

Data-Targeted Prior Distribution for Variational AutoEncoder

Nissrine Akkari ^{1,*}, Fabien Casenave ¹, Thomas Daniel ¹ and David Ryckelynck ²

¹ Safran Tech, Digital Sciences and Technologies Department, Rue des Jeunes Bois, Châteaufort, 78114 Magny-Les-Hameaux, France; fabien.casenave@safrangroup.com (F.C.); thomas.daniel@safrangroup.com (T.D.)

² Centre des Matériaux (CMAT), CNRS UMR 7633, Mines ParisTech, PSL University, BP 87, 91003 Evry, France; david.ryckelynck@mines-paristech.fr

* Correspondence: nissrine.akkari@safrangroup.com

Abstract: Bayesian methods were studied in this paper using deep neural networks. We are interested in variational autoencoders, where an encoder approaches the true posterior and the decoder approaches the direct probability. Specifically, we applied these autoencoders for unsteady and compressible fluid flows in aircraft engines. We used inferential methods to compute a sharp approximation of the posterior probability of these parameters with the transient dynamics of the training velocity fields and to generate plausible velocity fields. An important application is the initialization of transient numerical simulations of unsteady fluid flows and large eddy simulations in fluid dynamics. It is known by the Bayes theorem that the choice of the prior distribution is very important for the computation of the posterior probability, proportional to the product of likelihood with the prior probability. Hence, we propose a new inference model based on a new prior defined by the density estimate with the realizations of the kernel proper orthogonal decomposition coefficients of the available training data. We numerically show that this inference model improves the results obtained with the usual standard normal prior distribution. This inference model was constructed using a new algorithm improving the convergence of the parametric optimization of the encoder probability distribution that approaches the posterior. This latter probability distribution is data-targeted, similarly to the prior distribution. This new generative approach can also be seen as an improvement of the kernel proper orthogonal decomposition method, for which we do not usually have a robust technique for expressing the pre-image in the input physical space of the stochastic reduced field in the feature high-dimensional space with a kernel inner product.

Keywords: kernel proper orthogonal decomposition (KPOD); variational autoencoder (VAE); inference model; prior distribution; Gaussian kernel; unsteady and compressible fluid flows



Citation: Akkari, N.; Casenave, F.; Daniel, T.; Ryckelynck, D. Data-Targeted Prior Distribution for Variational AutoEncoders. *Fluids* **2021**, *6*, 343. <https://doi.org/10.3390/fluids6100343>

Academic Editor: Giovanni Stabile

Received: 29 June 2021

Accepted: 21 September 2021

Published: 29 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Numerical simulation in the domain of fluid dynamics remains a difficult task, especially with respect to turbulent fluid flows for which the large eddy simulations are crucial for modeling the details of all the different spatial scales. Thanks to the precision that these techniques offer in the field of fluid dynamics with respect to other numerical methods for the resolution of Navier–Stokes equations, they are used in the aeronautical industry for helping the design conception of the different components of an aeronautical engine. Typically, these simulations are used for solving the reacting Navier–Stokes equations in order to predict the behavior of a turbulent flow in an aeronautical injection system and a combustion chamber [1,2]. However, the CPU cost of large eddy simulations is still prohibitive despite all the progress in high-performance computing on massively parallel super computers. This cost is essentially related to the need for performing multiple runs in parametric and geometric studies involved in optimization loops.

In the state of the art, we find different methods offering a solution to this problem: model order reduction techniques, metamodeling and deep learning approaches.

POD-based model order reduction offers a large panorama of solutions thanks to the projection of the model's equations upon a reduced order vector space obtained by POD method [3,4]. The capacity of the POD–Galerkin approach to adapt to highly non-linear convective equations, such as the incompressible Navier–Stokes equations, thanks to stabilization approaches [5–8], made of this approach a good candidate for the construction of parametric and geometric reduced order models for the incompressible Navier–Stokes equations, as can be seen in [9,10].

Metamodeling and kriging approaches [11,12] can be used as alternatives to projection-based model order reduction approaches, as these offer a completely non-intrusive framework for computing the solution in the reduced order vector space. The projection coefficients are obtained by solving a regression problem with respect to the real orthogonal coefficients from the available offline data, as can be seen in [13–15].

