing numerical artifacts into the system dynamics but also promotes physically realizable actuation by aligning the control strategy with physical constraints, such as controller or actuator latency. Consequently, this smoothing function serves as a control regularization mechanism that balances control effectiveness with practicality. The effect of the actuator latency, t_R , on the efficacy of control will be discussed in a subsequent section.

2. Loss function

We design a loss function for the optimization of the control law to simultaneously focus on the suppression of extreme events (by penalizing deviations from the background attractor mean statistics) and the cost reduction of the control signals required to achieve the former. Our loss function, therefore, is defined as

$$\mathcal{L}_{CT} = \frac{1}{T} \left(\underbrace{\gamma_{\epsilon} \int_0^T C_{\epsilon}(\epsilon_t) \left| \frac{\epsilon_t - \bar{\epsilon}}{\bar{\epsilon}} \right| dt}_{\text{Deviation from attractor mean}} + \underbrace{\gamma_{\beta} \int_0^T C_{\beta}(\mathbf{z}_t) |\beta_t| dt}_{\text{Cost reduction}} \right), \quad (8)$$

where the loss weights are selected to be $\gamma_{\epsilon} = 1$ and $\gamma_{\beta} = 10$, and the time-dependent coefficients are defined as

$$C_{\epsilon}(\epsilon_t) = \begin{cases} 20, & \epsilon_t \geq \tilde{\epsilon}_e \\ 10, & \epsilon_t \geq \epsilon_{\text{crit}} \\ 1, & \text{otherwise} \end{cases}, \quad \text{and} \quad C_{\beta}(\mathbf{z}_t) = \begin{cases} 10, & f_{\ell}(\mathbf{z}_t) = \text{normal} \\ 1, & \text{otherwise,} \end{cases} \quad (9)$$

which was found to be optimal for the current system and, thus, might differ for other flow regimes or chaotic systems. In the above, for conciseness, the subscript t is used to indicate time-dependency.

The first loss term in Eq. (8) penalizes energy dissipation values that deviate from the attractor mean $\bar{\epsilon}$. The piecewise constant coefficient $C_\epsilon(\epsilon_t)$ [Eq. (9)] scales extreme events and highly unstable precursors (here defined with threshold $\epsilon_{\text{crit}} = \bar{\epsilon} + n_{\sigma,\text{crit}}\sigma(\epsilon)$, where $n_{\sigma,\text{crit}} = 0.5$) with higher weights to prioritize them during optimization. These unstable precursors transition to extreme events in the short term, and thus are given higher priority compared to normal/precursor states where ϵ remains relatively closer to $\bar{\epsilon}$. This essentially helps speed up the convergence of the optimization process by manipulating the loss field to focus on losses with higher magnitudes. The second loss term aims to improve the cost efficiency of control by minimizing the magnitude of nonzero control forces. Similar to the first loss term, it utilizes a piecewise-constant coefficient $C_\beta(z_t)$, assigning higher weight to normal states to discourage control actions for states identified as normal by the labeling function f_ℓ . These time-dependent coefficients are not essential for a functional optimization and, therefore, can be set to unity. However, convergence rate is likely to be reduced as a consequence.

3. Gradient-based optimization

Enabled by a differentiable pseudospectral solver for the simulation of the Navier-Stokes equations, we leverage a gradient-based optimization approach. In contrast to alternative methods, such as gradient-free methods, this approach allows efficient propagation of information about how the system dynamics evolve in the long-term under the influence of a time-varying control signal, leading to better long-term stability and accuracy [45,46]. Thus, the absence of this capability renders other approaches less effective in capturing system responses over extended time horizons, which may eventually lead to unstable solutions.

In gradient-based optimization, the primary goal is to compute the loss sensitivity to the input parameters, $\nabla_{b_k} \mathcal{L}_{CT}$, which iteratively updates the inputs to minimize the loss. This gradient can be expressed through the chain rule by calculating the intermediate gradients for the entire trajectory as follows:

$$\nabla_{b_k} \mathcal{L}_{CT} = \frac{\partial \mathcal{L}_{CT}}{\partial \mathbf{u}(T)} \prod_{t=1}^T \frac{\partial \mathbf{u}(t)}{\partial \mathbf{u}(t-1)}, \quad (10)$$

where T is the simulation horizon. The per-step Jacobian $\frac{\partial \mathbf{u}(t)}{\partial \mathbf{u}(t-1)}$ is the product of Jacobians corresponding to intermediate stages, including dimensionality reduction $\mathbf{z}(t)$ [Eq. (3a)], control signal estimation $\beta(b_k, \mathbf{z}(t))$ [Eq. (6)], and time integration via the RK4 scheme. This decomposition explicitly captures how perturbations in the system state propagate through the encoder network, control policy, and numerical integration at each time step. The final gradient, $\nabla_{b_k} \mathcal{L}_{CT}$, is used to obtain an updated set of parameters $b_k^{(i+1)}$ from $b_k^{(i)}$ using a gradient-based optimization algorithm, such as gradient descent or Adam [47]. In this study the latter is employed for the optimization process with a constant learning rate of 0.01.

The temporal horizon T for optimization is ideally chosen such that it covers the entire duration of at least a single extreme event, along with a sufficiently long transition period to ensure a smooth control signal. Accordingly, the initial condition $\mathbf{u}(t=0)$ is selected such that, for the base case without control, the system exhibits an extreme event within the interval $t \in [0, T]$. For a time step size of $\Delta t = 0.0025$ and $T = 61.6$ time units (fitting an extreme event within the memory limits of the utilized GPU), this corresponds to 24 640 RK4 integration steps. Unrolling over such long temporal horizons poses significant challenges for gradient-based optimization, primarily due to the repeated multiplication of per-step Jacobians. When the dominant eigenvalues of these Jacobians are less than one, gradients tend to vanish exponentially, while eigenvalues greater than one can lead to exploding gradients, significantly impacting training stability and convergence, analogous

to training recurrent neural networks [48,49]. Such instabilities can lead to modeling errors and degraded policy performance over extended horizons.

To address these issues, we consider an optimization over only half the temporal horizon (i.e., $t \in [0, 0.5T]$) and with a bigger time step $\Delta t = 0.01$, resulting in a total of 3,080 time steps for each iteration—a reduction of 87.5% of the computational effort. Using a larger time step size for this short trajectory length was found to lead to negligible divergence relative to the trajectory with the finer time step, while the speed and efficiency of the optimization process are significantly improved.

4. Progressive optimization

Obtaining a set of optimized actuation values based on a single extreme event can often lead to few extreme events remaining during runtime, especially when unrolling for long temporal horizons. This implies that the control law may not generalize to different extreme event dynamics. Addressing this limitation, we optimize the control law by considering multiple extreme events obtained from the same or different trajectories. Nevertheless, optimizing with a set of extreme events $\mathcal{P}$ at once can be slow and inefficient, as observed during early experiments.

A better alternative is to incorporate these multiple extreme events in a progressive fashion. Therefore, we propose a progressive optimization algorithm (Algorithm 1) designed to improve control performance by gradually incorporating multiple extreme events during optimization. Instead of optimizing over all extreme events simultaneously, the method introduces them incrementally, which stabilizes convergence and enhances optimization efficiency. We also define a set of extreme events $\mathcal{P}_v$ to evaluate the efficiency and generalization capability of the optimized actuation values. This validation step is performed every few iterations in evaluation mode, i.e., using ϵ_e instead of $\tilde{\epsilon}_e$ as an extreme event threshold.

ALGORITHM 1. Progressive optimization algorithm using multiple extreme events.

Require: Solver $\mathcal{S}$, Set of initial conditions for extreme events $\mathcal{P}$, Unrolling time T_f

- 1: Initialize: $b_k, \tilde{\mathcal{P}} \leftarrow \{\}, N_{EE} \leftarrow 0$
- 2: **repeat**
- 3: **if** $N_{EE} = 0$ **and** $|\mathcal{P}| > 0$ **then**
- 4: $\tilde{\mathcal{P}} \leftarrow \tilde{\mathcal{P}} \cup \{\mathcal{P}.pop()\}$ ▷ Add initial condition to the active set from the main set
- 5: **end if**
- 6: $N_{EE} \leftarrow 0$ ▷ Reset the cumulative number of extreme events
- 7: **for** Each batch $\tilde{u}_0$ **in** $\tilde{\mathcal{P}}$ **do**
- 8: $\mathcal{L}_{CT}, n_{EE} \leftarrow \mathcal{S}(\tilde{u}_0, b_k, T_f)$ ▷ Unroll $\tilde{u}_0$ using b_k . n_{EE} : total number of extreme events
- 9: $N_{EE} \leftarrow N_{EE} + n_{EE}$
- 10: Take Adam step on $\nabla_{b_k} \mathcal{L}_{CT}$ ▷ Update b_k
- 11: **end for**
- 12: **until** converged

The algorithm begins by initializing the control parameters b_k and maintaining two sets of initial conditions: a pool $\mathcal{P}$ of preselected extreme events and an active set $\tilde{\mathcal{P}}$. At each iteration, if no new extreme events are detected ($N_{EE} = 0$) and unused conditions remain ($|\mathcal{P}| > 0$), one is added to the active set. Within each iteration, initial conditions are sampled from the active set $\tilde{\mathcal{P}}$ and are unrolled for the chosen temporal horizon ($0.5T$ time units in this study), during which the loss $\mathcal{L}_{CT}$ and the number of extreme events n_{EE} per trajectory are computed. The cumulative count N_{EE} is updated accordingly, and an Adam optimization step is performed on $\nabla_{b_k} \mathcal{L}_{CT}$. Optimization is perpetuated until convergence is achieved. This ensures that the control law progressively adapts to various scenarios only after previous ones have converged without any extreme events, ultimately leading to a more stable optimization.

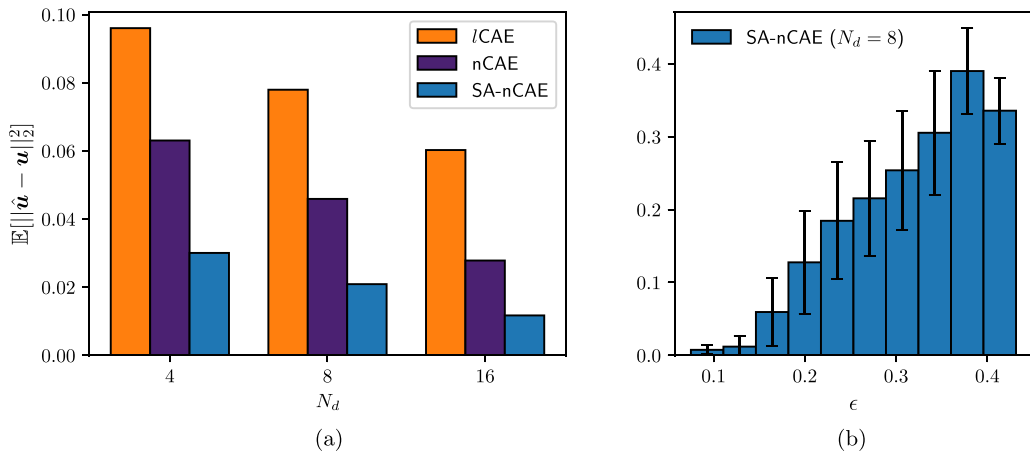


FIG. 5. Comparison of the reconstruction accuracy of the various AE types with different latent space dimension N_d (a) and the loss distribution over ϵ for the highest accuracy model, SA-nCAE, with $N_d = 8$ (b). Standard deviation per bin is displayed in (b).

