AutoEncoders latent space interpretability in the light of proper orthogonal decomposition: Machine learning of periodically forced fluid flows

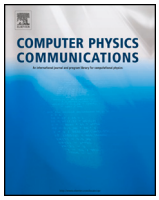
Rémi Bousquet, Caroline Nore, Didier Lucor

Computer Physics Communications 315 (2025) 109728

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



Computational Physics

AutoEncoders latent space interpretability in the light of proper orthogonal decomposition: Machine learning of periodically forced fluid flows

Rémi Bousquet *, Caroline Nore, Didier Lucor

Laboratoire Interdisciplinaire des Sciences du Numerique, CNRS, Campus universitaire Paris-Saclay, rue du Belvedere, Orsay, 91400, France

ARTICLE INFO

The review of this paper was arranged by Prof. Andrew Hazel

Keywords:
Autoencoders
Proper orthogonal decomposition
Interpretability
Dimensionality reduction
Fluid mechanics

ABSTRACT

This work explores the learning and interpretability challenges of Autoencoders (AEs) and Variational Autoencoders (VAEs) when applied to the reconstruction of dynamic velocity fields governed by the Navier-Stokes equations. Throughout model training, the emphasis is on understanding how flow features are encoded into the latent space and how this impacts the interpretability and usability of the models. Based on a parametric study of forced flows, i.e. flows around an oscillating cylinder, as well as a von Kármán swirling flow, we first investigate the trade-offs between reconstruction accuracy and regularization in VAEs. We confirm that increasing the regularization parameter degrades reconstruction quality, which underscores a significant limitation of the Gaussian prior from this point of vue. A comparative analysis reveals that standard AEs exhibit quite robust training behaviour, while VAEs show a sharper transition between non-learning and learning regimes, depending on the amount of regularization. By leveraging Proper Orthogonal Decomposition (POD) to identify characteristic flow structures and frequencies, we establish connections between latent space organisations and POD modes. To address the interpretability challenge, we then perform a symmetry analysis of latent spaces, stating equivariance relations between latent and physical variables. Despite reduced reconstruction precision, VAEs show greater fidelity in preserving these relationships. Building on this, we propose a clustering-inspired method to interpret latent representations, identifying characteristic states from temporal POD time coefficients to provide deeper insights into latent space structure and untangling. This work highlights pathways for autoencoder's analysis methodological advancements, emphasizing the critical need to align latent space representations with physical interpretation for broader applicability in fluid dynamics.

1. Introduction

In recent years, machine learning (ML) and, in particular, unsupervised techniques such as autoencoders (AE) have attracted growing interest in the field of computational physics and fluid mechanics, in particular for low-dimensional representations purposes. Numerous studies aimed at demonstrating their performance in various academic configurations, such as the channel flow [1], the flow around a cylinder [2], the pinball flow [3], Kolmogorov flow [4], or wall-bounded flows [5]. These efforts highlight the potential of ML in extracting low-dimensional *latent* representations of intricate fluid dynamic cases, but also underline the challenge of establishing consistent and interpretable frameworks for analysis, as well as for comparison between the different approaches. While these works show some great potential, their diversity in terms of datasets, applied models, and evaluation metrics makes it difficult to rigorously compare the performances of the models, and the analysis details. Furthermore, there is a growing need of robustness, which necessitates a reliable interpretation of the models to move beyond dependence on opaque "black-box" methods. This is critically important for improving disciplines such as reduced order modelling in AE's latent spaces, design optimization, or parameter inference directly from flow fields.

Linear methods such as proper orthogonal decomposition (POD), although interpretable and widely used, prove limited for complex phenomena such as turbulence or in context of fluid-structure interactions. And with good reason, the POD modes need to be skilfully grouped to understand the underlying flow structures. Indeed, the reconstruction of small-scale flow features often requires a very large number of modes, making this approach ineffective in many cases. AEs, on the other hand, offer a promising alternative for small-scale reconstruction, particularly in turbulent flows [4]. However, they present some major challenge: the lack of a generic method for analysing and interpreting the results,

* Corresponding author.
E-mail address: remi.bousquet@universite-paris-saclay.fr (R. Bousquet).

https://doi.org/10.1016/j.cpc.2025.109728
Received 28 January 2025; Received in revised form 16 June 2025; Accepted 19 June 2025
Available online 1 July 2025

particularly when it comes to interpreting the low-dimensional latent variables emerging from the models training. Advances have been made, in the field of reduced order modelling of fluid dynamics, by enforcing a known dynamical structure in the model, to learn the effective dynamics with dictionary-based learning [6,7]. Another approach is to hybridize AEs with more classical techniques such as POD Galerkin and the Mori-Zwanzig formalism, or with the Koopman operator formalism [3,8,9]. These approaches offer a relevant interpretation of the results from a dynamical point of view. They express dynamical relations between the latent variables, that can be linked *a posteriori* with known canonical systems. The current challenge lies in understanding and interpreting the latent space of AEs, expliciting the link between latent variables and flow structures. This need for interpretability is essential to make these models robust and reliable, so that they can be applied with guaranties to more complex, or real-world problems. Indeed, nonlinear dimension reductions are often very effective to compress the high-dimensional information, but latent variables have the tendency to be poorly organised in complex structures. This is particularly true for time-dependent data for which latent variables are compact representations of the temporal dynamics. It seems therefore important to disentangle these latent variables to identify independent physical mechanisms and parameters from data here obtained from PDEs governing our physical problem.

To address these challenges, several interpretation strategies have been proposed, first at an architectural level, for example with Mode Decomposing AutoEncoders (MD-AE) [2], Hierarchical AutoEncoders (HAE) [10,11], or Variational AutoEncoders (VAE) [12]. These efforts seek to explore the latent space by imposing various strategies on the decoder, in order to navigate this space and interpret what is encoded to the different latent directions, to better identify the information hidden in the data, and give it an interpretable meaning. In order to be efficient, these approaches need to properly disentangle the latent space, which, simply put, means that one characteristic of the flow should be controlled by one latent space dimension. Another way to say it, as in [13] is that: “the most robust approach to feature learning is to disentangle as many factors as possible, discarding as little information about the data as is practical.” The hope is that each latent component will encode a different characteristic of the flow, with the underlying idea that adding two characteristics in the latent space and using the decoder will result to adding these characteristics in the physical space. Another approach to the problem has been stated in [14], and more recently developed in [15]. It consists in analysing the latent space patterns by applying POD on the latent space manifold, using the local metric tensor of the latent space to define distances. While the first application of this idea [4] uses an Euclidian metric, it already brings interesting contributions, notably by decoding latent vector “modes”, performing POD in the latent space, then applying POD on the autoencoder reconstruction.

The approach proposed in the present work also leverages the link between POD and AEs, in the classic case where autoencoders optimize the mean square error (MSE) like POD. Certainly, both POD and AE optimize the MSE in two different fashions, but both approaches are governed by the hierarchy of the data, i.e. the principal energetic structures of the flows. By *structures*, we mean coherent flow organisations or patterns within the flow field that are related to observable physical phenomena, e.g. vortices or shear layers. For an in-depth discussion, see [16], in particular chapter two. These structures may be related to flow *features* which are quantitative descriptors often extracted by machine learning used to analyze, interpret, or reduce the dimensionality of flow data. For efficient and physically sound reduced order modelling, it makes sense to understand and promote the potential link between flow features and flow structures. For instance, the flow features are easier to relate to the flow structures when the feature extraction is guided by some physical principles or constraints, or when the model is linear as in POD. Nevertheless, interpretability issues may arise in POD as well, because fixed time localized structures of the flow are scattered over several modes of the decomposition, and then require numerous modes for a particular snapshot reconstruction. Not to mention the case where numerous modes have approximately the same energy and make up a single structure, where POD interpretation is not straightforward either. Thus, focusing on the notion of mode, as elements of an orthogonal basis of the flow, may not be the most appropriate option for interpreting AEs. Instead we want to find some characteristic states of the flow, that might not be orthogonal, but that describe meaningfully the flow with the least elements possible. Yet, using the POD to construct those characteristic states is a sound available option, as POD is well known and broadly used in the scientific community.

The main idea of the present approach is to show that AEs solve a data alignment task, i.e. they cluster alike data points next to each other in the latent space. Because fluid dynamics datasets are fields snapshots ordered in time, when the dataset has enough time resolution, one curvilinear alignment direction can be associated to time. Being the most energetic, the principal spatio-temporal components of the flow are encoded in the latent space. The way they manifest themselves in the latent space depends on the data specifics, the model architecture, or additional loss terms used. A starting point to build an interpretation of such models may be to define properly the word *interpretation*. We will distinguish between two kind of interpretations:

- When prior knowledge of the dynamical system is available, the interpretation task I_1 is to understand how the prior elements are expressed in the latent space.
- When prior knowledge is not available, the interpretation I_2 must apply a strategy to extract valuable information from the latent space.

We will try to show that I_1 can be done by confronting AE results to the POD, and will suggest a protocol for I_2, that we will not extensively apply.

This study will be carried out in several stages. Firstly, in section 2, we introduce notations, report on the basics of classical and variational autoencoders, and their application to physical field dimension reduction. In section 3.1, we propose a brief parametric study comparing the performance of a conventional autoencoder architecture with that of a variational autoencoder, using the POD as a reference. This comparison will enable us to assess the ability of these models to reconstruct flows and capture the main characteristics of the data. Several aspects will be taken into account in this comparative analysis, including the size of the latent space and the number of model filters. The aim is to determine the conditions for obtaining a parsimonious model, i.e. one that uses as few parameters as possible while maintaining satisfactory performance, in the lowest possible dimension. Next, in section 3.2, we will focus on the case of the flow around an oscillating cylinder, in periodic and quasi-periodic regimes, i.e. with a forcing frequency below and beyond the natural shedding frequency. We will use POD to have a first view into the principal components of these flows, then continue the analysis leveraging classical and variational autoencoders. We will demonstrate that autoencoders become truly efficient when they succeed in temporally aligning data points, during gradient descent learning. This property is essential for understanding the underlying dynamics of fluid flows and for extracting relevant information from latent space. An interpretation of the latent space will then be performed, where different characteristic states of the system will be identified and located in this space. This step is needed to demonstrate that autoencoders not only reconstruct data, but also provide valuable information on the dynamics of the system under study. While the interpretation is straightforward in the periodic flow case and will help as a pedagogical case, the quasi-periodic case already contains a rich vortex merging dynamics. Finally, the analysis will also be carried out on a turbulent flow case, in section 3.3: the von Kármán flow at Reynolds number $Re = 29000$. The von Kármán swirling flow is a canonical setup for experimental study of turbulence. It consists of two counter-rotating impellers driving a flow inside a closed cylinder. For the sake of simplicity, we limit the analysis to a single azimuthal Fourier mode, the most energetic passed the mean

flow field, representing the Kelvin-Helmholtz vortices [17]. This latter study will test the ability of autoencoders to capture complex turbulent phenomena, while remaining within a parsimonious framework.

2. Background

2.1. Notations

The physical field we analyse, denoted $\boldsymbol{u}(\boldsymbol{x},t)$, is a vector field of $n=2$ or $n=3$ velocity components, defined at each point of a two-dimensional space Ω_{phys}. The field is discretized in time and space on a Cartesian grid, and referred to with the following symbols indifferently:

$$\boldsymbol{u}(\boldsymbol{x},t) \equiv u_{tx_1x_2n} \equiv u_{tx}, \tag{1}$$

where: $t=1,\cdots,N_t$; $x_1=1,\cdots,N_1$; $x_2=1,\cdots,N_2$; $x=1,\cdots N$. N_t is the number of field snapshots, N_1 and N_2 are the number of discretization points in the first and second space direction, respectively, and $N=n\times N_1\times N_2$ is the total number of degrees of freedom taken into account for each recorded time. Whereas all these notations are equivalent, $u_{tx_1x_2n}$ has the direct meaning of the field discretization, and u_{tx} is the more abstract snapshot matrix. The scalar product hence reads:

$$\int_{\Omega_{phys}} \boldsymbol{u}(\boldsymbol{x},t)\cdot\boldsymbol{u}(\boldsymbol{x},t)d\boldsymbol{x} \equiv \sum_{x_1,x_2,n} u_{tx_1x_2n}u_{tx_1x_2n} \equiv \sum_x u_{tx}u_{tx}. \tag{2}$$

All along the paper, in an effort to compare methods, we will refer to N_{bf} as the total number of convolutional filters deployed in AEs (including all layers), and to the number of modes when talking about POD. The common ground is that both methods use some type of *basis functions* to describe the field and extract flow features. However POD is composing linearly the field and the modes, and the modes are orthogonal and defined globally on Ω_{phys}, whereas an AE is composing non linearly the field and the filters, which are local in space due to local receptive fields, and the filters are defined locally in the field.

2.2. Short introduction to autoencoders

An autoencoder is simply speaking a type of neural network with unsupervised learning that is used to compress and reconstruct data. It mainly consists of two parts: an encoder E that compresses the input data into a smaller representation and a decoder D that reconstructs the original data from this compressed representation. Both perform nonlinear mappings. Formally, we have:

$$\begin{aligned} &E:\mathbb{R}^N\to\mathbb{R}^d \\ &\boldsymbol{u}(\boldsymbol{x},t)\to \boldsymbol{Z}(\boldsymbol{\Theta}_E,t) \end{aligned} \tag{3}$$

$$\begin{aligned} &D:\mathbb{R}^d\to\mathbb{R}^N \\ &\boldsymbol{Z}(\boldsymbol{\Theta}_E,t)\to\tilde{\boldsymbol{u}}(\boldsymbol{\Theta}_E,\boldsymbol{\Theta}_D,\boldsymbol{x},t) \end{aligned} \tag{4}$$

such that $d\ll N$, and that E and D are neural networks parametrized by weights and biases that we gather in the following quantity: $\boldsymbol{\Theta}=(\boldsymbol{\Theta}_E,\boldsymbol{\Theta}_D)$. We will drop the explicit dependence for the sake of clarity in the notations, keeping in mind that we note $\tilde{\boldsymbol{u}}(\boldsymbol{x},t)\equiv\tilde{\boldsymbol{u}}(\boldsymbol{\Theta}_E,\boldsymbol{\Theta}_D,\boldsymbol{x},t)$, $\boldsymbol{Z}(t)\equiv\boldsymbol{Z}(\boldsymbol{\Theta}_E,t)$, and similarly for all the $\boldsymbol{\Theta}$-dependent variables. Then, we solve the following discretized optimization problem using an *AdamW* stochastic gradient descent step to evolve the $\boldsymbol{\Theta}$ parameters [18,19], see appendix A.1 for more details.

We now turn to the details of the E and D transformations. They are both made of two distinct parts: a Convolutional Neural Network (CNN) and a Multi Layer Perceptron (MLP). Representing the autoencoder architecture with a schematic, see Fig. 1a, one can identify the CNN parts on the exterior and the MLP parts in the middle. A MLP is composed of *dense* layers, which are fully connected layers, where all the input features are connected to all the output features: it acts on the field globally. A dense layer processes the field without the notion of locality, it acts on the field state vector $u_{tx}^{(\ell)}$ at layer ℓ as:

$$u_{ti}^{(\ell+1)} = \xi\Big(\sum_{j=1}^{N}\omega_{ij}^{(\ell)}u_{tj}^{(\ell)} + B_i^{(\ell)}\Big), \tag{5}$$

with $\omega_{ij}^{(\ell)}$ and $B_i^{(\ell)}$ the weights and biases of layer ℓ. ξ is a chosen activation function. Note that a dense layer with a choice of the point-wise activation function $\xi(x)=x$ simply accounts for a linear map. Using the hyperbolic tangent for ξ, a MLP becomes a sequence of nonlinear maps.

On the other hand, a *convolutional* layer of a CNN acts on the field locally, by taking the convolution of the field with a filter φ. Because the transformation takes into account the spatial structure, we denote the field processed at layer ℓ as $u_{thwc}^{(\ell)}\equiv u_{tx}^{(\ell)}$, h and w refer to height and width directions of the spatial domain at layer ℓ, with $N_x(\ell)=N_h(\ell)\times N_w(\ell)\times N_c(\ell)$ degrees of freedom in the total representation, and c is meant for channel, i.e. N_c different field filters. The function φ simultaneously filters all the different representations present at a given layer, it is defined on a $H_1\times H_2$ spatial domain. At each layer, we choose to train N_c different filters. A common choice is to take $H_1=H_2=3$, but it can be done differently, through the layers depth. A convolutional layer consists in applying $N_c(\ell+1)$ filters:

$$u_{tabc}^{(\ell+1)} = \xi\Big(\sum_{i=1}^{H_1}\sum_{j=1}^{H_2}\sum_{k}^{N_c(\ell)}\varphi_{ijkc}^{(\ell)}u_{t(a+i-G)(b+j-G)k}^{(\ell)} + B_c^{(\ell)}\Big), \tag{6}$$

with $G=\lfloor\frac{H_1}{2}\rfloor$, or $G=\lfloor\frac{H_2}{2}\rfloor$, using the floor function. This convolution operation is highly efficient, in the sense that it is local and involves considerably less parameters than a global dense operation, because one can always choose H_1 and H_2 such that $H_1\times H_2\ll N_h(\ell)\times N_w(\ell)$. Not only it decreases the computational cost but it also provides a representation which is equivariant under translations. For example, if one applies a shift $\boldsymbol{s}$ in input, we get a shift $\boldsymbol{s}'$ in output:

$$\begin{aligned} u_{tabc}^{(\ell)} &\to u_{t(a-s_1)(b-s_2)c}^{(\ell)} \\ u_{tabc}^{(\ell+1)} &\to u_{t(a-s_1')(b-s_2')c}^{(\ell+1)}. \end{aligned}$$

This property might turn out to be particularly relevant when flow structures experience translations, like in a von Kármán vortex street. It allows to recover with a hopefully small set of filters φ, all possible positions of a given structure. Finally, in this work, we have decided to use average pooling and up-sampling layers, to limit the number of convolution layers and keep a well resolved field in input.