Recently, the use of deep learning approaches for approximating solutions of fluid dynamics simulations has undergone an important development. We can cite [16], where VAEs [17,18] are used for computing missing data in fluid dynamics fields. This application has been compared to the gappy POD technique, as can be seen in [19,20]. We cite the work in [21], where a multiple temporal paths convolutional neural network (MTPC) was used to predict features in different time ranges for a turbulent flow, in order to perform a super-resolution reconstruction of these flows from low fidelity spatial fields. In [22], convolutional neural networks are used to learn the mapping from a coarse grid to the closure term of a large eddy simulation.

In this paper, we propose a solution to the problem stated in this introduction from a new point of view. We consider the transient dynamics (or the time evolution) of an unsteady fluid flow as a latent variable and we propose building a robust inferential model related to this variable. This might be helpful for the initialization of large eddy simulations in order to improve the efficiency of these simulations, by inferring to observed data the most accurate posterior approximation over this latent variable with respect to the true posterior. The problem of the initialization of transient simulations in turbulent fluid dynamics and the problem of turbulence injection are still open problems in the state of the art [23].

In general, uncertainty quantification and statistical methods are used in order to measure the effect of uncertainty or randomness in the parameters of a model on the solutions of this latter. Most of the time, this is applied to quantities of interest on the solutions. There are many ways to perform these uncertainty quantification and statistical methods: by applying a classical Monte Carlo approach [24] and by applying approximation techniques of the Monte Carlo method thanks to a large family of techniques, such as generative approaches where we can find, for example, the generative adversarial networks and the Bayesian approaches coupled to VAEs [25], the stochastic reduced order models such as polynomial chaos expansions, and the Gaussian process regressions coupled to data compression techniques such as the POD or with the pre-image problem for the KPOD [26], stochastic models of random matrices that appear in the nonparametric method of uncertainties [27], and the stochastic projection-based reduced-order model (SPROM) [28].

As already mentioned in the abstract, we chose the Bayesian approaches coupled to VAEs in order to generate plausible unsteady velocity fields of a compressible and unsteady fluid flow. The variational autoencoders introduced in [17,18] became very popular for tackling inferential techniques in latent spaces. In fact, we would like to take advantage of the inference model in order to be able to sharply estimate the posterior probability over the random conditional variable once the observed data have been taken into account. This allows us to draw samples in a generative approach and to generate plausible velocities of the fluid flow using the decoder of the VAE.

Practically, prior distribution over the random variable is used for the approximation of the true posterior following the Bayes theorem. In general, the prior distribution is a standard normal one. Hence, there is no practical way to verify the sharpness of the

inferred posterior approximation to the observed data over the random variable. In other words, the standard mean-field parametrization of the posterior approximation can limit the use of the VAE. It is only based on a prior belief that the considered standard normal prior distribution allows one to infer the true posterior to the observed data. Recent works on the state of the art tried to solve this problem. In [29], sampling from the VAE posterior approximation has been shown to improve the performance of the VAE. In [30], a Hamiltonian VAE (HVAE) inspired by Hamiltonian Monte Carlo (HMC) [31] which uses target information has been proposed.

In this paper, we propose a solution to this issue based on a data-targeted prior distribution based on the KPOD orthogonal coefficients with the observed data. Moreover, we enforce this data-targeted prior distribution thanks to a new training algorithm of the VAE, where the encoder approximation of the variance plays the role of a square error with respect to the data-targeted prior and not the variance of a probability distribution of a standard normal prior, as done in classical VAE. This enables us to interpret the inferred approximation of the true posterior and its parameterized mean function. Therefore, we were able to draw samples from the data-targeted parameterized mean and generate plausible velocities of the fluid flow using the VAE’s decoder. This new generative approach using the KPOD orthogonal coefficients is a simple procedure for expressing the pre-image in the physical space of the stochastic reduced field by the KPOD approach in the complex feature space.

The paper is organized as follows: in Section 2 we recall some definitions and notations necessary for the remainder of the paper; in Section 3, we give the problem and methodology formulations; in Section 4, we apply the new proposed algorithm on the instantaneous velocities of a reacting fluid flow in a fuel jet; and in Section 5, we conclude the paper and we give the prospects of this work.

2. Definitions and Notations

We recall in this section all the necessary definitions and we give the notations needed through the following sections.