Convergence assessment begins once the conditions in the set $\mathcal{P}$ are exhausted, i.e., no unused initial conditions remain. After every validation step, the optimization is considered converged when the relative change in the loss function remains below 1% for at least 10 consecutive iterations, indicating that the solution has reached a steady regime. In addition, the validation step must not produce any extreme events. The optimization is therefore terminated if this condition is violated for two consecutive validation steps, thereby preventing overfitting to the extreme events within the active set $\tilde{\mathcal{P}}$.

While it can be observed that our proposed multistep framework inherits hyperparameters from each stage involved, many of these hyperparameters are interdependent. They are typically selected based on considerations arising from downstream stages (e.g., N_d for CAEs should remain moderate to avoid imposing excessive computational costs on the subsequent clustering algorithm) or based on the performance of the respective stage (e.g., the clustering should be selected to maximize accuracy in state-type identification). Further, the values provided in this work can serve as a baseline, to be tuned for different flow regimes and configurations. For instance, when extending the framework to $Re = 200$, as will be shown in Sec. IV D, only a few parameters were adjusted to attain better convergence during the optimization of the control law.

IV. RESULTS

In this section, we present and discuss the results obtained using the methodology detailed in the previous section. We first compare the reconstruction accuracy achieved by each autoencoder (lCAE, nCAE, and SA-nCAE) in Sec. IV A followed by an analysis in Sec. IV B of how the clusters formed in the latent space of each AE influence the accuracy of precursor and extreme event identification. Finally, Sec. IV C presents the resulting control for the top-performing AE and demonstrates the overall effectiveness and practicality of the proposed approach.

A. Reconstruction accuracy of CAEs

The reconstruction accuracy, assessed by the reconstruction loss $\mathcal{L}_{AE}$ on the test dataset, for the models under consideration is summarized in Fig. 5(a). As expected, for small latent space dimensions, nonlinear CAEs outperform the lCAE across all N_d [36]. Among these, the most accurate model is the SA-nCAE due to its symmetry-aware architecture that enables it to identify invariant

solutions that minimize the reconstruction loss more effectively compared to a vanilla nCAE. The distribution of the loss across different ϵ ranges for this model [see Fig. 5(b)] demonstrates how the training accuracy correlates with the data distribution [cf. Fig. 1(b)]. The heavy-tailed distribution in Fig. 1(b), which is a result of rare extreme events, means that there is a significant imbalance in the sampling of states for model training. As a consequence, states with ϵ values near the tail of the distribution suffer the lowest reconstruction accuracy due to their rare presence in the training dataset.

Furthermore, note that we restricted our experiments to $N_d \leq 16$, albeit higher values of N_d would lead to a substantially higher reconstruction accuracy [38], as the growth of the latent space size will lead to prohibitively expensive computational and memory requirements for clustering, as outlined in Sec. III B. To balance between computational cost and accuracy, we select $N_d = 8$ for further investigation in later stages of our framework. However, despite having an outperforming SA-CAE model for this latent space size, we still need to examine its embedding space structure. This is because our primary objective is not only to maximize reconstruction accuracy but also to obtain a latent space where there is discernible separation between the different state types, facilitating effective clustering and efficient control.

B. State identification and latent space structure

We employ the standard true positive and false positive ratios to evaluate the accuracy in identifying extreme events and their associated precursors using clusters produced by the clustering algorithm. For extreme events, this is straightforward, as an extreme event is explicitly defined by the threshold ϵ_e (or $\tilde{\epsilon}_e$) and can therefore be directly assessed using these metrics. In contrast, no labels are available for precursors, since identifying them is the primary objective of the clustering algorithm. That said, we first introduce a formal definition of these evaluation metrics, which will be used consistently throughout the analysis.

Let $\hat{N}_{EwP}$ denote the number of identified extreme events preceded by a precursor, $\hat{N}_{EnP}$ the number of identified extreme events not preceded by a precursor, and $\hat{N}_{PnE}$ the number of identified precursors not followed by an extreme event. Using this notation, the true positives TP_P and false positives FP_P for precursors are defined as

$$TP_P = \frac{\hat{N}_{EwP}}{\hat{N}_{EwP} + \hat{N}_{EnP}} \quad \text{and} \quad FP_P = \frac{\hat{N}_{PnE}}{\hat{N}_{PnE} + \hat{N}_{EwP}}, \quad (11)$$

representing the proportion of precursors correctly associated with subsequent extreme events and the proportion of normal states erroneously classified as precursors, respectively. Similarly, let $\hat{N}_{TE}$ denote the number of correctly identified extreme events, N_E the total number of actual extreme events, and $\hat{N}_{FE}$ the number of falsely identified extreme events. The corresponding true positive TP_E and false positive FP_E rates for extreme events are then given by

$$TP_E = \frac{\hat{N}_{TE}}{N_E} \quad \text{and} \quad FP_E = \frac{\hat{N}_{FE}}{\hat{N}_{TE} + \hat{N}_{FE}}, \quad (12)$$

which quantify the fraction of extreme events correctly detected and the fraction of false alarms among all identified extreme events, respectively. With these evaluation metrics established, we proceed with presenting the results of applying the clustering algorithm to multiple test cases.

After obtaining embeddings of the test trajectory using the three CAEs (linear "l," nonlinear "n," and symmetry-aware nonlinear "SA-n") with $N_d = 8$, clustering is performed on a segment from the initial portion of the trajectory, and evaluation is carried out on the final 8000 time units. We compare different clustering parameters, namely trajectory length, tessellation resolution M , temporal stride $\Delta t'$, and identification approaches, including cluster- and distance-based approaches. Here the temporal stride $\Delta t'$ refers to the time step size of recording a trajectory to be considered for clustering, which is usually different from Δt , the time step size for unrolling the PDE in time using the RK4 scheme. In the cluster-based approach, a latent space variable $\mathbf{z}(t)$ is mapped to a

TABLE II. Clustering statistics obtained from top-performing runs under varying parameters and identification methods (cluster- and distance-based). Cluster-based identification results are based on the trajectory used for clustering, while distance-based identification is applied to the final 8000 time units of the test trajectory. Normal is abbreviated as N, Precursors as P, and Extreme as E. LD is short for lead time. TP and FP are expressed in percentages. The column “N-P-E” shows the number of clusters for each cluster type. The top performance for each metric across the different CAE types is highlighted in bold.

Case	CAE	M	$\Delta t'$	Time	N-P-E	Cluster-based		Distance-based				
						TP _p	FP _p	TP _p	FP _p	LD	TP _E	FP _E
L1	l	20	0.2	1000	84-4-5	80	0	31.58	84.87	4.817	82.86	1.17
L2	l	20	0.2	2000	84-12-15	92.31	0	64.62	73.58	3.169	88.02	5.27
L3	l	10	0.4	2000	56-12-10	92.31	14.29	59.65	82.20	4.576	86.85	2.68
V1	n	20	0.2	1000	66-4-7	80	0	32.88	78.76	4.412	76.40	5.21
V2	n	20	0.2	2000	87-12-12	92.31	0	67.95	74.89	3.245	81.89	6.42
V3	n	10	0.4	2000	56-13-9	92.31	7.70	71.25	75.54	3.980	81.29	6.04
S1	SA- n	50	0.2	1000	86-4-7	100	60	58.65	69.19	3.121	87.78	12.63
S2	SA- n	50	0.2	2000	82-12-14	92.31	0	84.70	64.12	2.763	88.98	24.35
S3	SA- n	30	0.4	2000	59-11-10	92.31	0	91.98	71.51	3.979	80.97	17.35

hypercube from the tessellation used to obtain the clusters in order to identify its type. However, this is only applicable for states that have been used for obtaining the clusters in the first place. For unseen states, the distance-based approach is adopted, which utilizes the labeling function f_ℓ , introduced in Sec. III B, to associate the point with the nearest cluster centroid. The top-performing results for several cases with these different configurations are reported in Table II.

The results reveal several important trends in the accuracy of state identification of the various AEs. Using the cluster-based approach, precursors can be identified with almost 100% true positive rate and no false positives for most cases using all CAEs. In other words, the clustering algorithm can yield significantly similar results using any CAE type and for any latent space representation. These results are, nonetheless, not attainable in real time, as this approach requires a full trajectory for *a priori* analysis. Instead, the distance-based approach underscores the advantage of using SA- n CAEs over other models in generalizing to unseen states and trajectories in real time. It is also particularly relevant from a control perspective, as the control signal depends on the actuation values associated with the nearest cluster centroids. Hence, we focus our analysis on the results obtained from the distance-based approach for state identification.

First, there exists a strong correlation between the temporal horizon of the utilized trajectory and the percentage of true positives for precursors. A relatively longer trajectory (e.g., 2000 instead of 1000 time units) is more representative of the general structure of the latent space and is thus beneficial in obtaining clusters that can generalize well (compare cases L1, V1, and S1 with others from Table II). On the other hand, the false positives percentage for extreme events consistently deteriorates with longer trajectories. Therefore, while there might be a potential to increase the accuracy in identifying precursors by considering a longer trajectory, we get a much higher false positives rate for extreme events. In contrast, neither the lead time nor the stride size ($\Delta t'$) shows a clear correlation with accuracy. Although using a larger stride reduces computational cost, owing to the smaller number of states available for clustering, it does not improve the accuracy of clustering. These findings suggest that temporal resolution primarily affects efficiency rather than predictive performance, whereas the length of the considered trajectory plays a more critical role in characterizing the latent space.

Furthermore, the number of normal clusters dominates across all cases, which is consistent with their prevalence in the dataset. It is also observed that configurations with longer trajectories yield

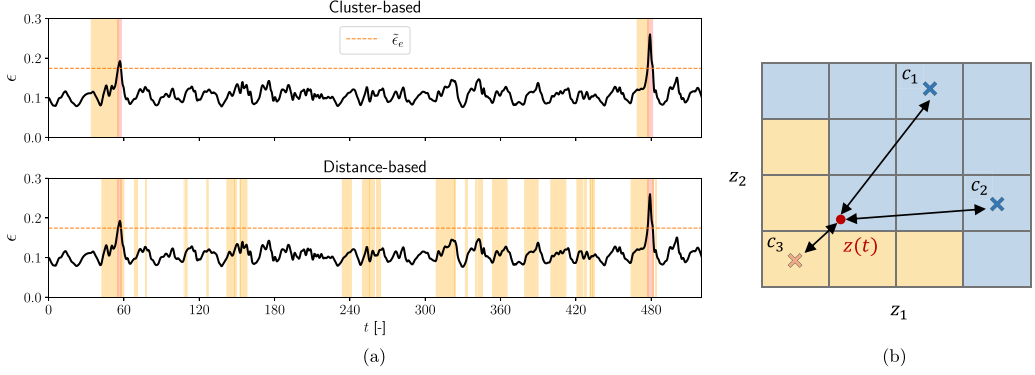


FIG. 6. Accuracy comparison of cluster- and distance-based approaches for the identification of precursors. (a) Sample time series of the energy dissipation with colored regions for precursors (orange-shaded) and extreme events (red-shaded) obtained using the cluster-based approach (top panel) and the distance-based approach (bottom panel). (b) Schematic of a two-dimensional latent space illustrating the difference in state identification using both approaches. The illustration demonstrates a false positive precursor for the distance-based approach due to distance minimization [i.e., $z(t)$ is closer to precursor cluster centroid c_3 than normal centroids c_1 and c_2] in comparison to the graph-based approach adopted by the clustering algorithm.

a higher number of precursors and extreme clusters, which correlates with better accuracy in state identification, as previously mentioned. The SA-nCAE model, in particular, demonstrates superior performance in case S3, where the combination of a proper tessellation resolution and the longest trajectory length results in the highest percentage of true positives for precursors ($\approx 92\%$). Notably, clusterings based on the SA-nCAE generally require higher tessellation resolution M to effectively delineate the distinct regions in the latent space.