2.3. Variational autoencoders

Contrary to classical autoencoders, variational autoencoders latent representations are constrained by some prior distribution, chosen by the user. This property can lead to a better organisation of the latent space, resulting in easier interpretation. VAE's have been introduced by Kingma in 2013 [20], as a Bayesian generative model. It leverages a statistical interpretation of autoencoders. Instead of reasoning on a single realisation of the reconstructed field $\tilde{\boldsymbol{u}}(\boldsymbol{x},t)$, we now consider all the possible realisations that follow the $p_{\boldsymbol{\Theta}}(\boldsymbol{u})$ probability given by the autoencoder, where this density is implicitly parametrized by the model weights $\boldsymbol{\Theta}$. The model ideal objective can thus be rewritten in terms of the field probability distribution function (PDF):

$$p_{\boldsymbol{\Theta}}(\boldsymbol{u}) \xrightarrow[\boldsymbol{\Theta}\to\boldsymbol{\Theta}^\star]{} p(\boldsymbol{u}), \tag{7}$$

where $p(\boldsymbol{u})$ is the true field PDF, the one from the data. This optimization can be achieved in principle by minimizing the Kullback-Leibler (KL) divergence:

$$\Theta^\star = \arg\min_{\boldsymbol{\Theta}} \int p(\boldsymbol{u})\log\frac{p(\boldsymbol{u})}{p_{\boldsymbol{\Theta}}(\boldsymbol{u})}d\boldsymbol{u}. \tag{8}$$

The general issue of this approach is that estimating $p(\boldsymbol{u})$ is not feasible, because $\boldsymbol{u}$ lives in dimension $N\gg 1$. However, using the joint probability:

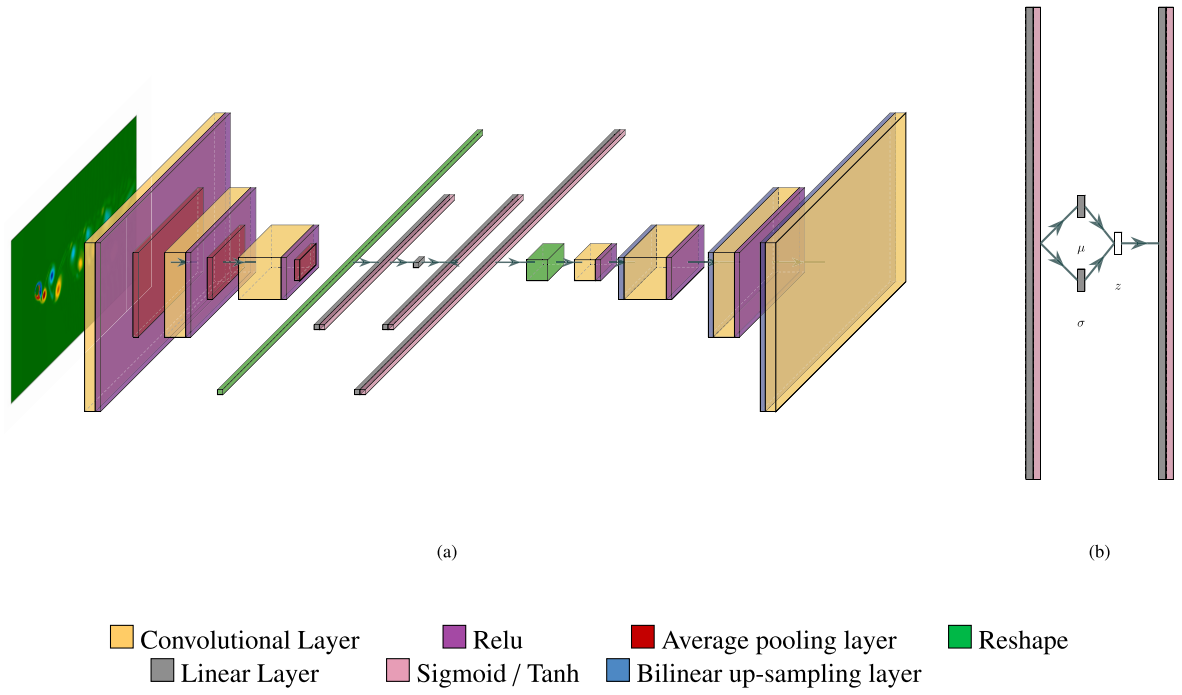


Fig. 1. Model architecture: (a) Convolutional AutoEncoder architecture, (b) Variational AutoEncoder setting. A vorticity field snapshot is represented for illustration purpose, although in practice we train the networks with velocity components input, i.e. we stack the $n = 2$ velocity components (u_x, u_y) for the 2D oscillating cylinder application, and we stack the $n = 6$ velocity components $(u_r^{c,m_F}, u_r^{s,m_F}, u_\theta^{c,m_F}, u_\theta^{s,m_F}, u_z^{c,m_F}, u_z^{s,m_F})$, for the von Kármán flow case, restricted to $m_F = 3$. The input mesh is 256×512 for each component. We use 3×3 sized filters, with 4×4 average pooling windows, with stride 4. Bilinear upsampling is done conversely. The filters are distributed among the layers in a symmetric fashion between the encoder and the decoder, with different schemes $[8, 16, 32]$, $[16, 32, 64]$, and $[64, 128, 256]$, amounting to a total number of filters $N_{bf} \in [112, 224, 896]$, respectively. We take 64 neurons in the inner dense layer, for the two first network sizes when $N_{bf} \in [112, 224]$, and take 128 neurons when $N_{bf} = 896$. In the encoder, the dense layer activation is sigmoid, and we do not use activation function at the latent space layer, to allow an $\mathbb{R}^2$ span. The decoder dense layers both have tanh activation, to keep the $\boldsymbol{Z} \to -\boldsymbol{Z}$ symmetry. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

$$p_{\boldsymbol{\Theta}}(\boldsymbol{u}) = \int p_{\boldsymbol{\Theta}}(\boldsymbol{u}, \boldsymbol{Z}) d\boldsymbol{Z} = \int p_{\boldsymbol{\Theta}}(\boldsymbol{Z}|\boldsymbol{u}) \frac{p_{\boldsymbol{\Theta}}(\boldsymbol{u}, \boldsymbol{Z})}{p_{\boldsymbol{\Theta}}(\boldsymbol{Z}|\boldsymbol{u})} d\boldsymbol{Z}, \tag{9}$$

we can write a Jensen inequality:

$$\log p_{\boldsymbol{\Theta}}(\boldsymbol{u}) \geq \int p_{\boldsymbol{\Theta}}(\boldsymbol{Z}|\boldsymbol{u}) \log \frac{p_{\boldsymbol{\Theta}}(\boldsymbol{u}, \boldsymbol{Z})}{p_{\boldsymbol{\Theta}}(\boldsymbol{Z}|\boldsymbol{u})} d\boldsymbol{Z}, \tag{10}$$

which, by expanding $p_{\boldsymbol{\Theta}}(\boldsymbol{u}, \boldsymbol{Z}) = p_{\boldsymbol{\Theta}}(\boldsymbol{u}|\boldsymbol{Z}) p(\boldsymbol{Z})$, gives:

$$\log p_{\boldsymbol{\Theta}}(\boldsymbol{u}) \geq \int p_{\boldsymbol{\Theta}}(\boldsymbol{Z}|\boldsymbol{u}) \log p_{\boldsymbol{\Theta}}(\boldsymbol{u}|\boldsymbol{Z}) d\boldsymbol{Z} - \int p_{\boldsymbol{\Theta}}(\boldsymbol{Z}|\boldsymbol{u}) \log \frac{p_{\boldsymbol{\Theta}}(\boldsymbol{Z}|\boldsymbol{u})}{p(\boldsymbol{Z})} d\boldsymbol{Z}. \tag{11}$$

Here, the first term in the right hand side can be estimated with the usual mean square error, whereas the second term is a KL term between a prior distribution $p(\boldsymbol{Z})$, chosen by the user, and a posterior distribution $p_{\boldsymbol{\Theta}}(\boldsymbol{Z}|\boldsymbol{u})$ given by the encoder. In the case of a unit Gaussian prior, $p(\boldsymbol{Z}) = \mathcal{N}(0, 1)$, the KL term can be computed in closed form. Then, since the KL is a positive valued functional, the VAE objective (8) can be approximated by the bound (11).

From a more pragmatic point of view, AE and VAE are very much alike: the MLP architecture is slightly modified (see Fig. 1b), and the latent representation $\boldsymbol{Z}$ is reparametrized to a univariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$:

$$Z_{tk} = \mu_{tk} + \epsilon_{tk} \cdot \sigma_{tk}, \tag{12}$$

where μ_{tk} represents the average coordinate of Z_{tk}, and σ_{tk} represents its standard deviation. The number ϵ_{tk} is drawn at random from $p(\boldsymbol{Z}) = \mathcal{N}(0, 1)$ which introduces the prior constraint and produces variability on Z_{tk} at fixed $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. Finally, the loss function expressed in (11) is approximated by:

$$\mathcal{L} = \frac{1}{N_t N} \sum_{t,x} (u_{tx} - \tilde{u}_{tx})^2 + \frac{\beta}{2d N_t} \sum_{t,k} (\mu_{tk}^2 + \sigma_{tk}^2 - \log \sigma_{tk}^2 - 1), \tag{13}$$

where the hyper-parameter β controls the relative weight of the KL term with respect to the energy residue. Note that the KL term is zero when $(\mu_{tk} = 0, \sigma_{tk} = 1)$ for all t, k, when the posterior and the prior match exactly. For more details in the derivations, see for example [20–22].

2.4. Data description

In the following, we use simulations performed with in-house codes, SUNFLUIDH [23] and SFEMaNS [24]. We first use the SUNFLUIDH database for the two-dimensional flow driven by a forced oscillating cylinder along the cross-flow direction at $Re = 185$. The imposed vertical motion of the cylinder takes the following form: $A \sin(2\pi f_e t)$, where A is half the cylinder diameter. At this regime, and under a forcing frequency equals to the natural shedding frequency, it has been shown that the flow does not exhibit three-dimensionality, see [25], making our analysis in principle directly comparable to experiments. The excitation frequency f_e will take two values: $f_e = 0.95 f_c$ and $f_e = 1.2 f_c$, with f_c the natural frequency at which the vortex shedding occurs when the cylinder is not moving. We first used the existing database [26],[1] then, we ran another simulation in the $f_e = 1.2 f_c$ flow case, dividing the sampling time by four, and computing for 2×10^4 snapshots. Details on the geometry and boundary conditions are available via [26].

[1] For $f_e = 0.95 f_c$ the sampling is approximately 13.8 snapshots per cylinder period $\frac{1}{f_e}$, which translates to $f_{sampling} \sim 14.5 f_c$. For $f_e = 1.2 f_c$ the sampling is approximately 10.9 snapshots per cylinder period $\frac{1}{f_e}$, which translates to $f_{sampling} \sim 9 f_c$.

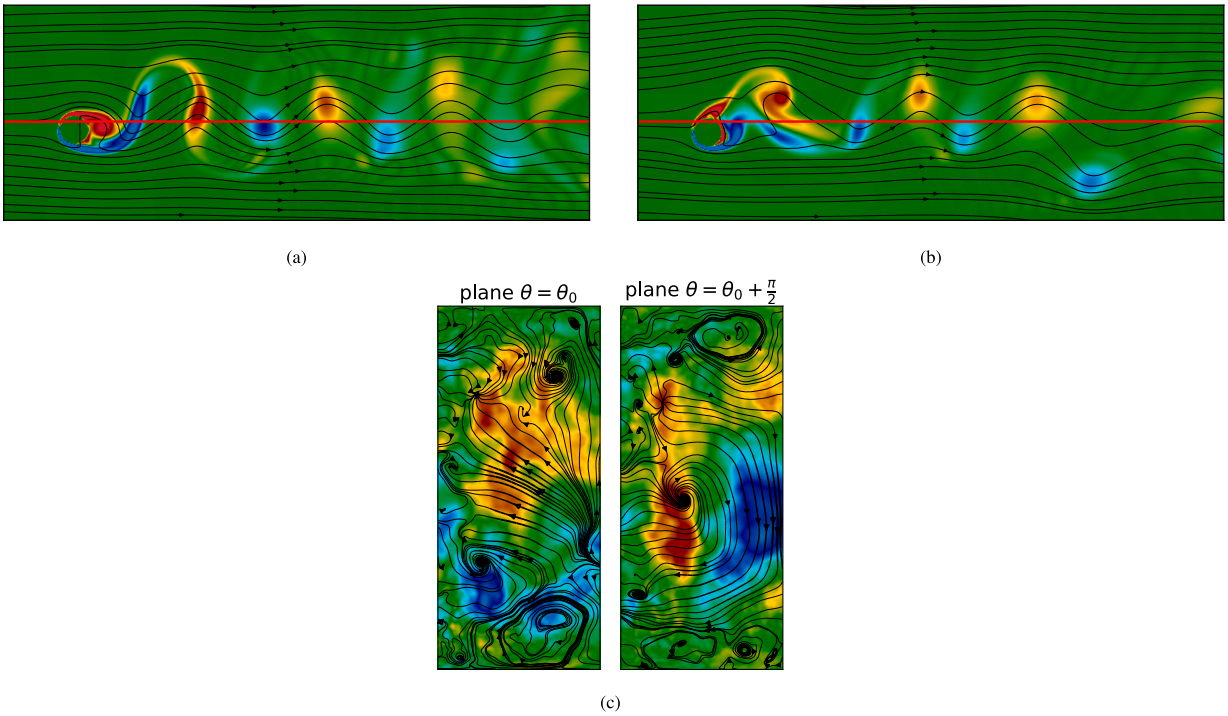


Fig. 2. Qualitative snapshots of the flows under study. Figures (a) and (b) show the flow around the transversely oscillating cylinder, in the periodic *locked-in* (a) and the quasi-periodic case (b), respectively. These snapshots are taken at times when the cylinder is almost at its lowest position, i.e. $y_{cyl} = -0.5$. Streamlines are drawn in black lines on the $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ flow components, while the colormap illustrates the vorticity field. The red horizontal line represents the $y = 0$ axis in the fixed frame of reference. Figure (c) shows a snapshot of the Fourier mode 3 of the turbulent 3D von Kármán (vK) swirling flow ($m_F = 3$, see (26)), taken at a given time in two orthogonal (r,z) planes. Streamlines are drawn on the $\boldsymbol{u}_r$ and $\boldsymbol{u}_z$ flow components, while the colormap shows the $\boldsymbol{u}_\theta$ component.

The von Kármán swirling flow has been simulated with SFEMaNS, at $Re = 29 \times 10^3$, all the details can be found in [17]. Since both flow configurations have immersed boundaries, modelled in the two codes by the Pasquetti pseudo-penalty method [27], the fluid domain depends on time. Consequently, we consider the fluid domain and the solid domain together, defining a velocity field on the full simulation domain which does not depend on time. In all the subsequent analysis, we chose to analyse directly the outputs of the codes: we did not remove the time averages from the fields and did not normalise the data.

The flows selected in this study share the common characteristic of being periodically forced, which we assume gives rise to distinct response frequencies. These frequencies serve as reference points in our analysis of how AEs encode flow structures. The complexity of the cases increases progressively: we begin with the periodic cylinder flow, dominated by a single temporal frequency expressed in two relevant spatial modes and associated with vortex shedding. We then consider a quasi-periodic case, where additional frequencies emerge due to vortex merging and advection, introducing greater dynamical complexity. These two cases are described in detail in Section 3.2.1. Autoencoders are first applied to the periodic case at lock-in (Section 3.2.2), and then to the quasi-periodic case (Section 3.2.3). Finally, the von Kármán swirling flow is introduced as a significantly more complex case, being fully turbulent at $Re = 29 \times 10^3$ (Section 3.3). For this case, we focus on a single Fourier component (identified in another publication [17]) that contains vortices generated by the Kelvin–Helmholtz instability, characterized by a dominant temporal frequency and a large number of spatial degrees of freedom associated with small scales. This choice is very natural as the CFD code that generated these three-dimensional data relies on a discrete Fourier representation along the azimuthal spatial coordinate. This makes it a stringent test for AEs, challenging their ability to reconstruct fine spatial structures and to represent multi-scale temporal dynamics in the latent space. See Fig. 2.

3. Results

3.1. Parametric and comparative study

We analyse in the first place the convergence of autoencoders (AE) and variational autoencoders (VAE) accuracy, through the following E_d quantity:

$$E_d = 1 - \frac{1}{N_t} \sum_{t}^{N_t} \frac{\|\boldsymbol{u}(\boldsymbol{x},t) - \tilde{\boldsymbol{u}}(\boldsymbol{x},t)\|_2^2}{\|\boldsymbol{u}(\boldsymbol{x},t)\|_2^2}, \tag{14}$$

where d is the dimension of the autoencoders' latent space. We recall that $\tilde{\boldsymbol{u}}$, Definition (4), refers to the estimation of the field reconstructed by the models. The results are compared with the same metric evaluated on the POD, where d is this time the number of POD modes used in the reconstruction:

$$\boldsymbol{u}(\boldsymbol{x},t) = \sum_{k=0}^{N_t-1} a_k(t)\boldsymbol{\phi}_k(\boldsymbol{x}) = \tilde{\boldsymbol{u}}_{POD}(\boldsymbol{x},t) + \sum_{k=d}^{N_t-1} a_k(t)\boldsymbol{\phi}_k(\boldsymbol{x}), \tag{15}$$

with $\boldsymbol{\phi}_k(\boldsymbol{x})$ and $a_k(t)$ the POD modes and time coefficients, respectively.