2.1. VAE’s Cost Function

In Bayesian and variational methods, particularly for variational autoencoders, the optimized quantity is expressed as follows:

$$\mathcal{L}_{\text{classical}}(\Phi, \theta, u(t)) = E_{z \sim q_{\Phi}(z|u(t))}[-\log(p_{\theta}(u(t)|z))] + D_{KL}(q_{\Phi}(z|u(t)) || p_{\theta}(z|u(t))), \quad (1)$$

where, without any loss of generality:

- $u(t)$ is the observed data which in this case represent the instantaneous velocity field of an unsteady fluid flow; and $t \in [0, T]$, T is the maximum physical time.
- z is the random variable. We consider the transient dynamics (or the time evolution) of the unsteady fluid flow as a latent variable and we propose building a robust inferential model related to this variable.
- $q_{\Phi}(z|u(t))$ is a distribution over z which is optimized as an approximation of the true posterior $p_{\theta}(z|u(t))$. Hence, Φ represents the encoder parameters.
- $p_{\theta}(u(t)|z)$ is the direct probability of the occurrence of an observed variable $u(t)$ given a random variable z . Hence, θ are the decoder parameters.
- D_{KL} denotes the Kullback–Leibler divergence between the distribution approximation and the true posterior. As we do not have access to the true posterior of the observable fields, we assume an a priori distribution over z following the Bayes theorem.
- According to the Bayes theorem, the posterior probability is given by

$$p_{\theta}(z|u(t)) = \frac{p_{\theta}(u(t)|z)p_{\theta}(z)}{p_{\theta}(u(t))}. \quad (2)$$

where $p_\theta(u(t)) = \int_z p_\theta(u(t)|z)p_\theta(z)dz$ is the expectation of $p_\theta(u(t)|z)$ following $p_\theta(z)$.

2.2. Evidence Lower Bound

The evidence lower bound (ELBO) is defined by

$$\log(p_\theta(u(t))) - D_{KL}(q_\Phi(z|u(t))||p_\theta(z|u(t))) = E_{z \sim q_\Phi(z|u(t))} \log \frac{p_\theta(u(t)|z)p_\theta(z)}{q_\Phi(z|u(t))}, \quad (3)$$

where:

- $p_\theta(z)$ is the prior distribution on the latent variables and it is usually defined as a multivariate standard normal distribution: $p_\theta(z) = N(0, 1)$.
- As $p_\theta(u(t))$ does not depend on q , then Equation (3) shows that maximizing the evidence lower bound minimizes $D_{KL}(q_\Phi(z|u(t))||p_\theta(z|u(t)))$. Hence, the choice of prior distribution over z is very important in order to approximate at best the true posterior over z given an observed datum.

2.3. Physical Problem of Interest

We consider $u(t)$ as the instantaneous velocity field of the Navier–Stokes equations with a variable density. These equations are often used for turbulent flows with a low Mach number, for example, flows which we encounter in aeronautical combusters.

We write the evolution problem of the velocity $u(t)$ in a formal fashion as follows:

Find $u(t) \in V \subset X$, such that:

$$\frac{\partial u(t)}{\partial t} = A(u(t)), \quad (4)$$

where:

- V is a real Hilbert space.
- $X = [L^2(\Omega)]^d$, Ω is an open subset of \mathbb{R}^d .
- A is a formal representation of the convection–diffusion Navier–Stokes equations.

We denote by $(,)$ the endowed energetic inner product from X to V . We denote by $\| \cdot \|$ the associated norm, which is the norm of the square integrable \mathbb{R}^d -valued functions over Ω .

We define V_h as a subspace of V , spanned by M snapshots or instantaneous realizations of the unsteady velocity field u : these snapshots are denoted by $u_h(t_i), i = 1, \dots, M$.

2.4. KPOD and Kernel Trick

In this part, we recall the kernel proper orthogonal decomposition method. This technique can be seen as a non-linear dimension reduction in the Hilbert space V_h . The main motivation behind the use of the KPOD technique is that, many physical phenomena are hardly linearly separable or independent in a subspace of V_h of reduced dimension. This is typically the case of reacting fluid flows in aeronautical combustion chambers, where the flow velocity together with the flame propagation are governed by a transport behavior. In this case, the realizations are mostly linearly independent in V_h . The KPOD defines a map ϕ which relates the physical input space V_h to another space F_h :

$$\phi : V_h \longrightarrow F_h. \quad (5)$$

F_h is called the feature space. The realizations which are mostly linearly independent in V_h become linearly dependent in F_h which is endowed by a new inner product $(,)_{F_h}$ that enables accounting for the correlations between the realizations.