On the downside, nearly all cases suffer from a very high false-positives rate for precursors. To elucidate the reason for the mismatch in the false positives rate between the distance- and cluster-based approaches, the schematic in Fig. 6(b) highlights why the distance-based approach can lead to many false positives for precursors. As communities are formed by the clustering algorithm, each hypercube in the tessellated phase space with at least a single point is assigned to a community based on the transition probability matrix P and its corresponding graph interpretation. This yields hypercubes which can be relatively distant from the corresponding cluster centroid but still belong to the same community. The cluster-based approach makes use of these pre-identified community assignments for the hypercubes and is, therefore, able to identify precursors and extreme events with little to no false positives. The distance-based approach, on the other hand, relies solely on information about the cluster centroids, which can falsely identify normal states as precursors, as seen in Fig. 6(a).

Moreover, the false-positive rate for precursors is considerably higher than for extreme events (see Table II). This stems from the fact that precursors are, by definition, normal states whose energy dissipation rate lies below the extreme-event threshold, yet they reside in the unstable region of the background attractor and can transition into extreme events. Among all normal states, only a small subset qualifies as true precursors, namely those that lie within unstable regions of the background attractor and are followed by extreme states. As a result, the probability of a normal state being mislabeled as a precursor is much higher than the probability of a normal/precursor state being labeled as extreme. Consequently, the rate of false positives for precursor identification are comparatively much higher than for extreme states.

A direct consequence of these high rates of false positives, in principle, would be an overly conservative control law with reduced efficiency. However, in practice, this does not necessarily degrade our closed-loop performance, as will be seen in the next section. The control law is

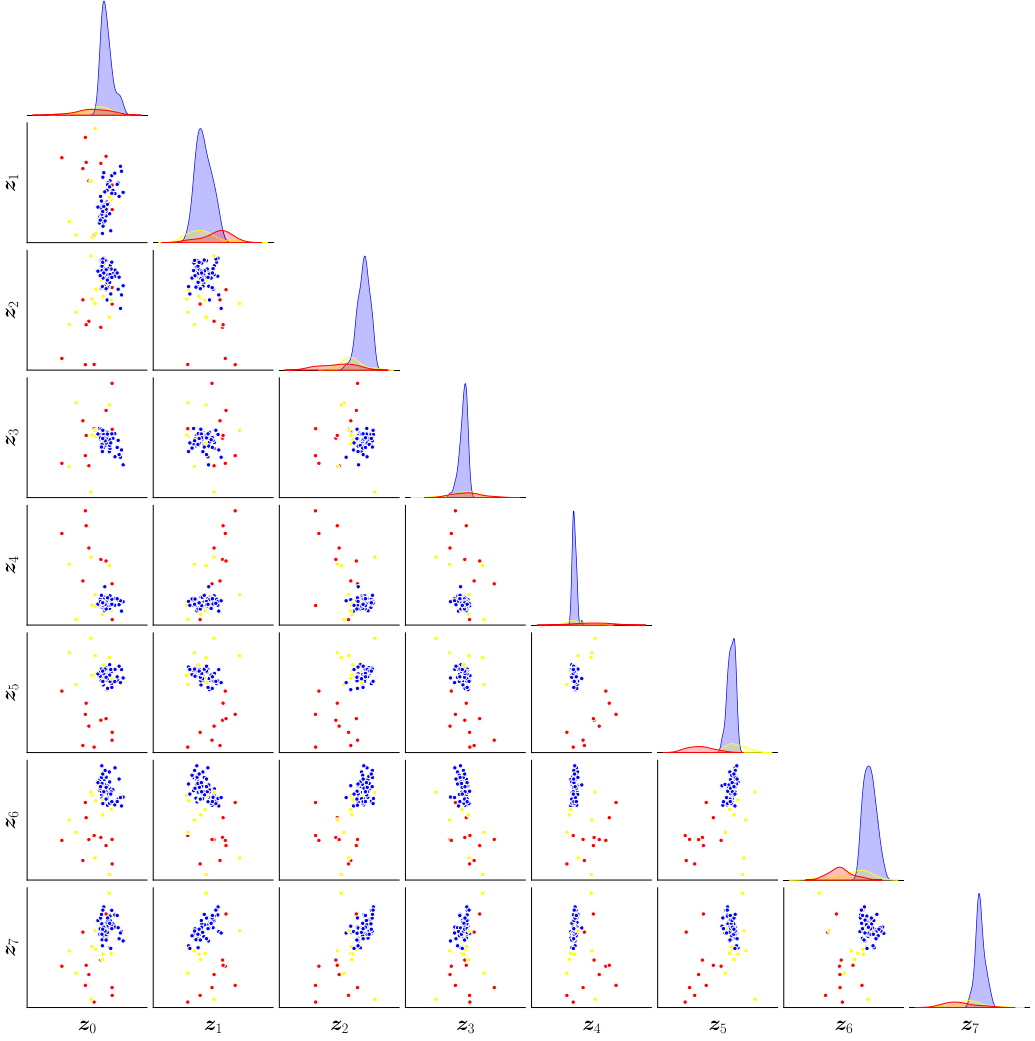


FIG. 7. Cluster centroids visualized in the latent space for the Case S3 of SA-nCAE. Normal, precursor, and extreme clusters are colored blue, orange, and red, respectively. Diagonal plots show the kernel density estimation (KDE) for each dimension z_k .

optimized with a control-cost loss term [second term in Eq. (8)], balancing actuation effort against control benefit. Consequently, even if many precursor detections are spurious, the control law is optimized to act only when such activations lead to net performance improvements. Thus, the control-law optimization may eventually compensate for inaccuracies introduced by the clustering algorithm.

Furthermore, the centroids of the clusters corresponding to the cases with the highest true positives for precursors are visualized in Figs. 7, 16, and 17 (Appendix A) for SA-nCAE, lCAE, and nCAE, respectively. These visuals highlight the capability of the SA-nCAE (see Fig. 7) to produce a clear delineation between the different cluster types compared to lCAE and nCAE. The high accuracy in state identification for SA-nCAEs can be attributed to this well-structured latent space, which in turn enhances control performance by ensuring minimal disruptive overlap between the actuation signals of each peak. To elaborate, the spread of the actuation signal is dictated by the

intercluster variance, j_i , as seen in Eq. (6). Consequently, the separation between the different state types implies that the actuation values have minimal influence on one another, as they are relatively distant in the latent space.

Based on these results, we adopt clustering case S3 for the remainder of this study, as it satisfies the objectives specified in Sec. III B, namely high percentage of true positives for precursors and mean time from precursors to extreme events, in addition to a well-structured latent space. Moreover, we compare this case in terms of runtime and performance across various latent sizes in Appendix B, further supporting the choice of $N_d = 8$.

C. Control

We initialize the optimization by sampling b_k from a normal distribution with zero mean and a covariance of 0.1, i.e., $b_k \sim \mathcal{N}(0, 0.1 \cdot \mathbf{I})$. We also chose $|\mathcal{P}| = 5$ and $|\mathcal{P}_v| = 15$, as we found during early experiments that once convergence is achieved for $\mathcal{P}$ on these five extreme events, the validation step using $\mathcal{P}_v$ usually converges as well. Thus, a larger set for optimization is unnecessary. During evaluation, we use an RK4 time step size of $\Delta t = 0.0025$. Unless otherwise stated, β is updated every $4\Delta t$ time units, i.e., $\Delta t_\beta = 4\Delta t$, representing the controller delay or the time required for the controller to generate a new control signal for the actuator, with response time $t_R = 1\Delta t$ (i.e., instantaneous control) that represents actuator latency.

1. Influence of latent space structure on control

We first investigate the control performance using the SA-nCAE and nCAE. The lCAE is not considered in our comparison, as it exhibited a similar (unstructured) latent space and lower accuracy in state identification compared to nCAE, as discussed in the previous sections. Its result is, therefore, anticipated to be redundant.

Using the SA-nCAE for encoding and the clustering of case S3, the optimization converged after 140 iterations with a validation step performed every five iterations. In contrast, using nCAE and case V3 for clustering, the optimization failed to cover all extreme events in the main set $\mathcal{P}$ for $\gamma_\beta = 10$ and 5, as the loss contribution from the control signal substantially exceeded that from the dissipation loss. For $\gamma_\beta = 1$, although the set $\mathcal{P}$ was exhausted, convergence was not achieved after more than 1000 iterations. Nevertheless, the control exhibited improved generalization by effectively suppressing all extreme events in the validation set except one. While convergence might have been attainable with a different selection of extreme events for optimization and validation, for the sake of a fair comparison, we evaluate the results obtained at 1000 iterations, which yielded the lowest validation loss.

For the SA-nCAE-based control, Fig. 8 shows the time series evolution of the dissipation rate and the control signal β . In the top panel of Fig. 8(a), we observe how the controlled time series of the energy dissipation does not exhibit high peaks compared to the uncontrolled case and remains close to the mean value. The implementation of the control strategy led to a substantial reduction in the occurrence and intensity of extreme events. Without control, extreme events constituted 4.74% of the time of the entire trajectory, whereas with control, their frequency decreased to only 0.028%, corresponding to an overall reduction of approximately 99.4%. The few extreme events that persisted under control exhibited significantly lower magnitudes of energy dissipation rate (slightly above ϵ_e), indicating that the control mechanism not only suppresses their frequency but also effectively mitigates the severity of those that occur with much less frequency, as depicted in Fig. 8(b). A closer look of the first extreme event encountered in this evaluation trajectory, along with the corresponding vorticity fields at different time steps, is shown in Fig. 2.