First, as shown on Fig. 3, classical AEs perform well for all the flows under study. For a fixed number of convolutional filters, increasing d leads to a better accuracy, with a quite rapid saturation reaching a plateau from $d \sim 5$. We see that for large enough d, increasing N_{bf} just increases slightly the plateau value, cf. Figs. 3a, 3b and 3c. This behaviour shows the relevance of decoupling the number of basis functions, N_{bf}, from the number of latent components d. With the POD, we expect in total as many modes as number of snapshots. So in this case, the number of basis functions, the dimension of the reduced space and the number of snapshots coincide: $N_{bf} = d = N_t$.

Autoencoders' major asset is to allow an efficient $d \ll N_{bf}$ compression, with no built-in constraints relative to N_t. Another remark is that we can consider that AEs use local basis functions (the filters), whereas POD use global basis functions (the eigenmodes). The sizes of the filters

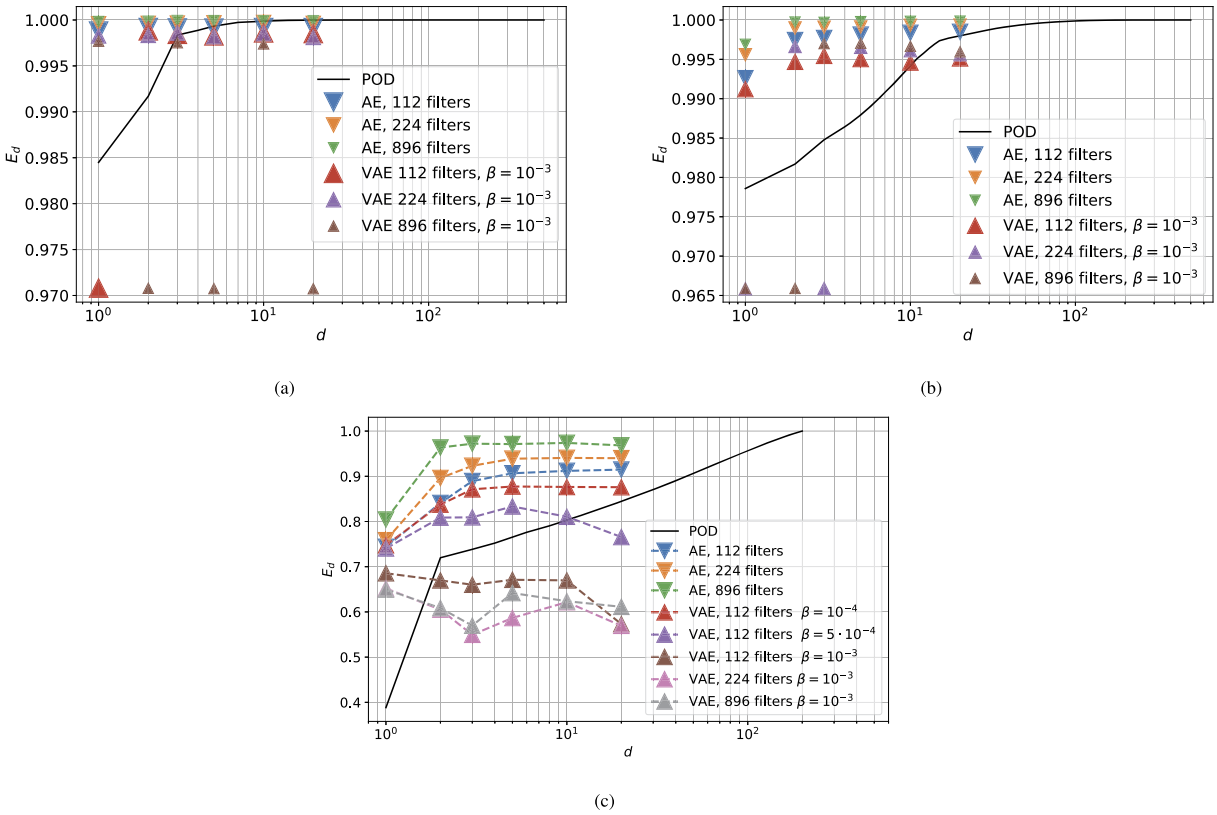


Fig. 3. Accuracy comparison between AE, VAE and POD as the dimension of the representation k is increased. (a) oscillating cylinder case at lock-in, (b) oscillating cylinder quasi-periodic case, and (c) von Kármán swirling flow.

are often limited to a small number of coefficients, like (3×3) or (7×7), which is enough to get a great accuracy, whereas the number of parameters of POD modes is compelled to be the number of spatial discretisation points of the fluid flow. In our case, the fluid snapshots are all defined on a $(N_x, N_y) = (256, 512)$ meshgrid, and the total number of POD parameters to store and process all modes, scales as $N_t \times N_x \times N_y \times n$. At the end, autoencoders numerically scale better than POD, leveraging the decoupling between the global representation of the latent space, that aggregates all the system, and the local basis functions. They are also more efficient in gathering the number of *governing* coefficients in small latent spaces with respect to POD (see the black line in Fig. 3 and compare with AEs).

On the other hand, VAE performances show more complex behaviours that completely change with respect to the β value. It appears that sometimes the models fail to find a useful representation of the data, even on the simpler periodic cylinder flow (Fig. 3a), as well as on the quasi-periodic cylinder flow (Fig. 3b). When the training converges to a descent local minimum, the performances are still less than the classic AE. We detail the behaviour on the turbulent von Kármán swirling flow (Fig. 3c). For $\beta = 10^{-3}$, the VAE models perform poorly compared to POD whatever the number of filters N_{bf}. Conversely, decreasing β increases the accuracy. A more formal β analysis is conducted on the quasi-periodic cylinder flow and on the turbulent von Kármán swirling flow, by varying β at fixed $N_{bf} = 112$ for different latent space sizes d (Figs. 4a and 4b). As it turns out, there is a sharp transition between a *not learning* regime and a *learning* regime around a critical value β_c, that depends on the underlying data. The fact that decreasing the β value increases the performance, also shown to some extent in [12] and discussed in [5], is a critical issue for the construction of robust VAE methods. The gain of using VAE, as argued in the aforementioned articles, is that increasing β decreases the correlation between the latent variables and makes easier the interpretation. The resulting *orthogonal representation*, i.e. one that enables flow features to be represented in a linearly independent way, should indeed allow direction by direction sampling of the latent space to visualize these features. The resulting interpretation is not always straightforward while looking at reconstructed fields [12,10,2]. However, this logic must accept the trade-off between reconstruction accuracy and obtaining a so-called disentangled latent space.

With VAEs, each snapshot has an average coordinate in the latent space, together with an assigned variance. This choice is built in the VAE architecture (Fig. 1b), and setting the variance to zero leads back to the classical AE architecture. To have an idea of the latent space variance relation with respect to β, we can compute an average variance $\langle \sigma_d \rangle = \frac{1}{d \cdot N_t} \sum_{t=1}^{N_t} \sum_{k=1}^{d} \sigma_{tk}$, where σ_{tk} is the variance of time t snapshot on latent dimension k, see equation (12). The variance $\langle \sigma_d \rangle$ is always one (the optimisation objective) in the *not learning* regime when β is too high, then it decreases when β decreases in the *learning* regime (Figs. 5a and 5b). Larger latent space models, e.g. for $d = 20$, keep some relatively high variance, together with high accuracy, which is a profitable behaviour. However, the data points in higher dimensional spaces are getting rapidly distant from each other (Figs. 5a and 5b). It is visible on the average distance between the latent space points $\langle D_d \rangle = \frac{1}{N_t^2} \sum_{t,t'}^{N_t} \sqrt{\sum_{k=1}^{d} (\mu_{tk} - \mu_{t'k})^2}$. The *not learning* to *learning* transition is also visible on the averaged distance of the latent space data points and on VAE averaged latent space variance. However, it is very likely that the network initialization matters, as well as the gradient descent step size, in order to go in the learning regime at higher β. In particular, we observed that reducing the gradient descent step size allows to go from the *not learning* to the *learning* regime for some fixed β. Further hyper-parameter exploration is still needed to estimate the scaling of this behaviour. Overall, VAE models only perform well when $\langle \sigma_d \rangle$ / $\langle D_d \rangle$ is small, meaning that the encoded snapshots are far enough apart that their probability envelopes do not overlap too much. This fact is relevant to our objective of analysing the effective dynamics of fluid flows.

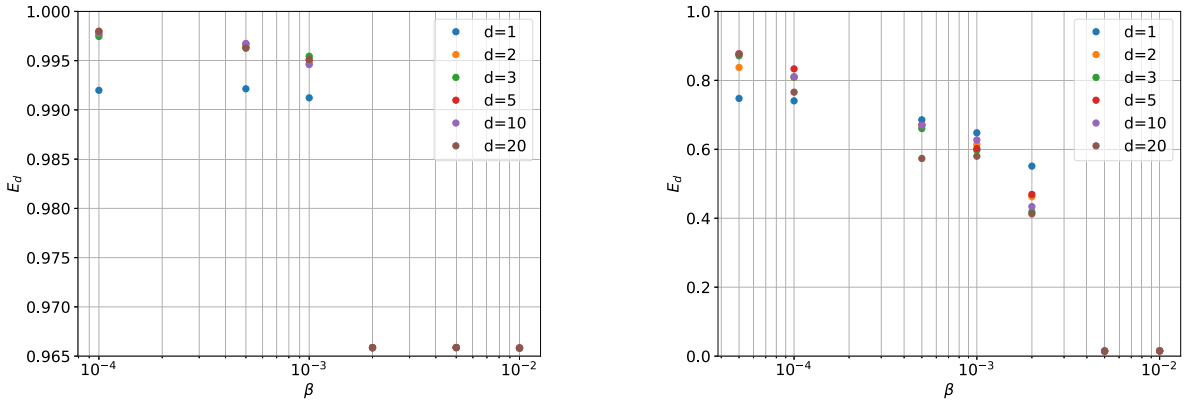


Fig. 4. Dependence of the VAE accuracy with β for: (a) oscillating cylinder in the quasi-periodic case, and (b) von Kármán swirling flow. We report that the threshold value β_c depends as well on the learning rate, i.e. the gradient descent step size in the parameter space. Using smaller steps increases β_c. This threshold can also depend on the model number of parameters together with the number of snapshots used for training. However it seems not to depend on the latent space size, because the (E_k, β) dependence does not depend significantly on k, so that several points are indistinguishable.

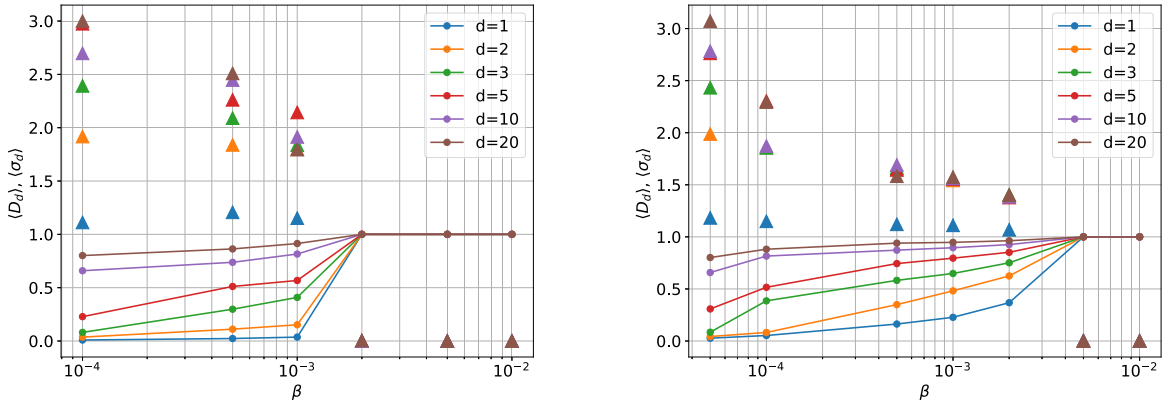


Fig. 5. VAE average variance $\langle \sigma_k \rangle$: circle markers + solid lines, and average distances between points $\langle D_k \rangle$: triangles, for (a) oscillating cylinder in the quasi-periodic case, and (b) von Kármán swirling flow. Note that we represent these two quantities in the same plot to show how they vary with the latent space size. The ratio $\langle \sigma_k \rangle / \langle D_k \rangle$ approximately collapses on one single curve, suggesting that the same process is happening in the network at the transition point β_c whatever the latent space size k. The regime for which the models are efficiently reconstructing the data is for $\langle \sigma_k \rangle < \langle D_k \rangle$, see Fig. 4.

If one wants to get the notions of trajectory in the latent space, the snapshots have to be correlated in time to see a smooth movement. Instead, the point of view of VAE, i.e. having a set of mean and variances that describes the flow global characteristics and variabilities, could be more delicate to hundle for dynamical systems, where trajectories matter. We will overcome this difficulty by considering trajectories described by the VAE mean μ_{tk}. However, the β analysis is clear: smaller β lead to more accurate models. An important matter is to know whether small β values give anyway the expected disentangling behaviour. Finally, as a generative statistical model, VAE could prove more useful for grouping multimodal behaviours, when snapshots are not correlated over time and the notion of trajectory matters less than the probability of having a given state. Another useful case for VAE might also be the aggregation of flow snapshots taken from different parametric applications of the governing equations.

To conclude this parametric analysis, we select the most parsimonious models, those with the least number of filters possible and the best E_k accuracy. Thus, we will investigate the medium case with $N_{bf} = 224$, for $d = 1$ and $d = 2$ for the periodic flow case (Fig. 3a), and with $d = 2$ only for the quasi-periodic case and for the von Kármán swirling flow. We will continue the comparative study between AE and VAE in the next sections in order to quantitatively and qualitatively characterize the representation learnt. To this end, we will first analyse the spatio-temporal characteristics of the flow, using POD, AEs and VAEs and show how autoencoders behave during training by projecting the intermediate model reconstruction onto the POD. In particular, we will focus on the monitoring of the most relevant energetic features. Next, we will pursue an analysis of the resulting latent spaces, once the training is complete, to understand how the dynamical system is encoded. Special attention will be paid to the expression of the field symmetries in the various latent spaces.

First, we will look at the flow around the oscillating cylinder, then the case of the von Kármán swirling flow.

3.2. Oscillating cylinder

3.2.1. Description of the flow

Here we consider canonical cases of laminar flows around a circular cylinder that exhibits oscillations. Frequency contents, vortex shedding modes and cylinder motion amplitudes are linked together and their combinations provide an interesting range of flows for test cases, e.g. [28,29]. In the case of a sinusoidally forced oscillating cylinder, several flow regimes arise due to the interplay between the *natural* vortex shedding of frequency f_c and the imposed motion and excitation frequency f_e of the cylinder. In this case forcing frequency and forcing amplitude, as well as Reynolds number are the critical parameters. We distinguish

the lock-in (or synchronisation) regime, where the vortex shedding frequency f synchronizes with the oscillation frequency f_e of the cylinder. This occurs when the oscillation frequency is close to the natural shedding frequency. For the particular case for which $f_e = 0.95 f_c$ then we recover the frequency of a stationary cylinder $f = f_e$ and the flow dynamically adjusts, leading to a stable phase relationship between the motion and the shedding. This synchronization can result in large *periodic* oscillation amplitudes and significant fluctuating forces due to resonance-like effects. In this case we get a wake with a 2S mode organisation: it is the classic von Kármán vortex street with two single vortices shed per oscillation cycle. Another interesting regime occurs at forcing frequencies further from the lock-in range. In this case the interaction between the cylinder motion and natural vortex shedding becomes more irregular. This results in *quasi-periodic* or chaotic wake structures characterized by a combination of frequencies without synchronization. These regimes are marked by irregular vortex shedding and more complex wake dynamics.

In both cases, almost all of the flow energy is captured by the first three POD modes, with more than 99.5% in the periodic flow case, and around 98.5% in the quasi-periodic case (see Figs. 3a and 3b). These modes represent the emission and advection of upward and downward vortices of the main wake, equally spaced in space and time. The $n_P = 0$ POD modes (Figs. 6, 7) gather the main steady flow component $u_x = \mathcal{O}(1)$, and show a small suction ($u_x < 0$, $u_x = \mathcal{O}(0)$) near the cylinder. The colorbars extreme values are set to $(-M_{ij}, M_{ij})$ with:

$$M_{ij} = \max\left(\left| \min(\boldsymbol{\phi}_i(\boldsymbol{x}) \cdot \boldsymbol{e}_j) \right| , \left| \max(\boldsymbol{\phi}_i(\boldsymbol{x}) \cdot \boldsymbol{e}_j) \right| \right), \tag{16}$$

and $\boldsymbol{e_j}$ the streamwise (or spanwise) basis vector. The only exception is for the e_x direction of the $n_P = 0$ mode: because it is mainly positive valued, the colorbar is normalised between the minimum and the maximum value of the mode. Modes $n_p = 1$ and $n_p = 2$ are spatially and temporally periodic and out of phase, and reconstruct the vortices of the wake as well as the approximate cylinder position. Because the POD has been constructed in the laboratory frame of reference, the changing cylinder position is represented by a lot of modes of weak energy along the POD spectrum.[2]

In the periodic flow case, while the $n_p = 1$ and $n_p = 2$ modes only contain the cylinder oscillating frequency $f_e = 0.95 f_c$ (see temporal spectra in Fig. 8a), the $n_p = 0$ mode carries mainly the $f = 0$ frequency, and some $f = \frac{1}{2} f_e$ and $f = 2 f_e$. Higher POD modes come in pairs and are harmonics of f_e. Spatially, the higher modes describe finer and finer scales near the cylinder, describing the boundary layer separation, as well as the wake modulation (see Fig. 6).