By definition, a POD basis associated with the set of realizations $\phi(u^h(t_i)), i = 1, \dots, M$, is an orthonormal basis $(\mathcal{V}_n)_{n \geq 1}$ of the space F_h endowed with the new inner product, which

maximizes the mean of the inner energy contained within the orthogonal projection of each realization over an orthonormal basis vector:

$$\frac{1}{M} \sum_{i=1}^M (\phi(u_h(t_i)), v)_{F_h}^2 = \max_{w \in F_h, \|w\|_{F_h}=1} \frac{1}{M} \sum_{i=1}^M (\phi(u_h(t_i)), w)_{F_h}^2 \tag{6}$$

The maximization problem (6) is equivalent to the search for the greatest eigenvalues of the following problem:

$$\mathcal{R}v = \lambda v, \tag{7}$$

where \mathcal{R} is defined as follows:

$$\begin{aligned} & F_h \rightarrow F_h \\ \mathcal{R} : v \mapsto \mathcal{R}v &= \frac{1}{M} \sum_{i=1}^M (u_h(t_i), v)_{F_h} u_h(t_i) \end{aligned} \tag{8}$$

\mathcal{R} is a Hilbert–Schmidt integral operator. Therefore, by the spectral theorem, the eigenvectors of this operator (solutions of Equation (7)) form an orthonormal basis of the space F_h . The associated eigenvalues are positive and decrease towards 0.

In practice, these eigenvalues are obtained following the classical snapshots POD approach introduced by Sirovich [4]: a singular value decomposition is performed to the correlations matrix defined as follows:

$$C_{ij} = (\phi(u_h(t_i)), \phi(u_h(t_j)))_{F_h}; i, j = 1, \dots, M. \tag{9}$$

The kernel trick is applied because ϕ is in general a complex map, in order to compute the scalar products of the correlations matrix. It is more convenient to denote the correlations (9) by \mathcal{K}_{ij} . We are particularly interested in the Gaussian kernel product, so these scalar products are given by

$$\mathcal{K}_{ij} = \exp\left(-\frac{\|u_h(t_i) - u_h(t_j)\|^2}{2\sigma^2}\right). \tag{10}$$

The kernel width parameter σ controls the flexibility of the kernel. As recommended in [32], a typical choice for σ is the average minimum distance between two realizations in the input physical space:

$$\sigma^2 = c \frac{1}{M} \sum_{i=1}^M \min_{j \neq i} \|u_h(t_i) - u_h(t_j)\|^2, j = 1, \dots, M. \tag{11}$$

where c is a user-defined parameter: a larger value of σ allows more mixing between elements of the realizations, whereas a smaller value of σ only uses a few significant realizations.

Therefore, as we usually do not have access to the mapping ϕ to compute the reduced order approximation in F_h , we can only compute the KPOD orthogonal projections $\alpha(t_i) = (\alpha_1(t_i), \dots, \alpha_M(t_i))$ as follows:

$$\alpha_n(t_i) = \sum_{k=1}^M a_k^n \mathcal{K}_{ki}. \tag{12}$$

where a^n is the n -th eigenvector of the correlations matrix (9).

We find in the state of the art many related works for solving the pre-image problem in the physical input space V_h of the reduced order approximation in the feature space F_h . More details about the pre-image problem can be found in [26,32–35].

2.5. VAEs

In this part, we recall the main concepts of variational autoencoders, all along with the reparametrization trick usually used in the training procedure of these neural network architectures.

A variational autoencoder is a particular way for studying Bayesian and variational approaches, where the probability distributions are approximated by the encoder and the decoder of the VAE and the parameters of these distributions are the ones optimized within the training phase of the VAE.