Moreover, in the bottom panel of Fig. 8(a), the β control signal can be seen to exhibit multiple peaks, with a maximum value ranging between 0.2 and 0.6, corresponding to potential extreme events that were successfully suppressed without overshooting, i.e., without leading to relatively low energy dissipation rates. Furthermore, although the time series plot of β shows false positive

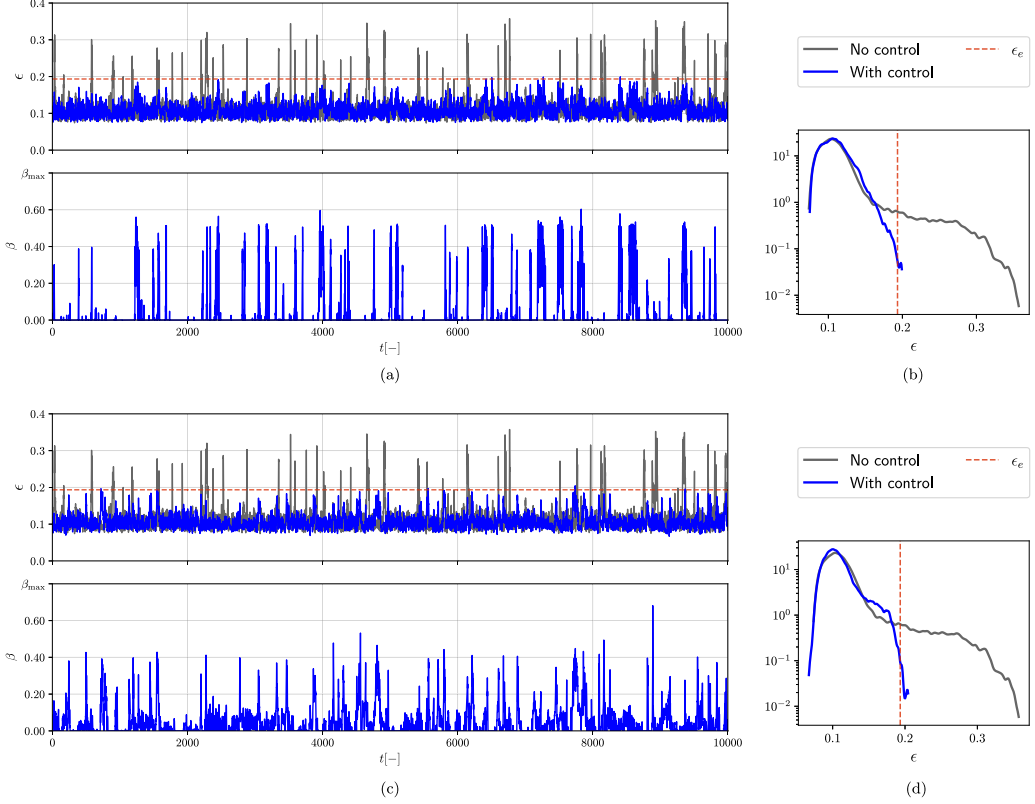


FIG. 8. Evaluation results of control obtained using SA-nCAE (a), (b) and nCAE (c), (d) encoding. The top panels in (a) and (c) show the time series for energy dissipation with and without control overlaid. The bottom panels show the corresponding control signal β for the controlled trajectory. The probability density function for the energy dissipation rate is shown before and after control for SA-nCAE (b) and nCAE (d).

control signals which might seemingly indicate perpetual control, the percentage of time where $\beta = 0$ is found to be $\approx 83\%$, implying that the vast majority of time is spent without control.

Under the nCAE-based control, the frequency and intensity of extreme events for the same trajectory was reduced to 0.0445%, as shown in Fig. 8(d), representing a reduction of approximately 99.1%. While this is comparable to the performance achieved by the SA-nCAE, the control signal for the nCAE remained active ($\beta > 0$) for 58.15% of the time [see Fig. 8(c), bottom panel], whereas it was active for only 17% of the time in the SA-nCAE case. This observation further supports the conclusion drawn from the clustering accuracy analysis based on the distance-based method, underscoring that a symmetry-unaware latent space can result in control signals that are unnecessarily active due to the spatial overlap among different state types within the latent space. Based on these results, the remainder of this section focuses on analyzing the results obtained using the SA-nCAE.

Moreover, we discuss our framework's performance in suppressing extreme events against existing literature. Farazmand and Sapsis [28] achieved complete suppression of extreme events without reporting the associated control cost required to obtain this result, while Racca and Magri report reductions in the occurrence of extreme events by up to one order of magnitude [23] and two orders of magnitude [27] in studies involving a different turbulent flow, described by ordinary differential equations. On the other hands, our framework achieves a reduction to 0.028% ($\approx$ two orders of magnitude) on an arguably more complex flow system, surpassing [23] and [27], and

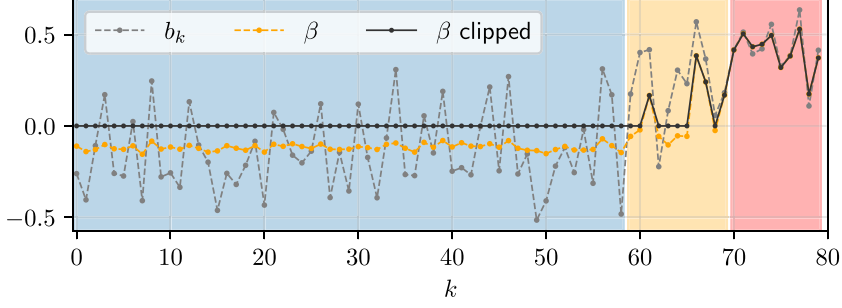


FIG. 9. Optimized actuation values b_k and control signal β (before and after clipping to the domain $\beta \in [0, \beta_{\max}]$) at each cluster centroid. Normal, precursor, and extreme clusters are highlighted with blue, orange, and red backgrounds, respectively.

matching [28]. A more detailed presentation of the computational cost of our framework, along with a comparative discussion with these other approaches, is provided in Appendix D.

2. Analysis of optimized actuation and control signals

We visualize the set of optimized actuation values, b_k , along with the corresponding control signal, β , at each cluster centroid in Fig. 9. Several insights can be drawn from this visualization. First, the control law based on radial basis functions exhibits a high correlation between b_k and β but almost always leads to different quantities for both variables due to the inherent nonlinearity of the control law. Second, for normal clusters, while b_k assumes both negative and positive values, the clipped β remains zero. This is an anticipated outcome of the optimization process, as the second loss term in Eq. (6) heavily penalizes any nonzero values of β for normal states. Conversely, the actuation values for the extreme clusters are substantially greater than zero, yet remain below the upper limit $\beta_{\max}$, which is also an anticipated result of the optimization procedure.

Nevertheless, for precursor clusters, most clusters have $\beta = 0$, which is unexpected. Figure 10(a) explains why this outcome should not be taken at face value. By analyzing the set $\mathcal{P}$ of extreme events utilized for optimization, Fig. 10(a) shows the mean control signal β associated with each cluster based on the distance-based method. This analysis confirms that most control signals corresponding to precursor clusters are greater than zero, except for clusters $k \in \{59, 62, 64\}$ where $\beta \approx 0$. The reason for this behavior is outlined in Table III, which makes use of the following

TABLE III. Insights into transitions from precursor clusters to normal and extreme ones, obtained after convergence on the set $\mathcal{P}$ used for the optimization. Clusters with $\beta \approx 0$ and correspondingly high probabilities of transiting to normal clusters are underlined, whereas those with $\beta > 0$ and correspondingly high probabilities of transitioning to extreme clusters are highlighted in bold.

k	$\text{Pr}_{\text{final}}(\text{N} k)$	$\text{Pr}_{\text{direct}}(\text{N} k)$	$\text{Pr}_{\text{direct}}(\text{E} k)$	β
59	<u>0.91</u>	<u>0.39</u>	0.00	<u>0.01 ± 0.03</u>
60	0.14	0.00	0.00	<u>0.03 ± 0.03</u>
61	0.04	1.00	0.59	0.18 ± 0.12
62	<u>0.98</u>	<u>1.00</u>	0.00	<u>0.00 ± 0.00</u>
64	<u>0.53</u>	<u>1.00</u>	0.00	<u>0.00 ± 0.00</u>
66	0.00	—	0.00	0.19 ± 0.07
67	0.27	1.00	0.88	0.23 ± 0.14
68	0.03	1.00	0.67	0.08 ± 0.09
69	0.00	—	0.90	0.21 ± 0.08

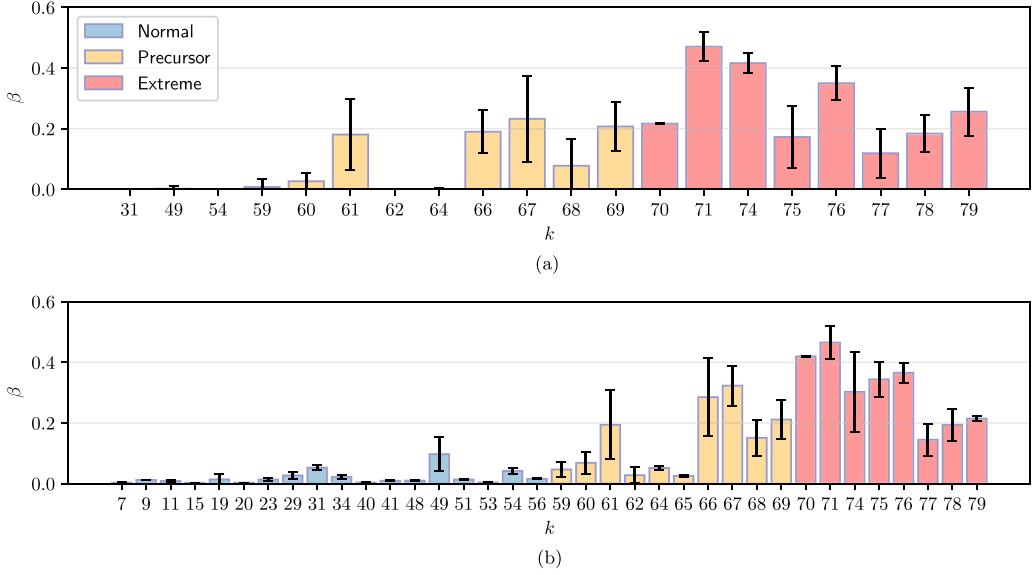


FIG. 10. Mean control signal β associated with each cluster along with the standard deviation for (a) optimized b_k and (b) modified b_k , where values associated with normal clusters are set to zero after optimization. Only clusters with nonzero states count and nonzero β are shown. Numbers are based on the extreme events from the set $\mathcal{P}$ after optimization.

conditional probabilities. The probability that a state associated with cluster k either transitions directly to a normal cluster or passes through one or more other precursor clusters before eventually transitioning to a normal one is denoted $\text{Pr}_{\text{final}}(\text{N} | k)$. The probabilities that a state associated with cluster k transitions directly to a normal or an extreme cluster are denoted $\text{Pr}_{\text{direct}}(\text{N} | k)$ and $\text{Pr}_{\text{direct}}(\text{E} | k)$, respectively. These conditional probabilities are defined as follows:

$$\text{Pr}_{\text{final}}(\text{N} | k) = \frac{C(k \rightarrow \text{N}) + C(k \rightarrow \dots \rightarrow \text{N})}{C(k)}, \quad (13a)$$

$$\text{Pr}_{\text{direct}}(\text{N} | k) = \frac{C(k \rightarrow \text{N})}{C(k \rightarrow \text{N}) + C(k \rightarrow \dots \rightarrow \text{N})}, \quad (13b)$$

$$\text{Pr}_{\text{direct}}(\text{E} | k) = \frac{C(k \rightarrow \text{E})}{C(k \rightarrow \text{E}) + C(k \rightarrow \dots \rightarrow \text{E})}, \quad (13c)$$

where $C(\cdot)$ is the number of states satisfying a given condition. For example, $C(k)$ is the number of states associated with cluster k , $C(k \rightarrow \dots \rightarrow \text{N})$ denotes the number of states that are associated with cluster k and transition to one (or more) other precursor cluster(s) before eventually transitioning to a normal one, and $C(k \rightarrow \text{N})$ is the number of states that directly transition from cluster k to a normal cluster.