The quasi-periodic flow case, $f_e = 1.2 f_c$, is more complex. First, the shape of the wake is different, with notably more aspiration in the near cylinder region visible on the $\boldsymbol{e}_x$ component of the POD mode $n_p = 0$ (Fig. 7). This suction destabilizes the wake. The pair ($n_p = 1$, $n_p = 2$) now only describes the wake's vortices and does not encode the cylinder position (Fig. 7). Furthermore, we see that the vortices' advection is slowed down on the POD temporal spectra, for which ($n_p = 1$, $n_p = 2$) frequencies are now $f_{adv} = 0.86 f_c$. Another set of modes, mainly $n_p = 3$ and $n_P = 4$, pulses at the cylinder frequency $f_e = 1.2 f_c$, and describes globally the position of the cylinder and part of vortex shedding (Fig. 8b). Thus, the vortex advection and production processes have become desynchronised in the quasi-periodic case whereas, at lock-in, the two processes are synchronised and described by a single pair of POD modes. The quasi-periodic wake is then fed by more vortices than it can advect, so that vortex merging occurs (see video in supplementary material). The upward and downward vortices have opposite signs, (−) and (+), respectively. When two vortices of the same sign merge, a vortex is then deflected out of the wake. After vortices have interacted, the wake reflection symmetry is broken, and some residual vorticity is advected and dissipated without mixing too much until it gets out of the domain. The far wake is described by the $n_P = 5$ and $n_P = 6$ modes whereas the wake asymmetry, thus the deflected vortices advection, is described by the most relevant modes $n_p = 7$, $n_p = 8$, $n_p = 9$, that are characterized by large spatial scale as well as by low time frequencies (Fig. 7 and Fig. 8b). In particular, the $n_p = 8$ and $n_p = 9$ modes share a characteristic frequency at around $f_d = f_e - f_{adv} \sim 0.34$, which is in fact the frequency at which a vortex deflection event occurs. Lastly, the $n_P = 5$ and $n_P = 6$ modes share dominantly the frequency $f_{adv} - f_d \sim 0.53$ and have reflection symmetry, thus they represent a slowing down, dominantly in the far wake. These flow configurations have been studied previously in [30]. The periodic flow can be labelled 2S because two single vortices are shed in one period $1/f_e$. In contrast, quasi-periodic flow enters the *coalescence* regimes described in [31].

3.2.2. Autoencoders-type analysis of the flow at lock-in

We would now like to analyse the way in which AE and VAE map physics in latent space. As a pedagogical case, we start with the periodic flow regime with latent space dimensions of dimension either $d = 1$ and $d = 2$. The periodic flow case is a good proxy, because its periodicity can be easily found in the latent space. Here, we are clearly in the I_1 task defined in the introduction: we know that the flow is periodic, thus, in one dimension, the encoder should provide some representation of the phase of the system.

By highlighting in the latent space the coordinates of the snapshots at which the cylinder reaches its minimum and maximum amplitude, we can see that these points are almost perfectly aligned for the AE and for the VAE. They approximately share the same value in the latent space (see Figs. 9a and 9b). For both models, set with $d = 1$, we obtain some kind of periodic solution where the system jumps, in one time step, from maximum Z_1 to minimum Z_1, or *vice versa*. However, there is some regularity difference between the two encodings: in the AE case, the Z_1 distance between two consecutive encoded snapshots is not constant, whereas it tends to be constant for the VAE encoding. We refer to this property as a *time interval conservation*. The consequence is that the VAE encoding can be easily described, e.g. by a tangent function $Z \sim \tan(t)$, whereas the time evolution is a more complex function in the AE case.

We now look at the models for which the latent space size is set to [3] $d = 2$. We find that the AE encodes time with an angle $\theta = \arctan\left(Z_2 \,/\, Z_1\right)$ running in $[-\pi, 0]$, and the VAE encodes time with θ running in $[-\pi, \pi]$ (see Figs. 10a and 10b). Globally, both latent spaces have encoded a reflection symmetry in the Z_1 direction. Furthermore, we see from the highlighted points that the reflection symmetry in the latent space is equivariant to the y reflection symmetry in the physical space. Let us call $\mathcal{R}_y$ the velocity field reflection symmetry action:

$$\mathcal{R}_y : \begin{cases} u_x(x, -y, t) \longrightarrow +u_x(x, y, t), \\ u_y(x, -y, t) \longrightarrow -u_y(x, y, t), \end{cases} \tag{17}$$

and the latent space reflection symmetry action: $\mathcal{R}_{Z_1} \circ Z_1 = -Z_1$, then we have approximately the equivariance relation:

$$Z_1(\mathcal{R}_y \circ \boldsymbol{u}) = \mathcal{R}_{Z_1} \circ Z_1(\boldsymbol{u}). \tag{18}$$

In addition, the VAE latent space has another equivariance relation which is related to time. Analysing in more details the VAE latent space of Fig. 10b, it appears that time flows following the clockwise direction on the manifold (Z_1, Z_2). Thus the $\mathcal{R}_{Z_2}$ reflection symmetry can

[2] We have checked that performing a POD on corrected velocity fields (i.e. velocity fields minus the cylinder velocity) suppresses these modes but adds the same contribution $u_y(t) = 2\pi f_e A \cos\left(2\pi f_e t\right)$ to the u_y components of the $n_P = 1$ mode. This change did not substantially affect the shape of the other modes, so we kept the POD within the laboratory frame of reference.

[3] Note that models obtained with $d = 1$ do not provide a Z_1 representation which is similar or redundant to the ($d = 2$)-models. By setting $d = 2$ the network architecture is changed, and the model is trained from scratch. Consequently, the optimal weights are different, and the latent coordinates behaviours are different.

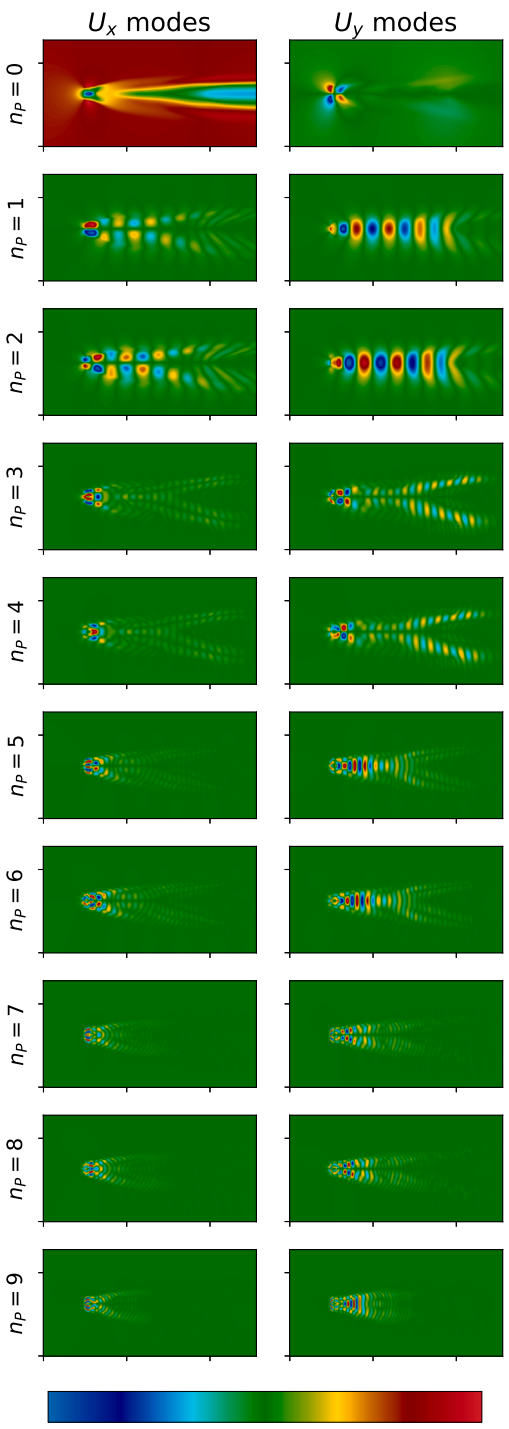


Fig. 6. Periodic flow POD modes: velocity components.

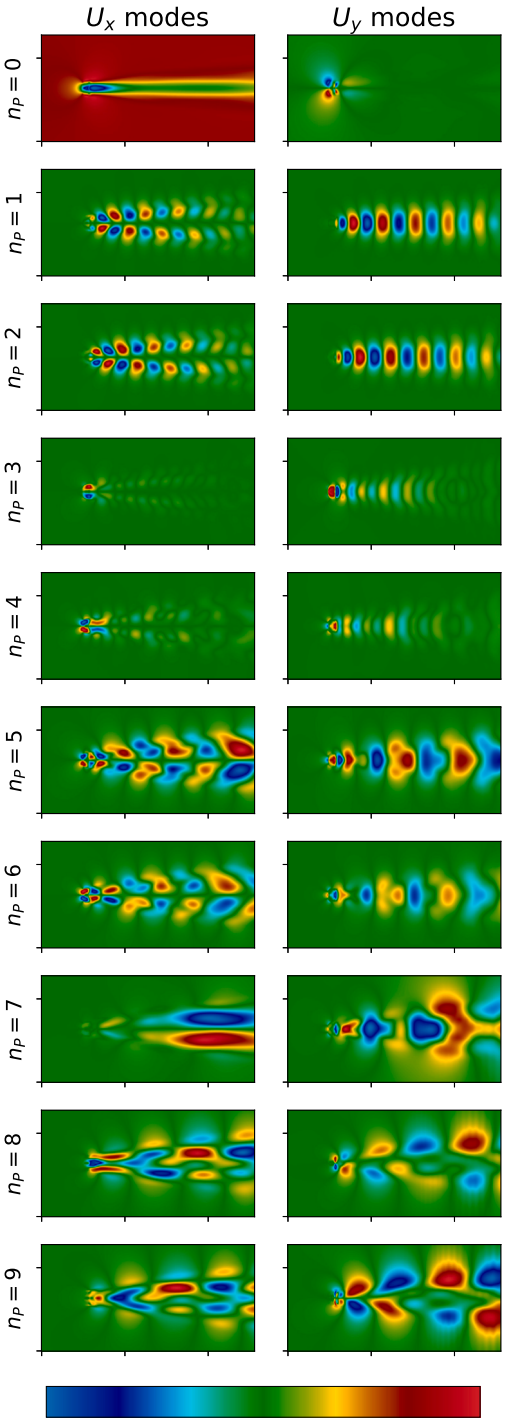


Fig. 7. Quasi-periodic flow POD modes: velocity components.

be associated with a time reflection $\mathcal{R}_t$ in physical space. Denoting the cylinder position as $y_{cyl}(t) = A \sin \frac{2\pi t}{T}$, one can define a given y_{cyl} maximum at time τ, given by:

$$\tau = \left(\left\lfloor \frac{t}{T} \right\rfloor + \frac{1}{4} \right) \cdot T, \tag{19}$$

such that:

$$\mathcal{R}_t : \boldsymbol{u}(\boldsymbol{x}, t) \longrightarrow \boldsymbol{u}\Big(\boldsymbol{x}, t + 2 \cdot (\tau - t)\Big). \tag{20}$$

Then the equivariance relation reads:

$$Z_2(\mathcal{R}_t \circ \boldsymbol{u}) = \mathcal{R}_{Z_2} \circ Z_2(\boldsymbol{u}). \tag{21}$$

Indeed, in the VAE case, these symmetry schemes are only valid *on average*, i.e. considering μ_{tk} instead of the reparametrized Z_{tk}, cf. equation (12).

Summarizing the previous analysis: both AEs and VAEs latent spaces are straightforward to interpret, the flow being periodic in time. However, VAEs better express the velocity field symmetries and produce smoother trajectories, described by the μ_{tk} variable. Another fundamental difference between AEs and VAEs is that AE latent spaces are discrete, in the sense that the space is defined only at the encoded snapshot points where, in VAE, spaces are continuous in the sense that all the latent space points are defined thanks to the Gaussian reparametrization. Yet,

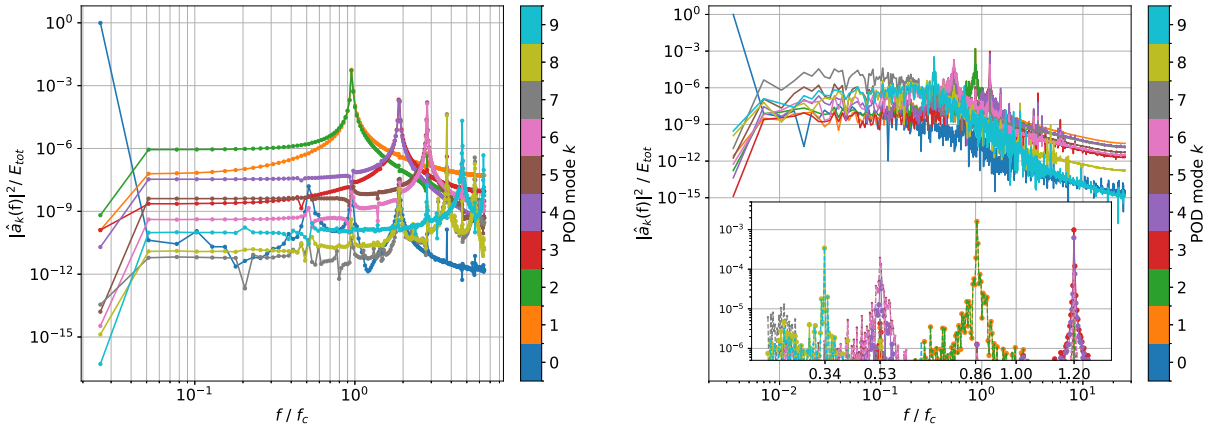


Fig. 8. POD time coefficient Fourier spectra: (a) oscillating cylinder case at lock-in, (b) oscillating cylinder, quasi-periodic case.

to qualify these results, it is important to keep in mind the difference in final accuracy: the gain in symmetry in VAEs in fact translates in a loss of accuracy, see Fig. 3a. For example, while watching the latent space along the training phase (video in supplementary material), we can note that the AE latent space first gets a unique smooth curvilinear trajectory, just like the VAE, before it starts to distinguish between the different phase periods. This is in fact happening because the snapshot sampling time is not a multiple of the shedding frequency, and consequently, each snapshot is unique in the dataset.

In order to understand how, and quantify which flow structures are captured during the learning process, we propose to project the field $\tilde{\boldsymbol{u}}$, at several training times, on the true POD basis $\{\phi_k(\boldsymbol{x})\}$ [4]:

$$\tilde{a}_k(t) = \int \tilde{\boldsymbol{u}}(\boldsymbol{x},t)\cdot\boldsymbol{\phi}_k(\boldsymbol{x})d\boldsymbol{x}, \tag{22}$$

to estimate a *per mode* accuracy:

$$e_k = \max\left(0, 1 - \frac{\sum_t (a_k(t)-\tilde{a}_k(t))^2}{\sum_t a_k(t)^2}\right). \tag{23}$$

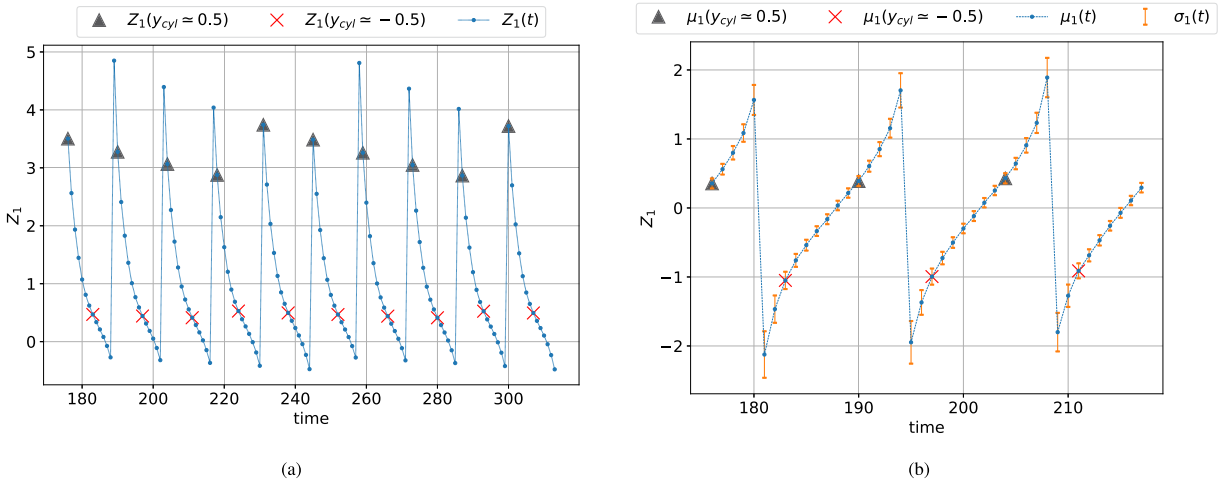
We display these quantities as a function of the training epochs for the AE case on Fig. 11a and for the VAE on Fig. 11b in the $d=2$ case. The results are striking and show that, from the POD perspective, the AE learns sooner and better than the VAE. To be fair, it should be emphasized that VAE *learning jump* can occur at different epochs depending on the choice of the learning rate η: i.e. reducing η increases the probability to trigger the learning earlier, at a lower epoch. Second, while the AE learns all POD modes equivalently, with a rate that follows their energy, the VAE fails to encode successfully high order modes, with the emergence of a hierarchy that follows the mode's energy. Finally, it can be argued in favour of the VAE that the POD modes beyond $n_P = 2$ are irrelevant, as they carry very little energy and information about the flow. Eliminating irrelevant information results in greater symmetry, and therefore greater interpretability. In a way, we can consider that the AE is overfitting the data, by distinguishing between different periods, and that it does not help the interpretation. Indeed, to counter this issue, we can select an AE model at earlier time. In conclusion, both models have their pros and cons. The AEs have better accuracy, and the VAEs have a better latent representation. This trade-off, between accuracy and representation symmetry, is summarized in the β parameter, that allows to control the VAE regularisation.