For the clusters 59 and 62, the conditional probability $\text{Pr}_{\text{final}}(\text{N} | k)$ is close to unity (see Table III), indicating that these clusters are usually associated with false positives, which explains why $\beta \approx 0$. For cluster 64, this probability is not negligible (0.53), although significantly lower. All of these states, however, were found to transition directly to a normal cluster, as $\text{Pr}_{\text{direct}}(\text{N} | k) = 1$. Even with $\text{Pr}_{\text{final}}(\text{E} | k) = 1 - \text{Pr}_{\text{final}}(\text{N} | k) = 0.47$ being a non-negligible probability of a state eventually transitioning to an extreme cluster, the states were consistently found to transition first to other precursor clusters (clusters 60 and 61), as $\text{Pr}_{\text{direct}}(\text{E} | k) = 0$. These intermediate precursor clusters exhibit nonzero control signals, thereby compensating for the inactive control ($\beta \approx 0$) of cluster 64.

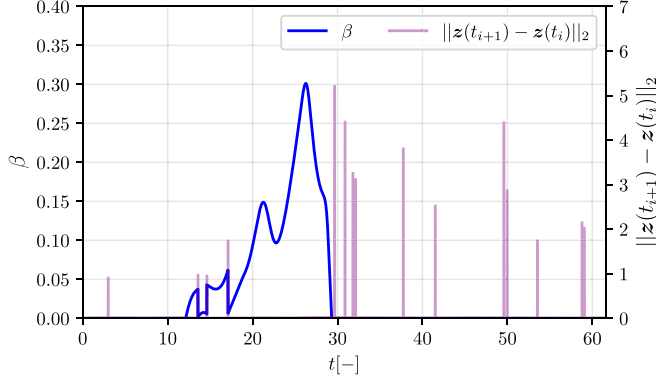


FIG. 11. Control signal β evolution over time for suppressing an extreme event, demonstrating jumps that correlate with jumps in the distances between two time-consecutive latent space variables.

Further, since β for all normal clusters is zero [see Figs. 9 and 10(a)], when a state transitions from the proximity of these precursor clusters to a normal one, it enters a region of the latent space where β is predominantly zero, which can be interpreted as a false positive. In other words, their trajectories in the latent space reveal a low probability of leading to extreme events. Therefore, no control signal should be applied to avoid perpetual control without imminent bursts in the energy dissipation rate. It is important to emphasize that these findings are specific to the set of extreme events, $\mathcal{P}$, used for optimization.

Moreover, this observation regarding the β being almost always zero for normal clusters could falsely indicate that the presence of these actuation values can be omitted (i.e., forced to be zero during runtime). However, due to the nonlinearity of the control law and the proximity of normal clusters to (some) precursor and extreme ones, enforcing this modification of b_k results in a nonzero control signal for many normal states, as shown in Fig. 10(b), and even increases the overall magnitude of β corresponding to both precursor and extreme clusters. Thus, this demonstrates the interactions of the actuation values that influence the resulting control distribution.

3. Enforcing a smooth control signal

Examining the control signal for a single extreme event reveals abrupt changes in β , which is one of the drawbacks of using SA-nCAEs. The core idea of the SA-nCAE is to transform an input state such that its latent-space representation achieves the highest possible reconstruction accuracy among all invariant transformations. This process introduces jumps in the latent space between two consecutive points in the state space, yielding jumps in the control signal. As seen in Fig. 11, the discontinuities observed in the control parameter β arise precisely at instances where the Euclidean distance between consecutive latent states, $\|z(t_{i+1}) - z(t_i)\|_2$, exhibits abrupt changes. Otherwise, β evolves as a smooth control signal over time. This relationship provides a clear explanation for the observed jumps in the β control signal. Although these discontinuities did not induce any short- or long-term detrimental effects on the numerical stability of the solution, they may pose practical challenges for real-world implementations, where smooth control dynamics are often required due to the finite response time of actuators.

To prevent such abrupt variations in the control signal, we apply the smoothing function defined in Eq. (7), parameterized by the response time t_R . The optimization was initially performed assuming $\Delta t_\beta = 4\Delta t$ and $t_R = 1\Delta t$, corresponding to an instantaneous control response. To assess the influence of actuator latency, we evaluate the performance for a range of controller latencies ($\Delta t_\beta = \{4\Delta t, 16\Delta t\}$) and response times ($t_R = \{1\Delta t, 12\Delta t, 28\Delta t\}$), each representing a controller-actuator setup with distinct latency characteristics. As shown in Fig. 12, all cases exhibit suppression of extreme events. Even for large response times (e.g., $t_R = 28$), the controller remains effective

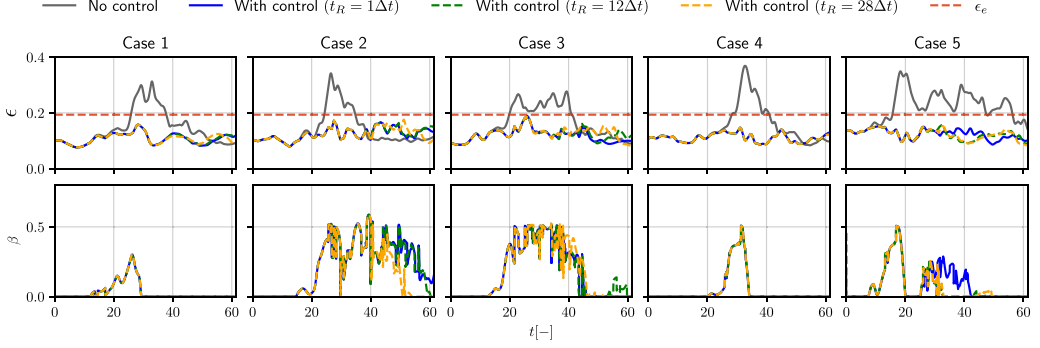


FIG. 12. Evaluating the effect of different response times $t_R = \{1\Delta t, 12\Delta t, 28\Delta t\}$ on the control effectiveness for various extreme events. All cases are evaluated using $\Delta t_\beta = 16\Delta t$.

in suppressing extreme events, which is also true for long time horizons (10 000 time units), as shown in Fig. 13. These results indicate that Δt_β and t_R may be chosen to match the intrinsic latency of a given controller-actuator setup, enabling robust performance across a range of practical implementations.

D. Scaling to higher Reynolds numbers

This section investigates the generalizability and scalability of our framework. We first evaluate its generalizability by applying the framework directly to higher Reynolds numbers without retraining. We then examine its scalability by retraining the framework for a higher Reynolds number case where the model trained at $Re = 40$ fails to generalize. For $Re > 40$, the data generation follows the same procedure discussed in Sec. II A, but uses 256 collocation points in each spatial direction (similar to [28]) and constant $\Delta t = 0.001$.

1. Zero-shot control

The framework is here applied to higher Re cases to demonstrate its ability to generalize to more chaotic flow regimes in Kolmogorov flow. We reuse the SA-nCAE model, clustering, and b_k parameters obtained for the $Re = 40$ case and apply them, without retraining or fine-tuning (also known as *zero-shot* control), for a range of more chaotic flow regimes.

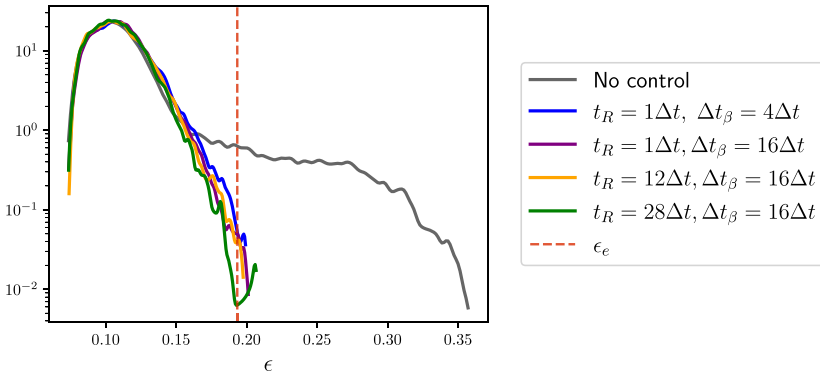


FIG. 13. The probability density function for the energy dissipation rate obtained for trajectories of 10 000 time units and different combinations of controller (Δt_β) and actuator (t_R) latency values.

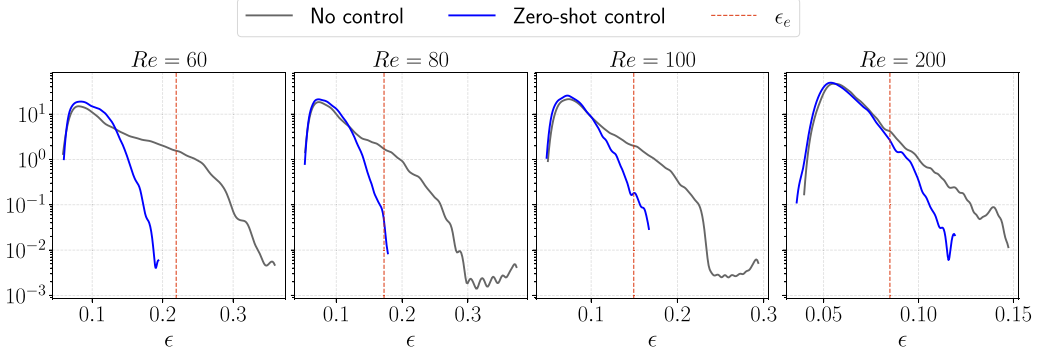


FIG. 14. Probability density function of energy dissipation rate for zero-shot control over a range of Reynolds numbers. Note the nearly flat segment at the upper end of the tail of the uncontrolled case for $Re = 80$ and 100 is caused by a single extreme event whose maximum value is notably larger than others.

As seen in Fig. 14 and Table IV, zero-shot control reduced the occurrence of extreme events for $Re = \{60, 80, 100\}$ by 100%, 99.97%, and 99.21%, respectively, albeit with a slightly longer control activation duration (i.e., smaller $\beta = 0$ percentage). These results demonstrate that the framework generalizes effectively across this range of Reynolds numbers. Nonetheless, retraining is necessary only for the substantially more chaotic $Re = 200$ case, which exhibits $\approx 2\%$ of extreme events after control. To recover optimal performance in terms of the percentage reduction in extreme events, we focus on retraining for the $Re = 200$ case in the following subsection.

2. Retraining for a higher Reynolds number

For $Re = 200$, we consider an SA-nCAE with $N_d = 8$ and train it for the same number of iterations with the same dataset size, resulting in a relatively higher reconstruction loss of 0.10497. At higher Re , the system exhibits more chaotic behavior and finer flow scales, leading to a more intricate structure of the turbulent attractor in state space. Consequently, a simple CAE architecture with a very low-dimensional latent space is insufficient to capture these high-dimensional structure of the flow manifold [50]. Nevertheless, for a direct comparison with the $Re = 40$ case and to assess scalability at higher Re with minimal adjustments, this configuration is adopted for the downstream tasks.

A summary of the corresponding clustering accuracy for state identification is presented in Table V. In comparison to the results shown in Table II for the lower Re case, we observe a slight reduction in performance, particularly for TP_P and FP_E for the distance-based method. This is attributed to the increased difficulty of distinguishing between the different state types at higher Re , where $\sigma(\epsilon)$ is smaller. Moreover, notice that the LD is approximately one-third of the lower Re case. Due to the different dynamics of the current flow regime with lower Lyapunov time, extreme events

TABLE IV. Zero-shot control performance for different Reynolds numbers. EE denotes extreme events.

	Re			
	60	80	100	200
EE (%) controlled	0.00	0.01	0.18	1.98
EE (%) no control	6.14	5.91	5.92	4.51
EE reduction (%)	100.00	99.97	99.21	89.02
$\beta = 0$ (%)	71.20	70.40	72.80	81.50

TABLE V. Clustering statistics obtained from top-performing SA-nCAE runs under varying parameters and identification methods (Cluster- and distance-based) for the $Re = 200$ case. Same details as in Table II.

Case	M	$\Delta t'$	Time	N-P-E	Cluster-based		Distance-based				
					TP _p	FP _p	TP _p	FP _p	LD	TP _E	FP _E
S1'	40	0.2	1000	46-27-27	100	0	72.02	76.70	0.851	83.61	58.35
S2'	50	0.2	2000	61-47-47	100	0	77.23	74.64	0.9234	88.86	60.02
S3'	30	0.4	2000	36-46-46	100	0	87.37	67.92	1.175	89.68	62.86

are found to occur 2.35 times more frequently and have an average duration that is 2.47 times shorter over a trajectory of 10 000 time units. LD is then observed to be proportionately lower. Case S3' (Table V) is selected as it yields the highest TP_p and lowest FP_p among all considered cases using the distance-based method. It can be seen for this case that the number of precursor and extreme event clusters is significantly higher than normal clusters as opposed to case S3 (cf. Table II). This difference is associated with the increased frequency of extreme events in the higher Re regime.

For optimization of the control law, we adopt the same settings as outlined previously, unless otherwise stated. The temporal horizon is chosen to be $T = 10$, since the LD and average duration of an extreme event are found to be shorter than the lower Re case. To account for the change in the statistics of the energy dissipation rate between $Re = 40$ and 200, we set the loss scaling coefficients to $\gamma_\epsilon = 1$ and $\gamma_\beta = 3$ [Eq. (9)]. We also use $n_{\sigma, \text{crit}} = 1$ such that $\Pr(\epsilon \geq \epsilon_{\text{crit}} |_{Re=40} | Re = 40) \approx \Pr(\epsilon \geq \epsilon_{\text{crit}} |_{Re=200} | Re = 200)$. In other words, $n_{\sigma, \text{crit}}$, which is used to define the threshold $\epsilon_{\text{crit}} = \bar{\epsilon} + n_{\sigma, \text{crit}}\sigma(\epsilon)$ for highly unstable precursors, is adjusted to almost match the probability of the previous case. This ensures that control is activated only when necessary to mitigate true extreme events by acting on the highly unstable precursors, which now have shorter LD as well for higher Re.

After 77 iterations using a learning rate of 0.03, the optimization converged with a validation step performed every two iterations. The control law with the optimized actuation values was evaluated using the threshold $\epsilon_e = 0.085$ and achieved a reduction in the probability of extreme events by 96.9% (from 4.51% in the uncontrolled case to 0.138%), as shown in Fig. 15(b). Note that the occurrence of extreme events dropped from $\approx 2\%$ in the zero-shot control case (see Table IV) to 0.138% after retraining, highlighting the benefit of adapting the controller to this more complex flow

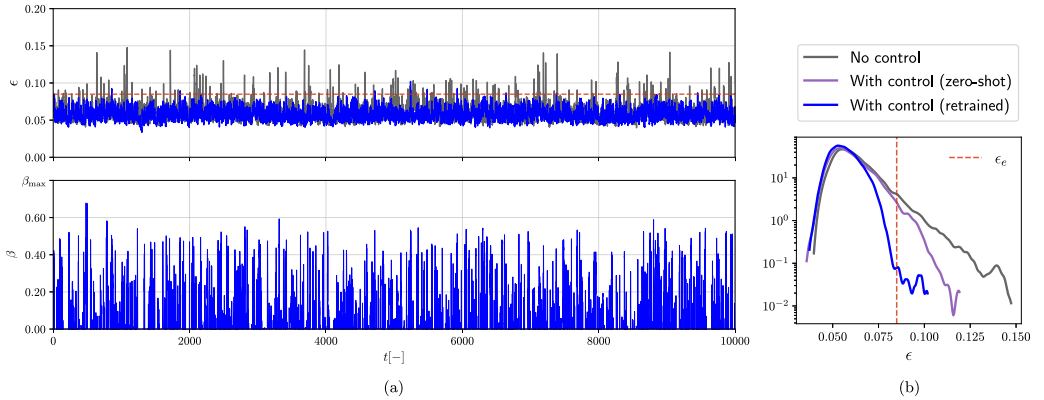


FIG. 15. Evaluation results of control obtained using SA-nCAE encoding for $Re = 200$. The top panel in (a) shows the time series for energy dissipation with and without control overlaid. The bottom panel shows the corresponding control signal β for the controlled trajectory. The probability density function for the energy dissipation rate is shown before and after control for the zero-shot and retrained cases in (b).

regime. Further, the control law is found to be active for about 14.5% of the time, which represents efficient control that is primarily active prior to extreme events. It is also active for a shorter duration than in the zero-shot case (18.5%), while more effectively suppressing extreme events.

Finally, the time series of the control signal and the energy dissipation rate before and after control is shown in Fig. 15(a). In comparison with the lower Re case [in Fig. 8(a)], it can be seen that the frequency of the observed β peaks (bottom panels) is higher for the higher Re case, although the control is active for a relatively smaller fraction of time. This increase is consistent with the higher frequency of extreme events observed in the uncontrolled trajectory (top panels).

V. CONCLUSION

We have presented a multistage, predominantly data-driven framework for the suppression of extreme events in turbulent flows. The proposed methodology relies on control signals derived in the latent space of a reduced-order model, where a clustering algorithm enables the identification of distinct flow states. By associating an optimal actuation value with each cluster, the resulting control law efficiently mitigates extreme events through timely interventions during their precursor phases.

The efficacy of the framework was demonstrated on the turbulent Kolmogorov flow, where the application of the learned control law led to a reduction of nearly all extreme events by up to 99.4%, and a negligible frequency and intensity of the few events that persisted beyond the extreme events threshold. Furthermore, the use of a symmetry-aware autoencoder proved instrumental in constructing a “structured” latent space that facilitates state identification and yields control that is active only when necessary. The framework is applied to a higher Re flow regime and achieves comparable performance in the reduction of the occurrence of extreme events by up to 96.9%, demonstrating scalability to more chaotic flow regimes.

Finally, we introduced a practical extension of the framework that accounts for controller and actuator latencies, demonstrating the robustness of the proposed control strategy for real world scenarios related to controller-actuator setups. Collectively, these results demonstrate the promise of data-driven, reduced-order modeling approaches for the real-time suppression of extreme events in turbulent flows.

For future work, the symmetry-aware autoencoder that utilizes prior knowledge of the considered system can be replaced with other system-agnostic approaches, e.g., [51], fostering a fully data-driven approach applicable to real life settings. Additionally, evaluating the performance of the proposed framework for higher-dimensional flows remains an important direction for future investigation.

ACKNOWLEDGMENT

Y.S. and N.A.K.D. acknowledge financial support from the European Research Council Starting Grant CONTEXT, Grant No. 1011612942. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

DATA AVAILABILITY

The data that support the findings of this study (at $Re = 40$) are openly available at [52].

APPENDIX A: LATENT SPACE STRUCTURE FOR MULTIPLE CAEs

This appendix presents Figs. 16 and 17, specifically corresponding to the latent space structure visualization for ICA E and $nCAE$. These supplementary figures provide additional visual details for comparison with Fig. 7.

APPENDIX B: CLUSTERING SCALING FOR MULTIPLE LATENT SIZES

We train multiple SA-nCAEs with latent sizes $N_d = \{4, 6, 8, 10\}$ using the same settings as Case S3 (see Table II). The clustering accuracy for precursor identification using both cluster- and distance-based approaches as well as runtime is presented in Fig. 18. $M = 20$ is used for $N_d = 4$ as it provides the highest accuracy for the presented metrics and for $N_d = 10$ due to algorithmic limitations for relatively higher dimensional spaces. $M = 30$ is used for the remaining cases. These results indicate that accuracy substantially improves as N_d increases, while runtime also increases correspondingly, until a plateau is reached at $N_d = 8$ and 10. $N_d = 8$ represents a practical balance between clustering accuracy and computational cost for the current flow problem, providing near-optimal performance without exceeding memory or runtime constraints.

Note that the minor drop in performance for $N_d = 10$ is due to the inability to explore higher tessellations with $M > 20$ where an optimum performance is expected to exist. This is a direct consequence of the current implementation of the clustering algorithm, which relies on a sparse matrix representation of the dense transition probability matrix, $\mathcal{P}$. This representation imposes a practical

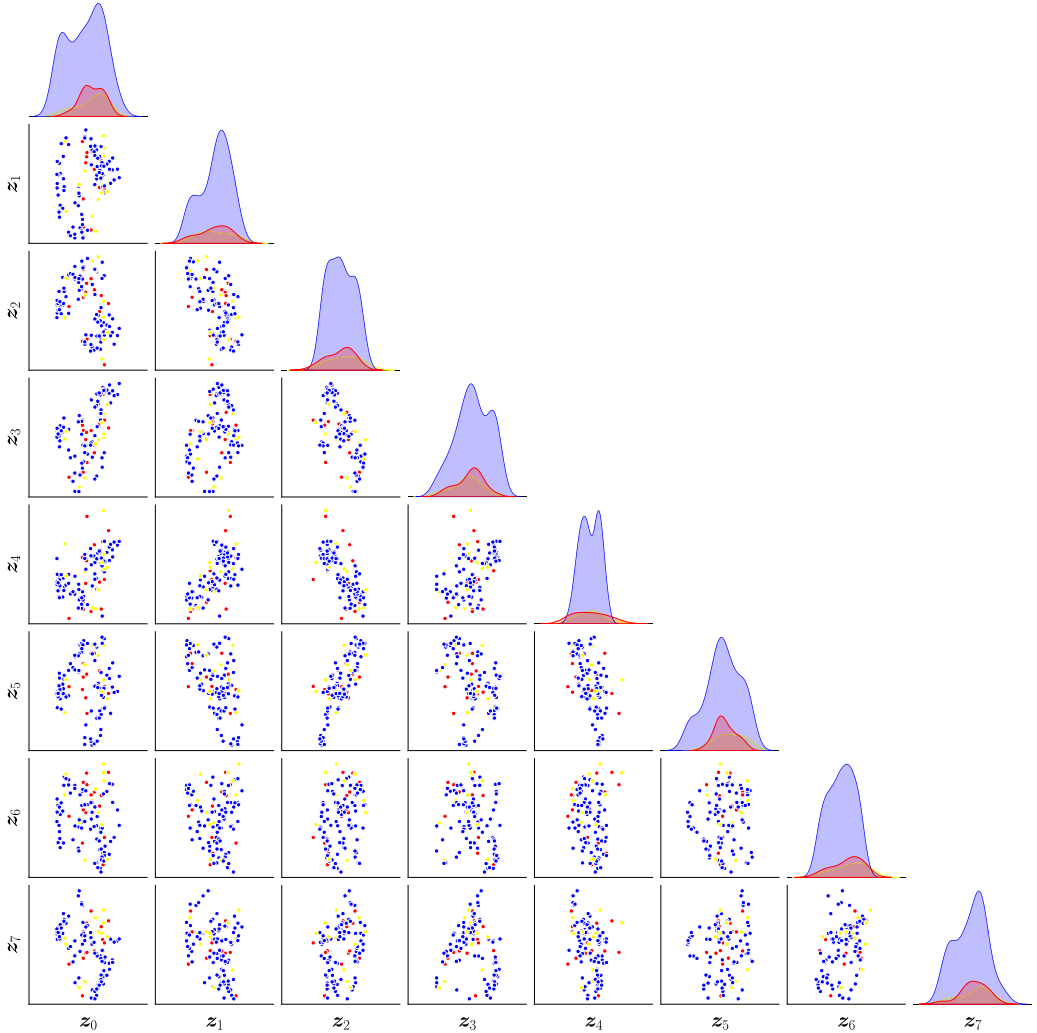


FIG. 16. Cluster centroids visualized in the latent space of the linear CAE (Case L2). See Fig. 7 for details.