[4] The POD basis is computed from the simulation data directly, such that it is exact: $\boldsymbol{u} = \sum_{k=0}^{N_t-1} a_k(t)\boldsymbol{\phi}_k(\boldsymbol{x})$.

3.2.3. Autoencoders-type analysis of the quasi-periodic flow case

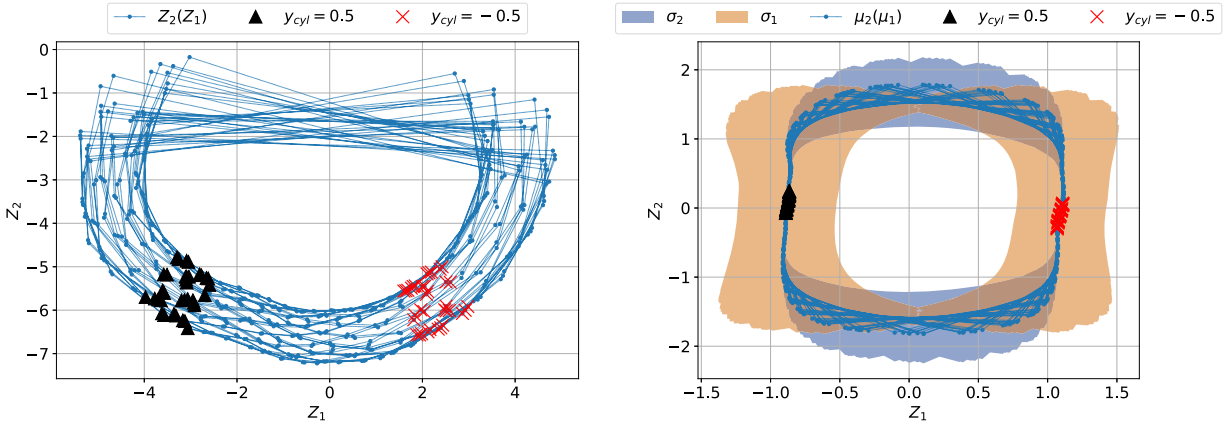
Now that the essence of AEs and VAEs is illustrated, we tackle the quasi-periodic flow case. We first trained a set of models considering a restricted number of flow snapshots $N_t = 500$ because it was more convenient to conduct the parametric study. Then we increased the number of training snapshots to $N_t = 15\,000$ to get confidence in the resulting latent space analysis. For the sake of comparison, we display the $d=2$ latent space resulting from the $N_t = 500$ training, for both the AE and the VAE on Figs. 12a and 12b, respectively. We observe again in the AE case that the model encodes the system trajectory and aligns similar snapshots along fixed $\theta = \arctan\left(Z_2 / Z_1\right)$ directions. This alignment property is less visible in the VAE latent space, that seems to split in two hemispheres, approximately defined by $Z_1 < 0$ and $Z_1 > 0$. In constrast with the periodic flow case, no particular symmetry can be identified, as both the aforementioned symmetries have been broken: the flow is not periodic anymore, nor has reflection symmetry in the y direction because of the emission of deflected vortices out of the wake, see section 3.2.1. Comparing with models trained with $N_t = 15\,000$, one sees that the shape of the latent space manifold changed consequently in the AE case (Fig. 13a) and conserved the hemispherical structure in the VAE case (Fig. 13b). Indeed, the shape of the VAE latent space is constrained to be a normal probability $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\sigma})$, but the same split around $Z_1 = 0$ is present. Again, no particular symmetry can be found on the latent space trajectories. The per-mode training accuracy globally shows the same behaviour as before: the increase in VAE accuracy comes later and rises more abruptly than for the AE (Figs. 14a and 14b). The most energetic structures are learnt first, except in the AE where $n_P = 7$ is present very early in the training, before the other modes. The particularity of $n_P = 7$ is that it carries the lower time frequencies, right after $n_P = 0$, see Fig. 8b. Looking in more details to Fig. 14a, we see that the POD mode pairs $(n_P = 5, n_P = 6)$ and $(n_P = 8, n_P = 9)$ start also to be learnt before the main energetic pair $(n_P = 1, n_P = 2)$. Thus, it appears that not only the most energetic structures matter in the training but also the low time frequencies are relevant. In the VAE case, everything happens as if there were a training epoch where suddenly the network starts learning all the modes at once, see Fig. 14b. Then, after quite some training time, the main pair $(n_P = 1, n_P = 2)$ is the best represented one, followed by the $(n_P = 8, n_P = 9)$ and $(n_P = 5, n_P = 6)$ pairs, a behaviour present for both models. Thus AEs and VAEs reconstruct best the most energetic modes, then at equivalent energy, they will select the modes with the lowest time frequencies. Indeed, at convergence, all the modes are well learnt, which means that all the snapshots are well reconstructed.

In order to carry the analysis further, we propose a way to interpret latent spaces, compatible with the I_1 task: we want to understand how the different modes of the flow are encoded. To this end, we first identify a particular subdomain Ω, bearing some structure in the latent



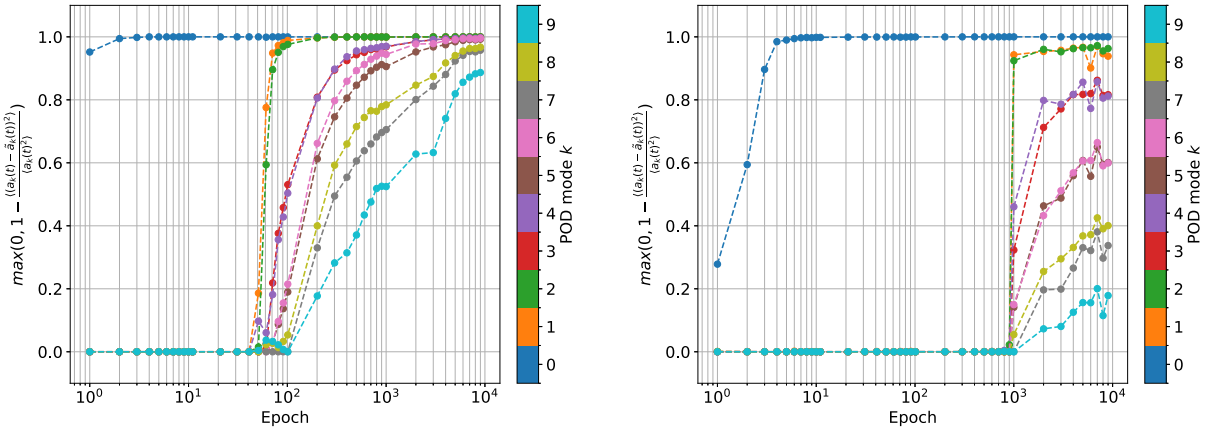
(a) (b)

Fig. 9. 1D Latent space visualization for the periodic flow case: (a) standard autoencoder, (b) variational autoencoder at $\beta = 10^{-3}$.



(a) (b)

Fig. 10. 2D latent space visualization for the periodic flow case: (a) standard autoencoder, (b) variational autoencoder at $\beta = 10^{-3}$.



(a) (b)

Fig. 11. Convergence of autoencoders ($d = 2$) along their training in the light of their POD contents for the cylinder lock-in case for: (a) AE, (b) VAE at $\beta = 10^{-3}$.

space manifold described by the training snapshots trajectories, directly corresponding to a set S of snapshots:

$$S_\Omega = \{\boldsymbol{u}(\boldsymbol{x},t) \mid \boldsymbol{Z} = E(\boldsymbol{u}) \in \Omega\}\,. \tag{24}$$

Once we hold this set of snapshots, it is easy to project them onto the POD basis of the original data and recover their POD coefficients:

$$A_k(\Omega) = \left\{ \int \boldsymbol{u}(\boldsymbol{x},t)\cdot\boldsymbol{\phi}_k(\boldsymbol{x})d\boldsymbol{x},\ \mid \boldsymbol{u}(\boldsymbol{x},t) \in S_\Omega \right\}. \tag{25}$$

It is then interesting to check which POD modes are clustered in Ω. We illustrate this interpretation method by splitting in two the latent spaces of Figs. 13a and 13b. The splitting criterion is set to the particular $Z_2 = \frac{1}{N_t}\sum_{t=0}^{N_t} Z_2(t)$ line for the AE space, and is chosen for the VAE space to isolate the two hemispheres from each other. These criteria are illustrated on the figures with dashed purple lines. Note that these splitting criteria approximately match the ones given by a K-mean algorithm [32], setting the number of clusters to 2, as we have checked *a posteriori*, see appendix A.3. In the AE case, the split between two regions Ω_{top} and Ω_{bot} induces a separation in the distribution of modes $(n_P = 8, n_P = 9)$, see Fig. 13a. As a reminder, these modes represent the vortices that are deflected out of the wake due to vortex merging. This means that a state $s_1 = (n_P = 8 < 0, n_P = 9 > 0)$ is dominant in Ω_{top}, corresponding to a strong aspiration in the near wake, and equivalently a state $s_2 = (n_P = 8 > 0, n_P = 9 < 0)$ is dominant in Ω_{bot}, corresponding to propulsion in the near wake, see u_x component in Fig. 7. The deflected vortices advection process is then represented by an oscillation between states s_1 and s_2, recovering some notion of symmetry $s_1 \sim -s_2$. The same analysis is more striking on the VAE space. This time, we define two regions Ω_{left} and Ω_{right} (purple dashed line delimiter on Fig. 13b), that effectively split the $(n_P = 5, n_P = 6)$ pair distribution, see Figs. 15a and 15b. The symmetry between the two identified states appears more clearly than in the AE example.

This methodology can also apply to task I_2 for which the objective is to extract new information from the latent space analysis. One can imagine a protocol to automatically compute a partitioning of the space, for example using a clustering algorithm, resulting in a set of $\{\Omega_i\}$ for a number i up to a given or learnt number of clusters. Then, using a statistical metric on $\{A_k(\Omega_i)\}$ like the Kullback-Leibler divergence or the mutual information, it might be possible to automatically identify and qualify the resulting states that are relevant in the field dynamics. We let the application of this idea to future work, as we yet have material to cover, demonstrating the easier task I_1. By a way of example, we applied K-means algorithm to Fig. 13b latent space with 4 clusters, that was found to be approximately the four quadrants of the (Z_1, Z_2) plane and that separate in addition s_1 from s_2. An apparent drawback of working here in dimension $d = 2$ is the question of generalization to larger d. However, most clustering algorithms work fine in $d > 3$ dimensions, and there are various dimension reduction methods, such T-SNE [33] or UMAP [34], which are affordable from dimension $d \ll N$, that can be used to visualize the resulting clusters in two or three dimensions. To conclude on the I_2 task, it is not faithfully applicable to our present fluid flow analysis since we willingly studied periodically forced fluid flows to get some prior knowledge. Indeed, looking at the different latent spaces, it is easy to see that there is a simple periodic behaviour in the lock-in case, and a more complex multi-periodic behaviour in the quasi-periodic case.

3.3. The von Kármán swirling flow

While the two first flows studied in the previous sections are interesting for their dynamics, that involve a limited number of modes, the turbulent von Kármán (vK) swirling flow is analysed here to apply the method on more complex field. Indeed, the vK flow, even when limited to a single azimuthal Fourier mode, requires a large number of POD modes to be described accurately, see Fig. 3c, due to the presence of small scale structures. The POD full analysis of the turbulent vK swirling flow has been conducted in a previous article [17]. In summary, a fluid contained in a cylinder is forced by counter-rotating disks fitted with blades, producing a turbulent swirling flow which is axisymmetric on average, with a dominant azimuthal Fourier mode $m_F = 0$. Here, we restrict our analysis to the next dominant Fourier mode, $m_F = 3$. This Fourier mode is crucial because it supports the Kelvin-Helmholtz instability, which arises from the shear layer located in the equatorial section of the cylinder geometry. The three-dimensional velocity field is described using the following Fourier representation:

$$\begin{aligned}\boldsymbol{u}(r,\theta,z,t) = \boldsymbol{u}^{c,0}(r,z,t) &+ \sum_{m_F=1}^{M-1} \boldsymbol{u}^{c,m_F}(r,z,t)\cos\left(m_F\theta\right) \\ &+ \sum_{m_F=1}^{M-1} \boldsymbol{u}^{s,m_F}(r,z,t)\sin\left(m_F\theta\right),\end{aligned} \tag{26}$$

from which we will select only the $m_F = 3$ contribution, hence the $n = 6$ velocity components $(u_r^{c,3}, u_r^{s,3}, u_\theta^{c,3}, u_\theta^{s,3}, u_z^{c,3}, u_z^{s,3})$.

One of the main results from [17] was that the resulting Kelvin-Helmoltz vortices (KHV) were found to undergo a steady solid body rotation at a specific Reynolds number of $Re = 29\ 000$. This motion is well described by the two first POD mode pair $(n_P = 1, n_P = 2)$ of the $m_F = 3$ Fourier mode, see Fig. 17. Other POD modes, e.g. $n_P = 3$, $n_P = 4$, further describe the shape of the KHV. From a global perspective, the $m_F = 3$ restricted von Kármán flow represents a more realistic and computationally involved case compared to the oscillating cylinder flow treated before, differing in three significant ways. First, this flow lacks a physically meaningful time average, as the dominant Fourier mode $m_F = 0$ is not included in the analysis. Second, the sampling timestep is relatively large compared to the forcing frequency. This choice ensures that the low-frequency rotation of the vortex structures is well-resolved but sacrifices the resolution of smaller time scales. This trade-off is necessary to manage computational costs while maintaining a reasonable number of snapshots, reflecting the practical constraints and complexity inherent in modelling such realistic flows.

In the following, we choose a small number of snapshots $N_t = 200$ on purpose, in order to match with the analysis of our previous paper, and also to make the parametric study of section 3.1 easier. The results are now described. Since the first POD mode pair dominates largely, see e.g. Fig. 3c, it is not a surprise that these structures are learnt first by both the AE (Fig. 18a) and the VAE (Fig. 18b). Nonetheless, the selected VAE only learns this first pair whereas the AE starts learning all the others after a reasonable training time. Our point here is not to claim that the VAE is less able or unable to learn. In fact, it is possible to build an accurate VAE from these data, see Fig. 3c, but we chose this illustrative case with respect to the latent space shape exhibited. In fact, while reducing β makes the VAE accuracy come close to the AE one, we observed that the VAE latent space gets fragmented and loses its orbital shape (Fig. 19b) just like the AE latent space (Fig. 19a). Because this fragmentation is meant to represent the higher temporal frequencies of the dynamic and because they are not resolved in the sampled snapshot set, these high frequencies can be considered as noise. In a way, the AE is fitting this noise, while the VAE does not, because of the regularization induced by a high enough β. This last point gives us a new way to interpret VAE beyond the accuracy metric: here we can interpret it as a filtering process which only retains the temporally resolved part of the dynamic. It is actually the case for this vK flow example where there is a time-scale separation between the dominant KHV and the rest of the flow.

In the AE case, we cannot identify clearly the latent manifold as two consecutive snapshots are encoded far apart from each other. This is to be understood again by the large sampling time that produces a sequence of snapshots with uncorrelated fluctuations. The VAE latent space follows a hollow ellipsoidal orbital shape. When one KH vortex is aligned with the x axis in the physical space, there is a maximum amplitude on POD mode $n_P = 0$, and when the same KH vortex is aligned with the y axis, $n_P = 1$ has maximum amplitude (the z-axis is then the cylin-

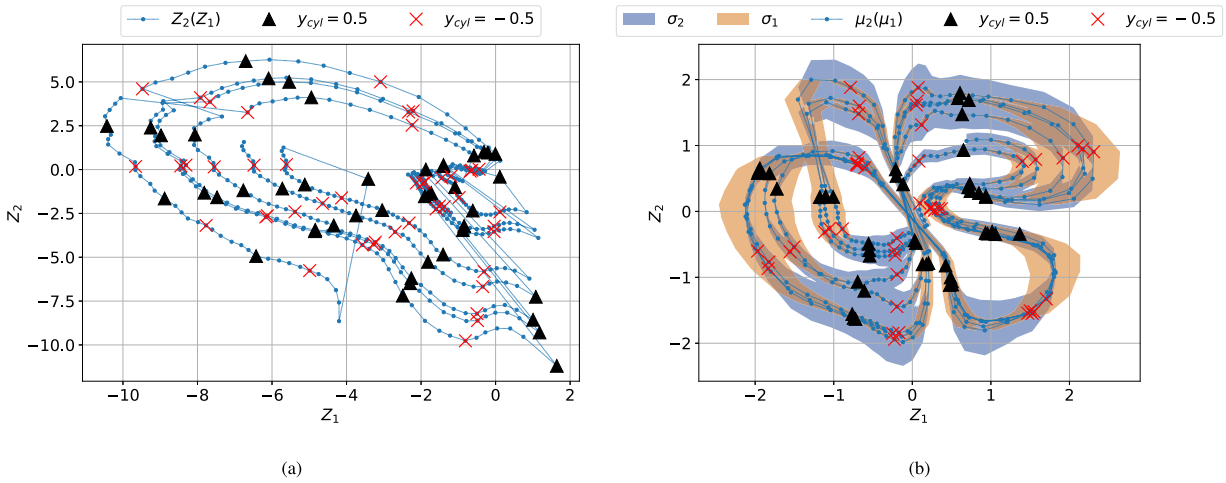


Fig. 12. 2D latent space visualization for the quasi-periodic flow case: (a) standard autoencoder, (b) variational autoencoder at $\beta = 10^{-3}$.