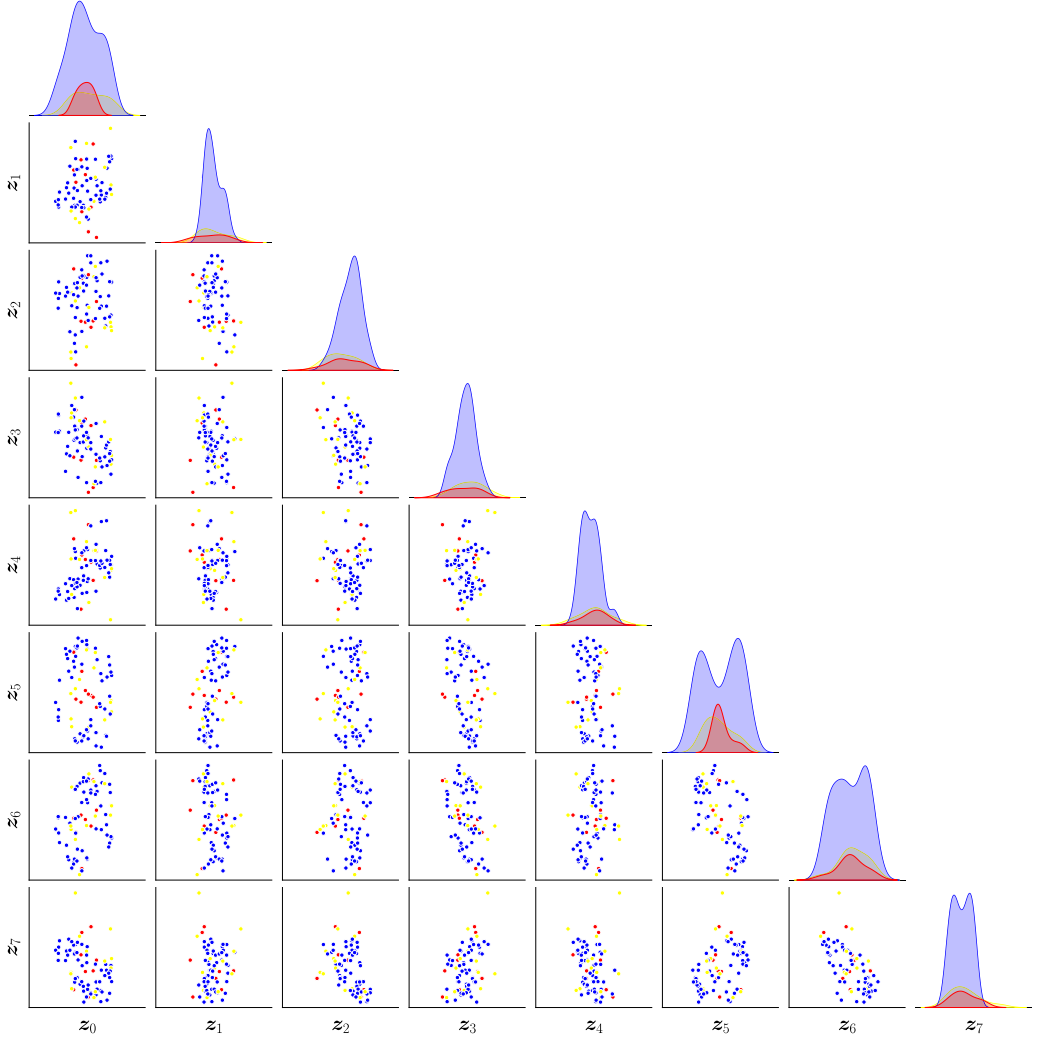


FIG. 17. Cluster centroids visualized in the latent space of the nonlinear CAE (Case V3). See Fig. 7 for details.

upper bound on the indexable nonzero entries, determined by the maximum value of a 64-bit integer (`int64`) used to store matrix indices in the C language. Therefore, while the computational cost does not scale exponentially with increasing N_d , although the total number of hypercubes is M^{N_d+1} , this growth leads to a practical upper limit on N_d within the current implementation. This is an algorithmic limitation specific to the current implementation. Consequently, this provides a cap to the maximum achievable performance from the clustering stage.

APPENDIX C: DIMENSIONALITY REDUCTION OF A FULL RESOLUTION INPUT

To investigate the effect of the downsampling performed prior to an AE on the performance of the downstream clustering task, we run a case where the AE is being applied to the velocity field with full resolution without downsampling (i.e., $N_x \times N_y = 128 \times 128$). The AE used for the higher

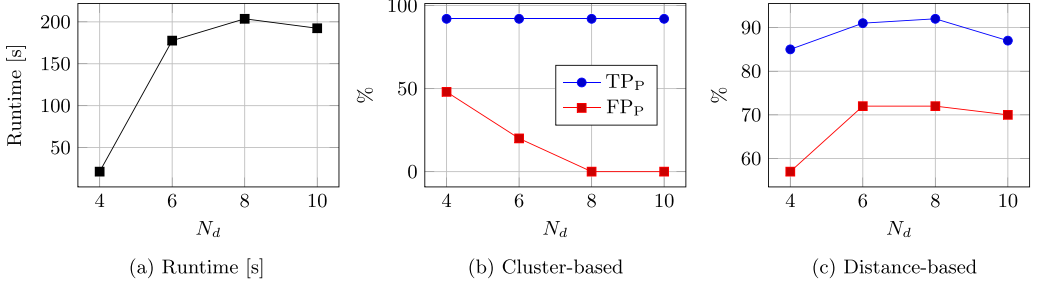


FIG. 18. Clustering performance metrics for multiple N_d values. The time taken (in seconds) to obtain the final set of clusters on the same workstation is shown in (a). The true and false positive rates for precursors (TP_P and FP_P) of the obtained clusterings are reported when using the cluster-based (b) and distance-based (c) methods. Runtime values are obtained by averaging five runs per case, with standard deviation below 1 for all cases (not shown).

resolution input has a slightly different architecture (see Table VI) to enable better feature extraction and reconstructions free from computational artifacts.

We compare the accuracy using SA-nCAE with $N_d = 8$ against the case with downsampling, i.e., case S3 from Table II with $N_x = N_y = 32$. The reconstruction loss, $\mathbb{E}_{\mathbf{u} \sim \mathcal{U}}[\|\hat{\mathbf{u}} - \mathbf{u}\|_2^2]$, for both cases is found to be almost the same (0.02088 and 0.02158 for the low and high resolution cases, respectively) and the clustering accuracy is quite comparable, as shown in Table VII. Nonetheless, it is worth mentioning that the AE used for the higher resolution input has $5\times$ more trainable parameters, requires higher memory capacity, and takes about $13.2\times$ longer to train on the same GPU. We, therefore, conclude that downsampling is a critical step to substantially reduce the computational cost of running the AE without adversely affecting the performance of downstream stages.

APPENDIX D: COMPUTATIONAL COST OF OUR FRAMEWORK

To facilitate future comparisons of the computational cost of our proposed framework with other approaches, we report the runtime for both training and evaluation of the Re = 40 case. All simulations/runs were conducted on a machine equipped with an NVIDIA RTX A4500 GPU (20 GB), an Intel Xeon processor (20 cores), and 192 GB of RAM. The training time for the nCAE and SA-nCAE models was approximately 3.5 minutes and 65 minutes, respectively. This

TABLE VI. Encoder and decoder architectures of the autoencoder applied to a full resolution velocity field at 128×128 resolution. The hyperbolic tangent activation function is used between layers, except for the final layer.

Encoder			Decoder		
Layer	Input size	Output size	Layer	Input size	Output size
Conv 3×3	$2 \times 128 \times 128$	$32 \times 64 \times 64$	Linear	N_d	2048
Conv 3×3	$32 \times 64 \times 64$	$64 \times 32 \times 32$	Unflatten	2048	$128 \times 4 \times 4$
Conv 3×3	$64 \times 32 \times 32$	$128 \times 16 \times 16$	Upsample + Conv 3×3	$128 \times 4 \times 4$	$128 \times 8 \times 8$
Conv 3×3	$128 \times 16 \times 16$	$128 \times 8 \times 8$	Upsample + Conv 3×3	$128 \times 8 \times 8$	$128 \times 16 \times 16$
Conv 3×3	$128 \times 8 \times 8$	$128 \times 4 \times 4$	Upsample + Conv 3×3	$128 \times 16 \times 16$	$128 \times 32 \times 32$
Flatten	$128 \times 4 \times 4$	2048	Upsample + Conv 3×3	$128 \times 32 \times 32$	$64 \times 64 \times 64$
Linear	2048	N_d	Upsample + Conv 3×3	$64 \times 64 \times 64$	$32 \times 128 \times 128$
			Conv 3×3	$32 \times 128 \times 128$	$2 \times 128 \times 128$

TABLE VII. Clustering statistics of SA-nCAE obtained from top-performing runs with different input resolution ($N_x \times N_y$) and identification methods (Cluster- and distance-based). Stride ($\Delta t'$) and sequence length used to obtain the clustering is 0.4 and 2000 time units, respectively. The first case corresponds to case S3 in Table II. Other details are similar to Table II.

$N_x \times N_y$	M	N-P-E	Cluster-based		Distance-based				
			TP _P	FP _P	TP _P	FP _P	LD	TP _E	FP _E
32×32	30	59-11-10	92.31	0	91.98	71.51	3.979	80.97	17.35
128×128	40	60-12-10	92.31	0	89.73	74.06	2.423	88.82	31.20

difference arises from the fact that a SA-nCAE, based on a Siamese network, has 16 branches of a single nCAE. The clustering stage incurred a comparatively negligible computational cost, requiring approximately 200 seconds for cases V3 and S3 (Table II). The iterative optimization procedure of b_k takes approximately 59 hours and 13 minutes for 1000 iterations of the nCAE-based encoding, corresponding to an average of 3.55 minutes per iteration. Similarly, the optimization using SA-nCAE over 140 iterations required approximately 9 hours and 50 minutes, yielding an average of about 4.25 minutes per iteration. The average cost is indeed lower during the initial iterations when $|\tilde{\mathcal{P}}| < |\mathcal{P}|$. During evaluation for the same trajectory length of 10 000 time units, nCAE and SA-nCAE took about 290 and 323 minutes, respectively, whereas an uncontrolled trajectory takes approximately 259 minutes.