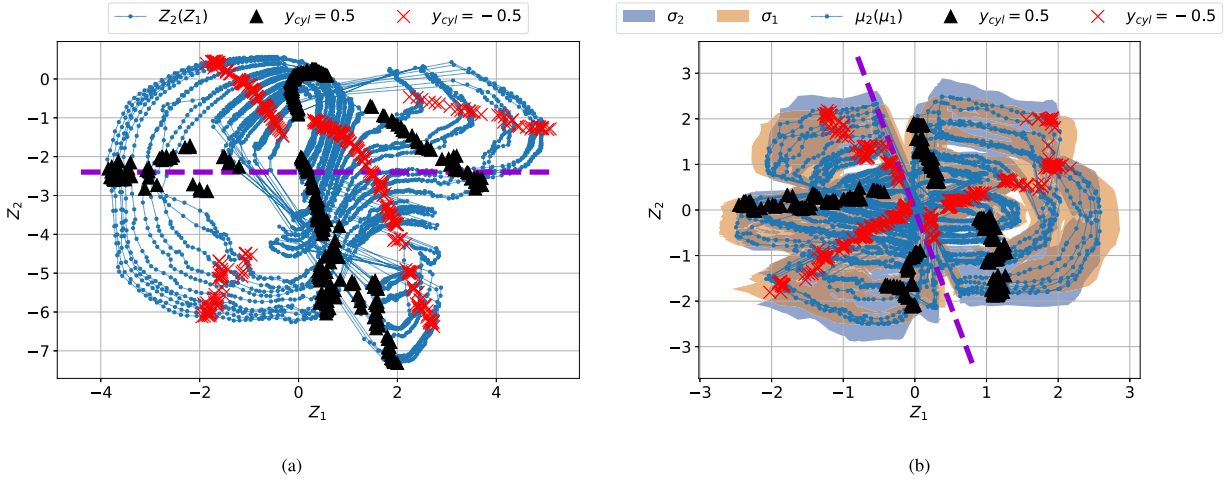


Fig. 13. Long run, 2D latent space visualization for the quasi-periodic flow case: (a) standard autoencoder, (b) variational autoencoder at $\beta = 10^{-3}$. The purple dashed line has been chosen in both cases to delimit 2 particular subspaces that can be interpreted using POD time coefficient distributions, see equations (24), (25), and Fig. 15a, 15b, 16a and 16b.

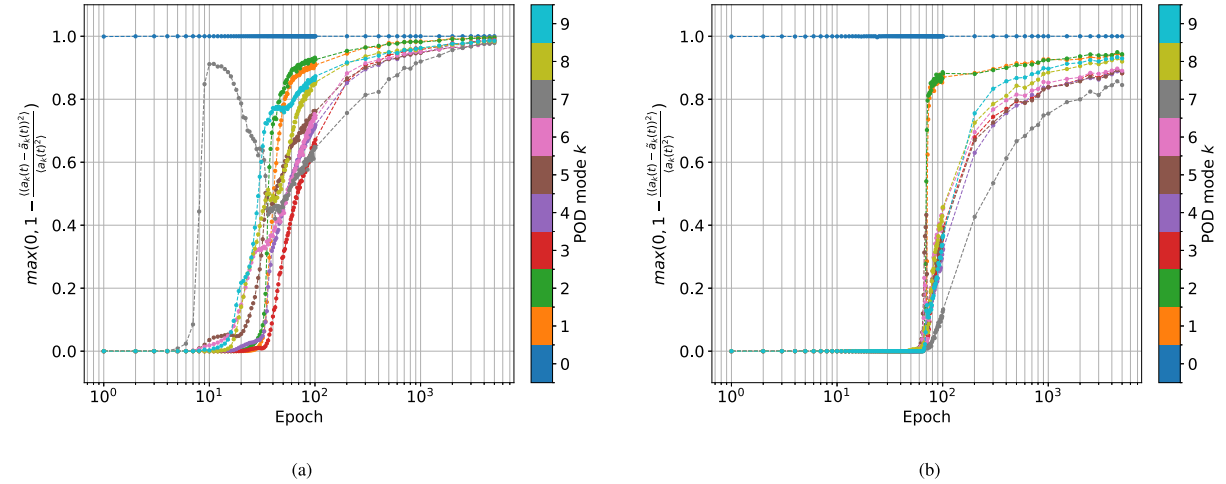


Fig. 14. Convergence of training per POD mode in the quasi-periodic flow case and d = 2 for models trained with $N_t = 15\,000$ snapshots: (a) standard autoencoder, (b) variational autoencoder at $\beta = 10^{-3}$.

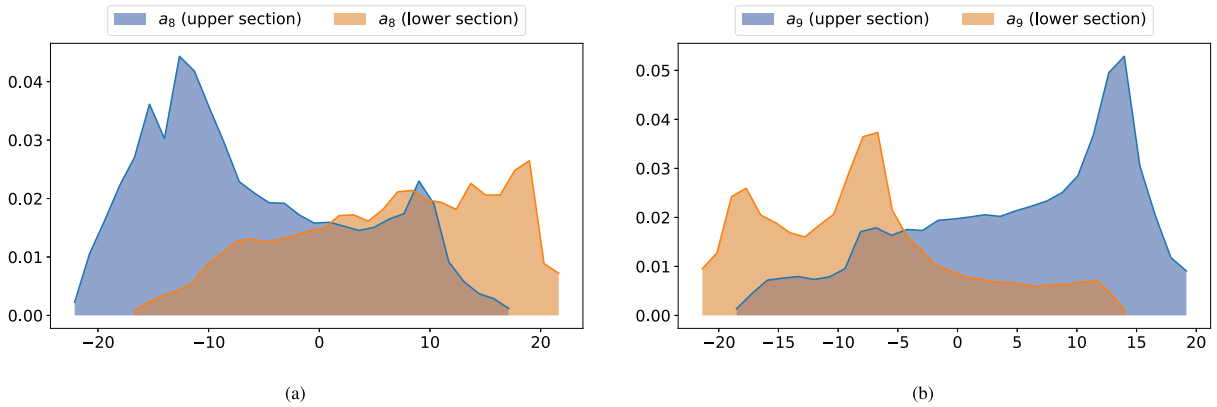


Fig. 15. POD coefficients probability distributions function, while selecting times in up and down subspaces of the AE latent space, as delimited on Fig. 13a.

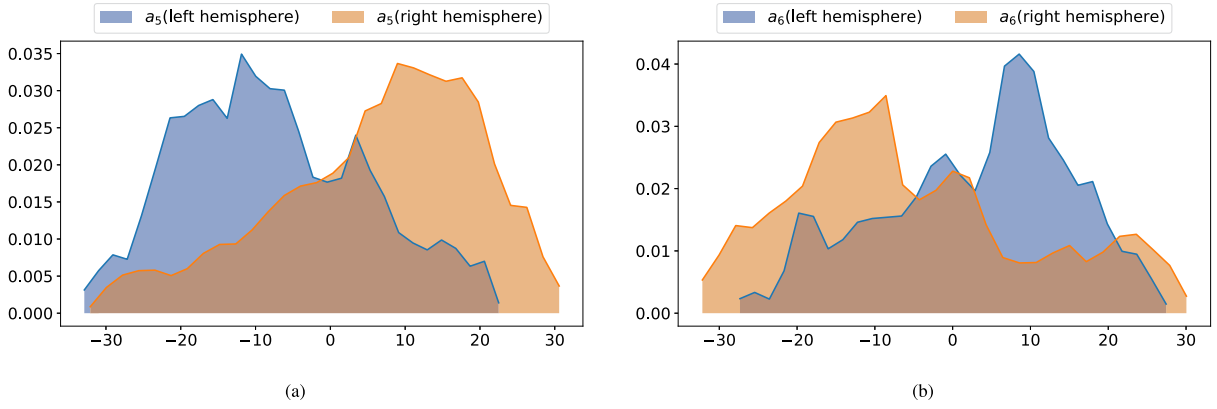


Fig. 16. POD coefficients probability distributions function, while selecting times in left and right hemisphere of the VAE latent space, as delimited on Fig. 13b.

der axis, such that $[\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z]$ forms a direct orthogonal frame). Then, relative amplitudes of $n_P = 0$ and $n_P = 1$ can express all different orientations of the KHV, as shown in [17]. These maxima are reported on the latent space figures to pin the space with this characteristic frequency. In this way, we recover a similar alignment property to the one that was found for the oscillating cylinder case, but with the KHV *response* frequency instead of the *forcing* frequency of the cylinder. Next, we split each latent space in i=1,2,3,4 regions $\{\Omega_i\}$ according to the simple $Z_k = 0$ line criterion, and we construct the sets S_{Ω_i}. To adopt a equivalent but complementary point of view to that given in the oscillating cylinder analysis, instead of directly analyzing the $A_k(\Omega_i)$ distributions, we take the following conditional mean to visualize the characteristic states directly in physical space:

$$\langle \boldsymbol{u} \rangle_{S_\Omega} = \frac{1}{|S_\Omega|} \sum_{\boldsymbol{v} \in S_\Omega} \boldsymbol{v}(\boldsymbol{x}, t). \tag{27}$$

This average is equivalent to the flow reconstructed using the average of $A_k(\Omega_i)$, as amplitude for each corresponding POD mode k. It follows from the linear nature of POD:

$$\langle \boldsymbol{u} \rangle = \frac{1}{N_t} \sum_t^{N_t} \boldsymbol{u}(\boldsymbol{x}, t) = \sum_k \left(\frac{1}{N_t} \sum_t^{N_t} a_k(t) \right) \boldsymbol{\phi}_k(\boldsymbol{x}), \tag{28}$$

that:

$$\langle \boldsymbol{u} \rangle_{S_\Omega} = \sum_k \left(\frac{1}{|A_k(\Omega_i)|} \sum_{a_k \in A_k(\Omega_i)} a_k \right) \boldsymbol{\phi}_k(\boldsymbol{x}). \tag{29}$$

The four resulting conditional averages $\langle \boldsymbol{u} \rangle_{S_\Omega}$ are shown for the AE case in Fig. 20, and for the VAE in Fig. 21. We clearly understand, from the AE and VAE latent spaces, that the VAE representation approximately aligns the KHV maxima on the Z_1 and Z_2 axes, so that the conditional average resembles the main POD mode pair (Fig. 17). This means that the VAE variables (Z_1, Z_2) encode roughly the same information as the POD coefficients associated with the KHV. In the AE case, the (Z_1, Z_2) variables encode more than the 2 first POD coefficients and modes, as shown in Fig. 18a. However, the global shape of the latent space is still ellipsoidal because the same low frequency is present, associated to the KHV change in orientation. We note also from Figs. 20 and 21, that the naive $Z_i > 0$, $Z_i < 0$ basic criterion is enough to demonstrate that the $\boldsymbol{Z} \longrightarrow -\boldsymbol{Z}$ symmetry is equivalent to a $(a_0, a_1) \longrightarrow (-a_0, -a_1)$ symmetry in POD space. In the physical space, it corresponds to the R_π symmetry (see [17] for more details). Once again, this symmetry is more valid in the case of VAE, at the cost of a loss of precision in the reconstruction, but it is also visible for AE. This further shows that autoencoders encode field symmetry using equivariant relations, i.e. mapping field symmetry to a set of reflection symmetries on the latent vector $\boldsymbol{Z}$. This property is also present for POD, in a data augmented by symmetry context [17].

4. Discussion

In this section, the results are summarized, and put into perspective in relation to literature.

4.1. Interpretation of latent spaces in term of trajectories

The first, and maybe the most important comment, is that autoencoders are interpretable. The underlying reason is that the encoder and decoder only act on the spatial part of the field, hence the encoder E expresses the physical system trajectory in phase space, $\Gamma_{phys} : t \longrightarrow u_{tx}$, in another set of coordinates $\Gamma_{lat} : t \longrightarrow Z_{tk}$. Considering a time window $[0, T_w]$:

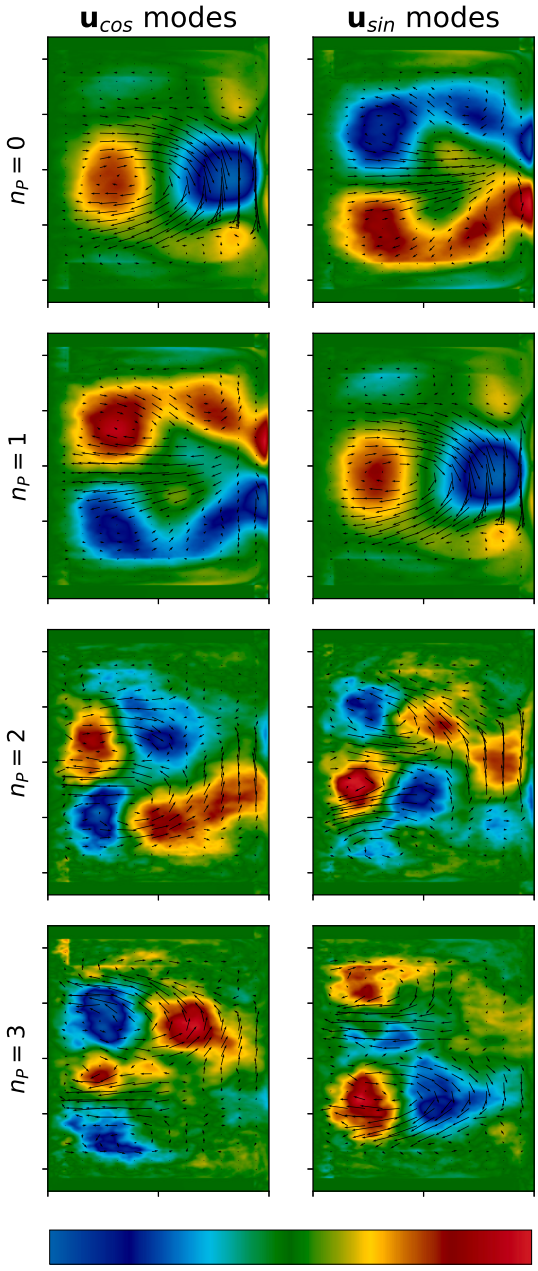


Fig. 17. The POD modes of the von Kármán swirling flow. Left and right flow components refer to $\boldsymbol{u}^{c,m_F=3}$ and $\boldsymbol{u}^{s,m_F=3}$, respectively, see equation (26). The u_θ component is shown with the colormap, while the u_r and u_z vector are drawn in plane.

$$\begin{array}{ccc} [0,T_w] & \xrightarrow{\Gamma_{phys}} & \mathbb{R}^N \\ \Gamma_{lat} \downarrow & \swarrow E & \\ \mathbb{R}^d & & \end{array}$$

such that:

$$\Gamma_{lat} = E \circ \Gamma_{phys}. \tag{30}$$

Interpretability comes from the fact that similar field snapshots are encoded into similar latent space location. In other words, the encoder preserves some notion of proximity. Consequently, the sub trajectories $\{\gamma_i \subset \Gamma \mid \gamma_i \cap \gamma_j = \varnothing,\ i \neq j\}$, that share similarities, will be encoded close from each other in the latent space. The quasi-periodic cylinder flow latent spaces show this point well, see Figs. 12, 13. For example, in Fig. 12a, three regions of the latent space can be highlighted:

$$\begin{gathered} \text{(i) close to zero,} \\ \text{(ii) second cadrant section with } Z_2 > -\frac{Z_1}{2}, \\ \text{(iii) third cadrant section with } Z_2 < -\frac{Z_1}{2}. \end{gathered}$$

The first idea one may have is to look at the selected sub trajectories to understand which characteristics of the flow are specific to each latent space subdomain Ω_i, labelling by hand the characteristics. By doing this laborious work, it is possible to label the subdomains, distinguishing two different far wake states: one with the deflected vortex located on the $y > 0$ half plane $\left|Wake_\uparrow\right\rangle$, or to the $y < 0$ half plane $\left|Wake_\downarrow\right\rangle$, together with two near wake states: again with emitted vortex up or down, $\left|Emis_\uparrow\right\rangle$ and $\left|Emis_\downarrow\right\rangle$. This is because the computational domain is large enough to accommodate two deflected vortices in the wake. We can then label: (i) $\left|Wake_\downarrow\right\rangle$ and $\left|Emis_\uparrow\right\rangle$; (ii) $\left|Wake_\uparrow\right\rangle$ and $\left|Emis_\downarrow\right\rangle$; and (iii) $\left|Wake_\downarrow\right\rangle$, $\left|Emis_\downarrow\right\rangle$. This example limited to $N_t = 500$ snapshots does not cover equally the phase space, whereas we start to see an equilibrium pattern emerging in the longer run as seen on Fig. 13a.

Then, we generalized this approach, defining $\{\Omega_i\}$ by generic criteria and characterizing the states using the $a_k(t)$ conditional distributions, summarized in equation (25). We proposed a protocol to generalize further to $d > 2$ by defining $\{\Omega_i\}$ with a clustering algorithm. We tested, to apply K-means clustering algorithm to latent space of Fig. 13b, and it gave back the generic criteria $Z_k > 0$ and $Z_k < 0$ as sub latent space border conditions. Further work is still needed to test and improve this approach, for example it might be interesting to leverage the latent space manifold geometry to define some distance between two sub trajectories, as a clustering criterion, connecting with the Proper Latent Decomposition (PLD) proposed in [14,15]. Meanwhile, we demonstrated the possibility to build an interpretation for both the standard autoencoder and for the variational autoencoder, despite the different latent spaces properties.

4.2. AE vs VAE latent space properties, equivariance relations, and implication for reduced order modelling

The formulation of AE and VAE is fundamentally different. The latent space of a classical autoencoder is defined point-wise in $\mathbb{R}^d$, at each snapshot coordinate $\boldsymbol{Z}(t)$, whereas the noise added in a variational autoencoder makes it in principle span all $\mathbb{R}^d$. The spaces constructed in AE can therefore be considered as discrete, constructed by a sum of Dirac functions, whereas VAE produce continuous spaces, constructed with a sum of Gaussians.

In the case of the periodic flow, we found that the VAE better preserves the space-time interval between consecutive points and, in relation with this property, better encodes the flow symmetries, see Figs. 9 and 10. The equivariance properties detailed in the analysis can be summarised by the diagram:

$$\begin{array}{ccc} \mathbb{R}^N & \xrightarrow{R_{phys}} & \mathbb{R}^N \\ E \downarrow & & \downarrow E \\ \mathbb{R}^d & \xrightarrow{R_{lat}} & \mathbb{R}^d \end{array}$$

where R_{phys} is a symmetry operator, acting on the physical field, and R_{lat} another symmetry operator, acting on the encoded field. However, the advantages of VAE are outweighed by the reduced precision of the reconstruction. We also demonstrated with the von Kármán swirling flow case the influence of the time discretization step of the trajectory Γ. While standard AE encodes the sub resolved modes (as does POD),

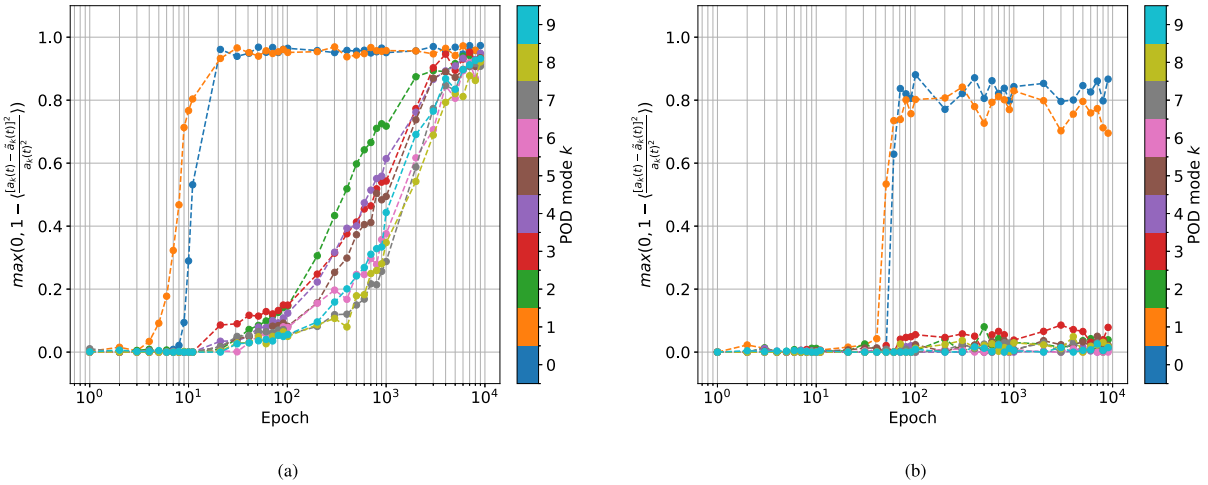


Fig. 18. Convergence of training per POD mode in the von Kármán swirling flow case, d = 2 for (a) standard autoencoder, (b) variational autoencoder at $\beta = 10^{-3}$.

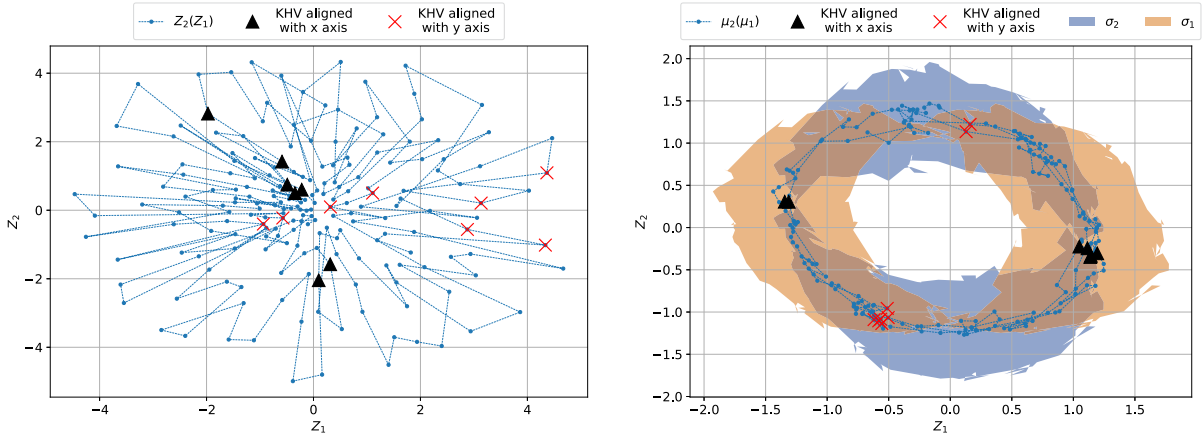


Fig. 19. 2D latent space visualization for the Kármán swirling flow case: (a) standard autoencoder, (b) variational autoencoder at $\beta = 10^{-3}$.

VAE does not, highlighting the links between encoding symmetries, regularization and generalization, see Figs. 18 and 19. Other approaches, in the literature, choose to erase the equivariance properties of the latent space, and treat the various field symmetries explicitly to compress the latent space further [35]. Here, we used the equivariance properties to keep the symmetry induced features in the latent space, for interpretation purpose. In addition, by denoting $\boldsymbol{J}$ the jacobian of the decoder $\boldsymbol{J} = \partial_{\boldsymbol{z}} D(\boldsymbol{z})$, it is possible to show (see appendix A.4) that:

$$\boldsymbol{J}\, R_{lat} = R_{phys}\, \boldsymbol{J}, \tag{31}$$

giving a general foundation of the equivarient property for any invertible mapping $E^{-1} = D$, such that $\boldsymbol{u} = D \circ E(\boldsymbol{u})$.

Another conclusion must be drawn from the von Kármán swirling flow case: the idea of adding a random noise in the latent space of the VAE tends to mix together high frequencies present in the data. Those are clearly encoded in the standard AE (see Fig. 19a), with the high frequencies of the prior random noise. This mixture prevents the VAE from reconstructing the small scales associated with high temporal frequencies, as they have been blurred by the noise. The reason behind it is that the small scales carry less energy than the large scales, or at least the energy is distributed intermittently on small spatial structures, such that the white noise is acting more on small scales than on the large scales, relatively speaking. This suggests that noise-addition methods destroy some of the information contained in the data, but in return allow the model to act as a filter for unstructured, decorrelated fluctuations. Consequently, in a Reduced Order Model (ROM) perspective, AEs are more suited to a deterministic modelling of time evolution, whereas VAEs are by design suited for probabilistic modelling. This idea has already infused in the community, considering that the aforementioned deterministic approaches use classical autoencoders [3,4,8], and probabilistic approaches use variational autoencoders [5]. Indeed, an Initial Value Problem (IVP) will not be well posed in a VAE built space because one given snapshot will have infinitely many images by the VAE encoder, whereas the same IVP will be well posed in a AE built space. A solution to this problem is to formulate the time evolution only on the mean variable of the VAE $\boldsymbol{\mu}(t)$ to recover a well posed problem.

Finally, speaking about ROMs using an AE latent space, it is important to mention another very promising approach, which is based on manifold Galerkin and least squares Petrov-Galerkin projection [36,37], where the lack of explanability of deep learning methods is balanced by guaranties on the error of the model. Because this methodology requires continuously differentiable manifolds, it might suffer from AE's tendency to encode high temporal frequencies that are not properly resolved in the training data. Meaning that for turbulent flow, the cost of running the full order model and storing snapshots with a sufficiently high sampling frequency, then training an autoencoder on this big dataset can make the computing price increase dramatically. In this context, building a ROM on a VAE mean $\boldsymbol{\mu}(t)$ should mitigate the issue, if picking a β that satisfies accuracy constraints, because the trajectory of the encoded system will be smoother.

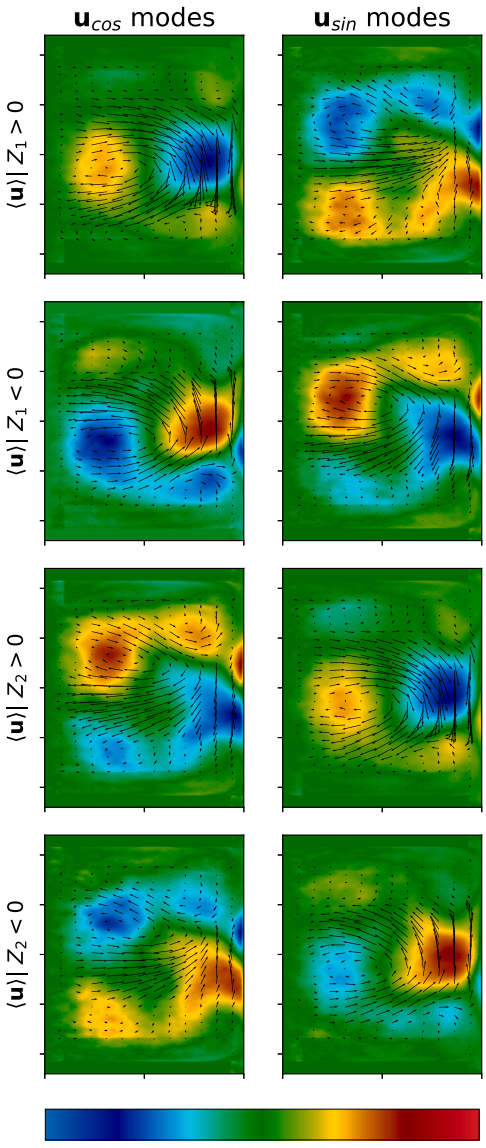


Fig. 20. Mode extraction by conditional averaging, for the standard autoencoder.

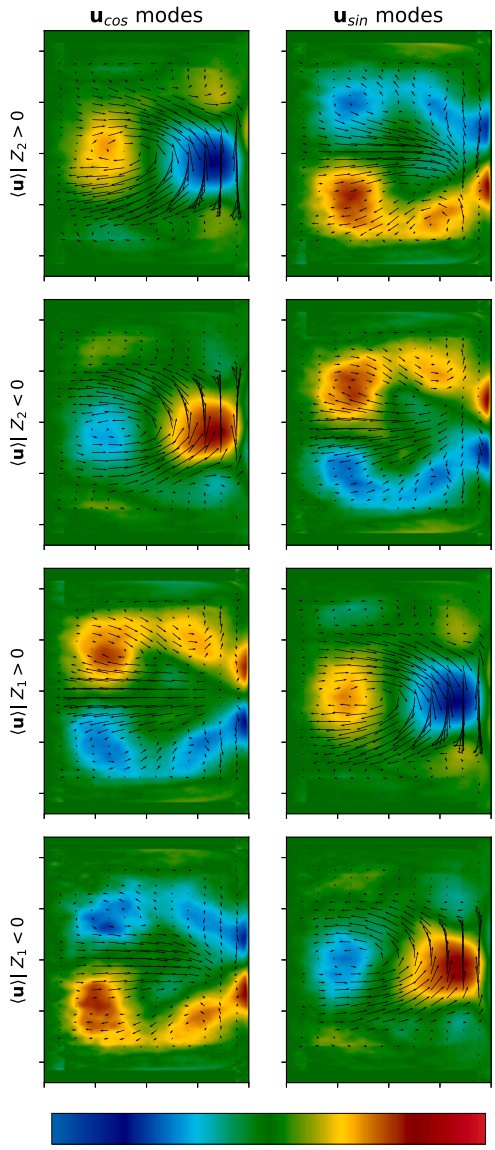


Fig. 21. Mode extraction by conditional averaging, for the variational autoencoder.

4.3. Related work

In regard to previous work, the present comparisons between AE and VAE are consistent with those of [12]. We use the same E_d metric, cf. equation (14), extending the analysis to a wider range of β values, to highlight the existence of two types of E_d saturation processes:

(i) E_d is converging to a fixed value while increasing the number of latent space dimension d,
(ii) The plateau value is converging towards 1, i.e. 100% reconstruction accuracy, when the number of filters N_{bf} is increased.

This behaviour shows the relevance of decoupling the latent space global representation (MLP), from the construction of many local filtering and transformations of the field (CNN). It is interesting because it shows that increasing the number of filters N_{bf} increases reconstruction accuracy, at fixed latent space dimension d. Furthermore, we also extended the E_d criterion to a *per POD mode* e_k, see equation (23), that allows to visualize which structures of the flow are present in the model reconstruction in a more compact and exhaustive way. It is today a common method to apply POD on the reconstructed field, in order to qualify the results, for example see [2,4,10]. However, the applied methodology, which consists of comparing the POD modes from the data and from the reconstructed field, can be cumbersome because one must look and compare images of the modes qualitatively. The analysis of e_k temporal coefficient errors is more quantitative and also allows us to qualify the same idea while the network is being trained (see e.g. Fig. 14). So, by making this criterion more compact, we have extended the idea to obtain more information about what the network is learning.

Next, we found that the POD time coefficients $a_k(t)$ can be used to interpret any autoencoder's latent spaces, by analysing their distributions over sub latent spaces (Fig. 16). This idea enables us to characterize precisely which flow features are encoded in which latent directions, or in local subspaces. We insist on the fact that the presented method generalizes the *sample linearly the latent space and decode* method, that was originally present in the machine learning and computer vision community, see e.g. [11,20], and that was adapted to fluid mechanics analysis, e.g. in [12]. The main improvement lies in the fact that disentangling latent space is not a prerequisite for interpretation. Instead of constraining the latent space shape, enforcing some disentanglement e.g. through the VAE objective [12], we argue that we can equivalently gain insights by

choosing relevant sub latent spaces $\{\Omega_i\}$. Moreover, the snapshot sets constructed from $\{\Omega_i\}$ do not have to be characterized with POD, they can also be analysed by other means, for example with wavelet analysis.

It is important to mention that recent work [38] has some link with the presented analysis. They used a criterion based on $\tilde{a}_k(t)$, eq. (22), to characterize how the different POD modes are sensible to the variation of the different latent space axes:

$$\epsilon_{ij} = \frac{1}{\int d\boldsymbol{Z}} \int \left| \frac{\partial \tilde{a}_j}{\partial z_i} \right| d\boldsymbol{Z}. \tag{32}$$

In particular, they showed plots of $\tilde{a}_j$ versus Z_i, on which some distribution split equivalent to our Fig. 16 can be recovered by eyes. However, they were analyzing the derivative $\frac{\partial \tilde{a}_j}{\partial z_i}$ which turns out to have minimum amplitude at maxima of $\tilde{a}_k$ distributions. The connection between the two approaches is that $\frac{\partial \tilde{a}}{\partial z} = 0$ values might be associated to some stable or metastable state, whereas $\max \left| \frac{\partial \tilde{a}}{\partial z} \right|$ may be a good criterion for the Ω_i boundaries. We state that the present work is meant to be a new approach from [12] and appears to be parallel, yet different from [38]. From a more global perspective, it is a good sign that we converged independently to a common methodology, as it underlines the robustness of the autoencoder framework.

5. Conclusion

This work addresses the dual challenges of learning and interpretability in AEs and VAEs applied to the reconstruction of unsteady velocity fields. Focusing on periodically forced flows, chosen for their clearer physical structure, we analyze how latent spaces encode flow features and how this impacts model usability. We conducted a systematic parametric study to identify the most efficient models and found, consistent with prior work, that increasing β in VAEs degrades reconstruction accuracy, highlighting the limitations of the standard Gaussian prior. The learning dynamics also revealed a sensitivity to initialization, exposing a transition between non-learning and structure recovering regimes.

By analyzing flows in canonical configurations, we used POD to reveal which physical structures are captured by autoencoders. Crucially, symmetry and frequency content in the latent space provided a direct link between learned variables and physical phenomena. This interpretability is not just an added benefit, it is essential. Understanding how autoencoders encode flow features enables us to diagnose, trust, and improve these models.

We proposed a preliminary methodology to characterize latent states via high-activation POD coefficients, bridging data-driven clustering and fluid mechanics. While still evolving, this approach offers a path toward physically meaningful, low-dimensional representations of complex dynamics, underscoring the necessity of interpretable latent geometries in deep learning for fluid flows.

CRediT authorship contribution statement

Rémi Bousquet: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization. **Caroline Nore:** Writing – review & editing, Validation, Supervision, Project administration, Methodology. **Didier Lucor:** Writing – review & editing, Validation, Project administration, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

Acknowledgements

We acknowledge Yann Fraigneau for his help using the SUNFLUIDH code. This work was performed using HPC resources from GENCI-IDRIS (Grant 2024-0254).

Appendix A

A.1. AdamW implementation

The AdamW routine main ingredients are detailed below:

$$\begin{aligned}
\mathcal{L} &= \frac{1}{N_t \cdot N} \sum_{t,x} (u_{tx} - \tilde{u}_{tx})^2, \\
\boldsymbol{m}_i &= \beta_1 \boldsymbol{m}_{i-1} + (1-\beta_1) \nabla_{\boldsymbol{\Theta}} \mathcal{L}, \\
\boldsymbol{s}_i &= \beta_2 \boldsymbol{s}_{i-1} + (1-\beta_2) (\nabla_{\boldsymbol{\Theta}} \mathcal{L})^2, \\
\hat{\boldsymbol{m}}_i &= \frac{\boldsymbol{m}_i}{1-(\beta_1)^i}, \\
\hat{\boldsymbol{s}}_i &= \frac{\boldsymbol{s}_i}{1-(\beta_2)^i}, \\
\boldsymbol{\Theta}_{i+1} &= \boldsymbol{\Theta}_i - \eta \left(\frac{\hat{\boldsymbol{m}}_i}{\sqrt{\hat{\boldsymbol{s}}_i} + \epsilon} + \lambda \boldsymbol{\Theta}_{i-1} \right),
\end{aligned} \tag{33}$$

where all the operations are to be understood element-wise. Parameters β_1 and β_2 are of order one and control the momentum of the gradient descent, whereas λ, η and ϵ are of order zero and control the rate of weight decay, the size of the step in the parameter space (called the learning rate), and ensure positive denominator, respectively.