A direct quantitative comparison of the computational cost with other methods is difficult since few works focus on the control of extreme events in turbulence and such metrics are rarely reported. For instance, Farazmand and Sapsis [9,28] employ a control strategy requiring a precursor from a PDE-constrained optimization problem that maximizes the growth of an observable (i.e., the energy dissipation rate). They describe their method as having a “relatively low computational cost” compared to classical adjoint-based techniques, which are generally known to be computationally demanding, requiring complex implementation and memory-intensive backward-in-time computations [9]. However, no quantitative analysis is provided. Similarly, in Racca and Magri [23], a surrogate model based on Echo State Networks (ESNs) is first trained to predict the forward-in-time dynamics of a flow and is then employed at each time step to predict the current flow state over a finite horizon to predict imminent extreme events. The computational cost scales with the length of this temporal horizon. However, the overall cost of the method is not discussed in their work and it is only validated for an ODE system (the “Moehlis, Faisst, and Eckhardt” model [53]) making any comparison with our present work difficult. In contrast, our approach requires only a single forward pass through the AE to locate the fluid state in the latent space and compute the control signal directly using Eq. (6).

-
- [1] S. B. Pope, *Turbulent Flows* (Cambridge University Press, Cambridge, 2000).
 - [2] E. N. Lorenz, Deterministic nonperiodic flow, *J. Atmos. Sci.* **20**, 130 (1963).
 - [3] S. H. Strogatz, *Nonlinear Dynamics and Chaos* (Chapman and Hall/CRC, Boca Raton, FL, 2024).
 - [4] K. Dysthe, H. E. Krogstad, and P. Müller, Oceanic rogue waves, *Annu. Rev. Fluid Mech.* **40**, 287 (2008).
 - [5] C. Kharif, E. Pelinovsky, and A. Slunyaev, *Rogue Waves in the Ocean* (Springer, Berlin, 2009).
 - [6] D. R. Easterling, J. L. Evans, P. Y. Groisman, T. R. Karl, K. E. Kunkel, and P. Ambenje, Observed variability and trends in extreme climate events: A brief review, *Bull. Am. Meteorol. Soc.* **81**, 417 (2000).
 - [7] R. B. Alley, J. Marotzke, W. D. Nordhaus, J. T. Overpeck, D. M. Peteet, R. A. Pielke, R. T. Pierrehumbert, P. B. Rhines, T. F. Stocker, L. D. Talley, and J. M. Wallace, Abrupt climate change, *Science* **299**, 2005 (2003).

- [8] V. Dakos, M. Scheffer, E. H. van Nes, V. Brovkin, V. Petoukhov, and H. Held, Slowing down as an early warning signal for abrupt climate change, *Proc. Natl. Acad. Sci. USA* **105**, 14308 (2008).
- [9] M. Farazmand and T. P. Sapsis, A variational approach to probing extreme events in turbulent dynamical systems, *Sci. Adv.* **3**, e1701533 (2017).
- [10] P. J. Blonigan, M. Farazmand, and T. P. Sapsis, Are extreme dissipation events predictable in turbulent fluid flows? *Phys. Rev. Fluids* **4**, 044606 (2019).
- [11] T. Lieuwen, V. McDonell, E. Petersen, and D. Santavicca, Fuel flexibility influences on premixed combustor blowout, flashback, autoignition, and stability, *J. Eng. Gas Turbines Power* **130**, 011506 (2008).
- [12] M. Hassanaly and V. Raman, Classification and computation of extreme events in turbulent combustion, *Prog. Energy Combust. Sci.* **87**, 100955 (2021).
- [13] M. Floris, T. Shiva Sai, D. Nayak, I. Langella, K. Aditya, and N. A. K. Doan, Data-driven identification of precursors of flashback in a lean hydrogen reheat combustor, *Proc. Combust. Inst.* **40**, 105524 (2024).
- [14] J. Page, M. P. Brenner, and R. R. Kerswell, Revealing the state space of turbulence using machine learning, *Phys. Rev. Fluids* **6**, 034402 (2021).
- [15] R. Maulik, B. Lusch, and P. Balaprakash, Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders, *Phys. Fluids* **33**, 037106 (2021).
- [16] N. A. K. Doan, W. Polifke, and L. Magri, Auto-encoded reservoir computing for turbulence learning, in *International Conference on Computational Science* (Springer, Cham, 2021), pp. 344–351.
- [17] A. J. Linot and M. D. Graham, Data-driven reduced-order modeling of spatiotemporal chaos with neural ordinary differential equations, *Chaos* **32**, 073110 (2022).
- [18] C. E. P. De Jesús, Carlos E. A. J. Linot, and M. D. Graham, Building symmetries into data-driven manifold dynamics models for complex flows, *Phys. Rev. Fluids* **10**, 064401 (2025).
- [19] T. Kurihana, I. Mastilovic, L. Wang, A. Meray, S. Praveen, Z. Xu, M. Memarzadeh, A. Lavin, and H. Wainwright, Identifying climate patterns using clustering autoencoder techniques, *AIES* **3** (2024).
- [20] F. R. Spuler, M. Kretschmer, M. A. Balmaseda, Y. Kovalchuk, and T. G. Shepherd, *Learning Predictable and Informative Dynamical Drivers of Extreme Precipitation using Variational Autoencoders* (Copernicus Publications, Göttingen, Germany, 2025).
- [21] K. Aoki, H. Nakano, N. Hirose, N. Usui, K. Sakamoto, T. Toyoda, S. Urakawa, and Y. Kawakami, Application of variational autoencoder-based clustering for geophysical fluid circulations with small sample size, *J. Geophys. Res.: Mach. Learn. Comput.* **3**, e2025JH001051 (2026).
- [22] N. A. K. Doan, W. Polifke, and L. Magri, Short- and long-term predictions of chaotic flows and extreme events: A physics-constrained reservoir computing approach, *Proc. R. Soc. A* **477**, 20210135 (2021).
- [23] A. Racca and L. Magri, Data-driven prediction and control of extreme events in a chaotic flow, *Phys. Rev. Fluids* **7**, 104402 (2022).
- [24] U. Golska and N. A. K. Doan, Clustering-based identification of precursors of extreme events in chaotic systems, in *Computational Science–ICCS 2023*, Lecture Notes in Computer Science Vol. 10476, edited by J. Mikyška, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot (Springer Nature, Cham, 2023), pp. 313–327.
- [25] V. Pyragas and K. Pyragas, Using reservoir computer to predict and prevent extreme events, *Phys. Lett. A* **384**, 126591 (2020).
- [26] W. E. Sherwood, Fitzhugh–Nagumo model, in *Encyclopedia of Computational Neuroscience*, edited by D. Jaeger and R. Jung (Springer New York, New York, 2013), pp. 1–11.
- [27] A. Racca and L. Magri, Control-aware echo state networks (Ca-ESN) for the suppression of extreme events, in Machine Learning and the Physical Sciences Workshop, [arXiv:2308.03095](https://arxiv.org/abs/2308.03095).
- [28] M. Farazmand and T. P. Sapsis, Closed-loop adaptive control of extreme events in a turbulent flow, *Phys. Rev. E* **100**, 033110 (2019).
- [29] A. M. Obukhov, Kolmogorov flow and laboratory simulation of it, *Russ. Math. Surv.* **38**, 113 (1983).
- [30] N. Platt, L. Sirovich, and N. Fitzmaurice, An investigation of chaotic Kolmogorov flows, *Phys. Fluids* **3**, 681 (1991).
- [31] D. Kelshaw, KolSol: Pseudospectral Kolmogorov flow solver, Github, 2022, <https://github.com/MagriLab/KolSol>.

- [32] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer, Berlin, 1988).
- [33] J. L. Lumley, *The Structure of Inhomogeneous Turbulent Flows, Atmospheric Turbulence and Radio Wave Propagation*, edited by A. M. Yaglom and V. I. Tatarsky (Nauka, Moscow, 1967), pp. 166–178.
- [34] G. Berkooz, P. Holmes, and J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annu. Rev. Fluid Mech.* **25**, 539 (1993).
- [35] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *J. Fluid Mech.* **656**, 5 (2010).
- [36] T. Murata, K. Fukami, and K. Fukagata, Nonlinear mode decomposition with convolutional neural networks for fluid dynamics, *J. Fluid Mech.* **882**, A13 (2020).
- [37] E. Plaut, From principal subspaces to principal components with linear autoencoders, [arXiv:1804.10253v3](https://arxiv.org/abs/1804.10253v3).
- [38] S. Kneer, T. Sayadi, D. Sipp, P. Schmid, and G. Rigas, Symmetry-aware autoencoders: s-PCA and s-nlPCA, [arXiv:2111.02893](https://arxiv.org/abs/2111.02893).
- [39] K. Zeng and M. D. Graham, Symmetry reduction for deep reinforcement learning active control of chaotic spatiotemporal dynamics, *Phys. Rev. E* **104**, 014210 (2021).
- [40] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, Spatial transformer networks, *Advances in Neural Information Processing Systems 28* (NIPS, 2015), pp. 2017–2025.
- [41] P. J. Schmid, A. García-Gutiérrez, and J. Jiménez, Description and detection of burst events in turbulent flows, *J. Phys.: Conf. Ser.* **1001**, 012015 (2018).
- [42] P. J. Schmid, O. T. Schmidt, A. Towne, and M. J. P. Hack, Analysis and prediction of rare events in turbulent flows, *Advances in Neural Information Processing Systems 28 (NIPS 2015)* (Curran Associates, Inc., Red Hook, NY, 2018), pp. 2017–2025.
- [43] A. G. Nair, C.-A. Yeh, E. Kaiser, B. R. Noack, S. L. Brunton, and K. Taira, Cluster-based feedback control of turbulent post-stall separated flows, *J. Fluid Mech.* **875**, 345 (2019).
- [44] I. Loshchilov and F. Hutter, SGDR: Stochastic gradient descent with warm restarts, in *International Conference on Learning Representations (ICLR, 2017)*.
- [45] T. A. Howell, S. Le Cleac’h, S. Singh, P. Florence, Z. Manchester, and V. Sindhvani, Trajectory optimization with optimization-based dynamics, *IEEE Rob. Autom. Lett.* **7**, 6750 (2022).
- [46] B. Sukhija, N. Köhler, M. Zamora, S. Zimmermann, S. Curi, A. Krause, and S. Coros, Gradient-based trajectory optimization with learned dynamics, in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, London, UK, 2022), pp. 1011–1018.
- [47] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *3rd International Conference on Learning Representations (ICLR 2015)* (OpenReview.net, San Diego, CA, USA, 2014).
- [48] Y. Bengio, P. Simard, and P. Frasconi, Learning long-term dependencies with gradient descent is difficult, *IEEE Trans. Neural Netw.* **5**, 157 (1994).
- [49] R. Pascanu, T. Mikolov, and Y. Bengio, On the difficulty of training recurrent neural networks, in *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 28, edited by S. Dasgupta and D. McAllester (PMLR, Atlanta, 2013), pp. 1310–1318.
- [50] A. Cleary and J. Page, Characterizing the Reynolds number dependence of the chaotic attractor in two-dimensional turbulence with dimension-minimizing autoencoders, *Phys. Rev. E* **112**, 055105 (2025).
- [51] M. Rudolph, B. Wandt, and B. Rosenhahn, Structuring autoencoders, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops* (IEEE, Seoul, South Korea, 2019), pp. 615–623.
- [52] Y. Shehata, K. Schuurman, P. D. Estevez, and N. A. K. Doan, Kolmogorov flow at Reynolds number = 40, Zenodo, 2026, <https://doi.org/10.5281/zenodo.21853176>.
- [53] J. Moehlis, H. Faisst, and B. Eckhardt, A low-dimensional model for turbulent shear flows, *New J. Phys.* **6**, 56 (2004).