A.2. Software performance details

All the gathered models were trained on a single GPU V100 or A6000 indifferently. Consequently, the standard deviation of elapsed time can be large, suggesting that the code performances depend strongly on the hardware and production management resources. Each run corresponds to one training session for different models, for which we have recorded the elapsed wall time each time. See Tables 1 and 2.

A.3. K-mean clustering

We give here some comparison between our partitioning method and another partitioning obtained with the K-mean criterion. We take the example of section 3.2.3 to show that K-mean in fact finds a partition that is very close to the intuitive criterion chosen, see Fig. 22.

A.4. Invertible mapping and equivariance relations

In what follows, we try to develop a general understanding of how symmetry is encoded in any reversible coordinate change (like an autoencoder). We first give ourselves two linear operators R_{phys} and R_{lat}, acting on $\mathbb{R}^N$ and $\mathbb{R}^d$, respectively, and also an arbitrary invertible transformation (an autoencoder) $E : \mathbb{R}^N \longrightarrow \mathbb{R}^d$, $E^{-1} = D : \mathbb{R}^d \longrightarrow \mathbb{R}^N$. We find easily in this case, that:

$$\partial_t \boldsymbol{u} = \partial_t D(\boldsymbol{z}) = \frac{\partial D(\boldsymbol{z})}{\partial \boldsymbol{z}} \partial_t \boldsymbol{z} = \boldsymbol{J} \partial_t \boldsymbol{z}, \tag{34}$$

where $\boldsymbol{J} = \frac{\partial D(\boldsymbol{z})}{\partial \boldsymbol{z}}$ is the jacobian of the decoder. Now we look for a linear map R_{lat} from $\boldsymbol{z}$ to $\boldsymbol{z}' = E(\boldsymbol{u}')$, considering that $\boldsymbol{u}' = R_{phys}\boldsymbol{u}$. This linear mapping takes the form:

$$\boldsymbol{z}' = R_{lat} \boldsymbol{z}. \tag{35}$$

This implies in general, using equation (34), that:

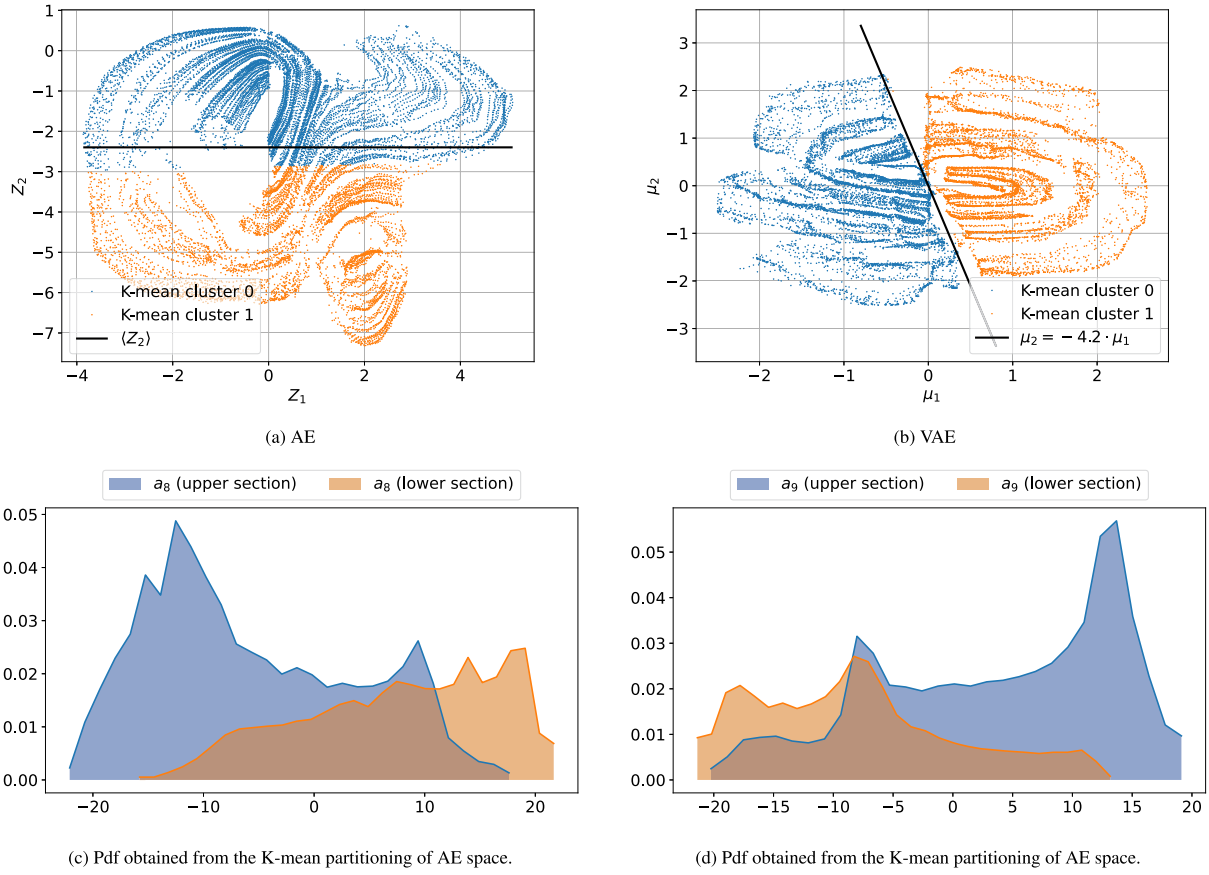
$$\boldsymbol{u}' - R_{phys} \boldsymbol{u} = 0$$

Table 1
Average and standard deviation of elapsed (wall) time per epoch, with respect to the number of convolutional filters N_{bf} together with the rounded total number of parameters of the network.

Number of runs	N_{bf} (Number of filters)	Total number of parameters	Elasped time per epoch (s) (mean ± std)
150	112	1.5×10^5	0.535 ± 0.268
63	224	4×10^5	1.618 ± 1.89
55	896	3.5×10^6	3.249 ± 2.313

Table 2
Average and standard deviation of elapsed (wall) time per epoch, with respect to the number of forward pass per epoch.

Number of runs	N_t /Batchsize (Number of forward pass per epoch)	Elasped time per epoch (s) (mean ± std)
76	6	0.592 ± 0.352
155	16	1.263 ± 1.309
21	117	4.71 ± 5.613
13	206	6.682 ± 2.222



(a) AE

(b) VAE

(c) Pdf obtained from the K-mean partitioning of AE space.

(d) Pdf obtained from the K-mean partitioning of AE space.

Fig. 22. (a) and (b): Comparison between the *ad hoc* criterion, the black line, and the K-mean criterion, represented by the blue and orange clusters. (c) and (d): distributions of a_8 and a_9, computed from the K-mean partitioning criterion, for comparison with Fig. 15.

$$\partial_t \boldsymbol{u}' - \partial_t R_{phys} \boldsymbol{u} = 0$$
$$\partial_t \boldsymbol{u}' - R_{phys} \partial_t \boldsymbol{u} = 0$$
$$\boldsymbol{J} \partial_t \boldsymbol{z}' - R_{phys} \boldsymbol{J} \partial_t \boldsymbol{z} = 0$$
$$\boldsymbol{J} \partial_t R_{lat} \boldsymbol{z} - R_{phys} \boldsymbol{J} \partial_t \boldsymbol{z} = 0$$
$$(\boldsymbol{J} R_{lat} - R_{phys} \boldsymbol{J}) \partial_t \boldsymbol{z} = 0,$$

where we also used the hypothesis that $\partial_t R_{lat} = R_{lat} \partial_t$, and $\partial_t R_{phys} = R_{phys} \partial_t$, meaning that the mappings do not depend on time. Then, the relationship between R_{phys} and R_{lat} is a general change of basis:

$$\boldsymbol{J} R_{lat} = R_{phys} \boldsymbol{J}. \tag{36}$$

Now, if we consider that R_{phys} is a symmetry operator, we can write that any $\boldsymbol{u}$ (solution of the underlying PDEs) is an eigenvector of R_{phys}, such that:

$$R_{phys} \boldsymbol{u} = \lambda \boldsymbol{u}, \tag{37}$$

with $\lambda \pm 1$, and we can compute the consequences on R_{lat}, using equations (34), (36) and (37):

$$\boldsymbol{J} \partial_t \boldsymbol{z}' = \partial_t \boldsymbol{u}'$$
$$\boldsymbol{J} \partial_t R_{lat} \boldsymbol{z} = \partial_t R_{phys} \boldsymbol{u}$$
$$= \partial_t \lambda \boldsymbol{u}$$

$$= \lambda \partial_t \boldsymbol{u}$$

$$= \lambda \boldsymbol{J} \partial_t \boldsymbol{z}$$

$$\boldsymbol{J} \partial_t R_{lat} \boldsymbol{z} = \boldsymbol{J} \partial_t \lambda \boldsymbol{z}. \tag{38}$$

Furthermore, $\boldsymbol{J}^{-1}$ exists in the case of VAE, because the orthogonality of $\boldsymbol{J}$ columns [39] implies that $\boldsymbol{J}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{J}^T$, with $\boldsymbol{A}$ a diagonal matrix whose elements are square L2 norm of $\boldsymbol{J}$ columns. In this case, when applying $\boldsymbol{J}^{-1}$ on equation (38) and integrating over time, we find:

$$R_{lat} \boldsymbol{z} = \lambda \boldsymbol{z} + C \tag{39}$$

for some constant C, that only induces a shift. These calculations show that the equivariance property $\boldsymbol{z}' = E(R_{phys}\boldsymbol{u}) = R_{lat}E(\boldsymbol{u}) = R_{lat}\boldsymbol{z}$ exists in general, in the form of equation (36), and that, when R_{phys} is a symmetry operator, then R_{lat} is also a symmetry operator. At the end, it is not a property of VAE but it is more generally associated to invertible mappings.

Appendix B. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cpc.2025.109728.

Data availability

Data will be made available on request.

References

[1] M. Milano, P. Koumoutsakos, Neural network modeling for near wall turbulent flow, J. Comput. Phys. 182 (1) (2002) 1–26, https://doi.org/10.1006/jcph.2002.7146.

[2] T. Murata, K. Fukami, K. Fukagata, Nonlinear mode decomposition with convolutional neural networks for fluid dynamics, J. Fluid Mech. 882 (2020) A13, https://doi.org/10.1017/jfm.2019.822.

[3] E. Menier, M.A. Bucci, M. Yagoubi, L. Mathelin, M. Schoenauer, CD-ROM: complemented deep - reduced order model, Comput. Methods Appl. Mech. Eng. 410 (115985) (2023) 115985.

[4] A. Racca, N.A.K. Doan, L. Magri, Predicting turbulent dynamics with the convolutional autoencoder echo state network, J. Fluid Mech. 975 (Nov. 2023), https://doi.org/10.1017/jfm.2023.716.

[5] A. Solera-Rico, C. Sanmiguel Vila, M. Gómez-López, Y. Wang, A. Almashjary, S.T.M. Dawson, R. Vinuesa, Beta-variational autoencoders and transformers for reduced-order modelling of fluid flows, Nat. Commun. 15 (1) (Feb. 2024), https://doi.org/10.1038/s41467-024-45578-4.

[6] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 113 (15) (2016) 3932–3937, https://doi.org/10.1073/pnas.1517384113.

[7] K. Champion, B. Lusch, J.N. Kutz, S.L. Brunton, Data-driven discovery of coordinates and governing equations, Proc. Natl. Acad. Sci. 116 (45) (2019) 22445–22451, https://doi.org/10.1073/pnas.1906995116.

[8] E. Menier, S. Kaltenbach, M. Yagoubi, M. Schoenauer, P. Koumoutsakos, Interpretable learning of effective dynamics for multiscale systems, arXiv:2309.05812, 2023.

[9] P. Gupta, P.J. Schmid, D. Sipp, T. Sayadi, G. Rigas, Mori-Zwanzig latent space Koopman closure for nonlinear autoencoder, https://doi.org/10.48550/ARXIV.2310.10745, https://arxiv.org/abs/2310.10745, 2023.

[10] K. Fukami, T. Nakamura, K. Fukagata, Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data, Phys. Fluids (1994) 32 (9) (2020) 095110.

[11] C.-H. Pham, S. Ladjal, A. Newson, PCA-AE: Principal component analysis autoencoder for organising the latent space of generative networks, J. Math. Imaging Vis. 64 (5) (2022) 569–585, https://doi.org/10.1007/s10851-022-01077-z, https://hal.science/hal-03713275.

[12] H. Eivazi, S. Le Clainche, S. Hoyas, R. Vinuesa, Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows, Expert Syst. Appl. 202 (117038) (2022) 117038.

[13] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828, https://doi.org/10.1109/TPAMI.2013.50.

[14] L. Magri, A.K. Doan, On interpretability and proper latent decomposition of autoencoders (2022), https://doi.org/10.48550/ARXIV.2211.08345, https://arxiv.org/abs/2211.08345.

[15] D. Kelshaw, L. Magri, Proper latent decomposition, https://doi.org/10.48550/ARXIV.2412.00785, https://arxiv.org/abs/2412.00785, 2024.

[16] P. Holmes, J.L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge Monographs on Mechanics, Cambridge University Press, 1996.

[17] R. Bousquet, O. Chaffard, M. Creff, D. Lucor, C. Nore, Large scale analysis of three-dimensional turbulent von Kármán swirling flows, Phys. Fluids (1994) 63 (10) (Oct. 2024).

[18] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, https://doi.org/10.48550/ARXIV.1412.6980, https://arxiv.org/abs/1412.6980, 2014.

[19] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, https://doi.org/10.48550/ARXIV.1711.05101, https://arxiv.org/abs/1711.05101, 2017.

[20] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, https://doi.org/10.48550/ARXIV.1312.6114, https://arxiv.org/abs/1312.6114, 2013.

[21] C. Jacobsen, K. Duraisamy, Disentangling generative factors of physical fields using variational autoencoders, Front. Phys. 10 (2022) 890910.

[22] S. Odaibo, Tutorial: deriving the standard variational autoencoder (VAE) loss function, https://doi.org/10.48550/ARXIV.1907.08956, https://arxiv.org/abs/1907.08956, 2019.

[23] A. Faugaret, Y. Duguet, Y. Fraigneau, L. Martin Witkowski, A simple model for arbitrary pollution effects on rotating free-surface flows, J. Fluid Mech. 935 (2022) A2, https://doi.org/10.1017/jfm.2021.980.

[24] J.-L. Guermond, R. Laguerre, J. Léorat, C. Nore, Nonlinear magnetohydrodynamics in axisymmetric heterogeneous domains using a Fourier/finite element technique and an interior penalty method, J. Comput. Phys. 228 (8) (2009) 2739–2757.

[25] J.S. Leontini, M.C. Thompson, K. Hourigan, Three-dimensional transition in the wake of a transversely oscillating cylinder, J. Fluid Mech. 577 (2007) 79–104, https://doi.org/10.1017/S0022112006004320.

[26] Y. Fraigneau, DATABASE-FLMEC: a CFD database on unsteady and turbulent flows, https://doi.org/10.57745/IE8RPF, 2024.

[27] R. Pasquetti, R. Bwemba, L. Cousin, A pseudo-penalization method for high Reynolds number unsteady flows, Appl. Numer. Math. 58 (7) (2008) 946–954.

[28] D. Lucor, M. Triantafyllou, Parametric study of a two degree-of-freedom cylinder subject to vortex-induced vibrations, J. Fluids Struct. 24 (8) (2008) 1284–1293.

[29] A. Placzek, J.-F. Sigrist, A. Hamdouni, Numerical simulation of an oscillating cylinder in a cross-flow at low Reynolds number: forced and free oscillations, Comput. Fluids 38 (1) (2009) 80–100.

[30] P. Anh-Hung, L. Chang-Yeol, S. Jang-Hoon, C. Ho-Hwan, K. Hee-Jung, H.-S. Yoon, D.-W. Park, I.-R. Park, Laminar flow past an oscillating circular cylinder in cross flow, J. Mar. Sci. Technol. 18 (3) (Jun. 2010), https://doi.org/10.51400/2709-6998.1881.

[31] C. Williamson, A. Roshko, Vortex formation in the wake of an oscillating cylinder, J. Fluids Struct. 2 (4) (1988) 355–381, https://doi.org/10.1016/s0889-9746(88)90058-8.

[32] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University, Of California Press, 1967.

[33] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (86) (2008) 2579–2605, http://jmlr.org/papers/v9/vandermaaten08a.html.

[34] L. McInnes, J. Healy, J. Melville, Umap: uniform manifold approximation and projection for dimension reduction, arXiv:1802.03426, https://arxiv.org/abs/1802.03426, 2020.

[35] S. Kneer, T. Sayadi, D. Sipp, P. Schmid, G. Rigas, Symmetry-Aware Autoencoders: s-PCA and s-nlPCA, working paper or preprint, Nov. 2021, https://hal.science/hal-03420320.

[36] K. Lee, K.T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, J. Comput. Phys. 404 (2020) 108973, https://doi.org/10.1016/j.jcp.2019.108973, https://www.sciencedirect.com/science/article/pii/S0021999119306783.

[37] F. Romor, G. Stabile, G. Rozza, Non-linear manifold reduced-order models with convolutional autoencoders and reduced over-collocation method, J. Sci. Comput. 94 (3) (Feb. 2023), https://doi.org/10.1007/s10915-023-02128-2.

[38] Y. Mo, T. Traverso, L. Magri, Decoder decomposition for the analysis of the latent space of nonlinear autoencoders with wind-tunnel experimental data, https://doi.org/10.48550/ARXIV.2404.19660, https://arxiv.org/abs/2404.19660, 2024.

[39] M. Rolinek, D. Zietlow, G. Martius, Variational autoencoders pursue PCA directions (by accident), https://doi.org/10.48550/ARXIV.1812.06775, https://arxiv.org/abs/1812.06775, 2018.