More precisely, we see in Figure 1 a formal representation of a VAE. The encoder $q_{\Phi}(z|u_h(t_i))$ on the left side of the figure is an approximation of the true posterior $p_{\theta}(z|u_h(t_i))$ (inverse probability) of the training realizations $u_h(t_i), i = 1, \dots, M$, over the random variable z . The latent random variable z at the end of the encoder layers is sampled following the mean $m(u(t_i), \Phi)$ (denoted $m_{\Phi}(u)$ on all the corresponding figures for simplicity) and the logarithm of the variance $\log(s^2(u_h(t_i), \Phi))$ (denoted $\delta_{\Phi}(u)$ on all the corresponding figures for simplicity), Φ being the optimization parameters of the encoder during the training of the VAE. In the formulation $z = m(u_h(t_i), \Phi) + \exp\left(\frac{1}{2} \log(s^2(u_h(t_i), \Phi))\right)U$, U is a random vector in \mathbb{R}^N (N being the dimension of z such that $N \ll M$) sampled from a standard normal distribution in each dimension. This formulation is usually considered during the training of the VAE as it allows the backward differentiation of the deep neural network only on deterministic quantities that depend on the parameters Φ of the encoder. The coordinates of U are independent and identically distributed following a standard normal distribution. This latter formulation is what is usually called the reparametrization trick. The decoder $p_{\theta}(u_h(t_i)|z)$ is the direct probability of occurrence of $u_h(t_i)$ given z , θ being the optimization parameters of this probability distribution. The latter is classically given as follows:

$$p_{\theta}(u_h(t_i)|z) = \exp\left(-\frac{\|u_h(t_i) - \mu_h(t_i)\|^2}{2}\right), \tag{13}$$

where $\mu_h(t_i)$ is the decoder output during the training phase given $u_h(t_i)$ as the input of the encoder, as shown in Figure 1.

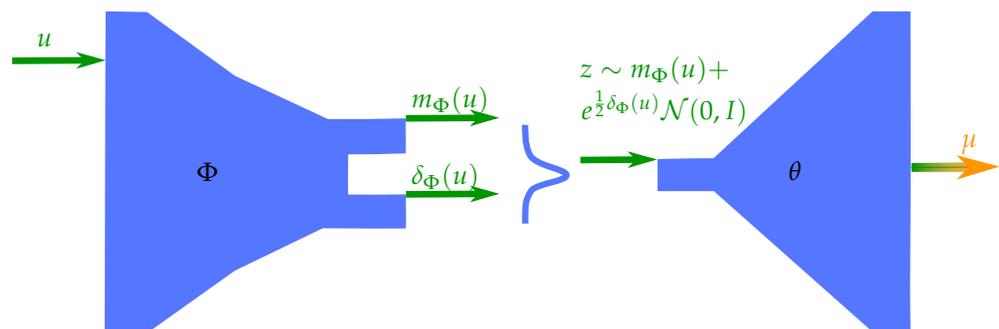


Figure 1. Schematical representation of the classical VAE.

The Kullback–Leibler divergence employed in classical VAEs (see Figure 1) and more generally, a common quantity in variational inference, is the relative entropy between a multivariate normal and standard normal distribution. Hence, the latent loss in VAEs is expressed as follows:

$$D_{KL}\left(N\left((m_1, m_2, \dots, m_N)^T, \text{diag}(s_1^2, s_2^2, \dots, s_N^2)\right) \parallel N(0, 1)\right) = \frac{1}{2} \sum_{i=1}^N \left(s_i^2 + m_i^2 - 1 - \ln(s_i^2)\right). \tag{14}$$

3. Problem and Methodology Formulations

3.1. Motivation for the Method

As already discussed, the choice of prior distribution $p_\theta(z)$ over the latent random variable in Bayesian techniques is very important. In this paper, we propose a data-targeted prior distribution that improves the results obtained by a classical VAE with a standard normal prior distribution. Furthermore, our implementation of this new prior distribution is verified under any choice of architecture for the encoder and the decoder parts of the VAE.

In the literature, we can find attempts of using prior distributions different from the multivariate standard normal distribution. We cite the work of Partaourides et al. [36], where a class of asymmetric deep generative models (AsyDGMs) has been proposed, characterized by asymmetric latent variable posteriors that are formulated as restricted multivariate skew-normal (rMSN) distributions. However, in this work, the neural network's architecture is considered always linear because of the difficulty of formulating the reparametrization trick in this particular case.

We also found attempts to improve the quality of a VAE by changing the direct probability. We cite the work of Berger et al. [37], where a new distribution class has been proposed for the observation model, i.e., the direct probability. Hence, the reconstruction likelihood is changed by supposing that the variance of the normal distribution is a learned parameter. Therefore, the likelihood is not a simple squared error loss.

In this paper, we are interested in proposing a new class of deep generative models based on data-targeted prior distribution defined by the kernel proper orthogonal decomposition orthogonal coefficients with the training data. This idea was inspired by model order reduction techniques based on linear projection approaches in the physical space of training fields [8,38,39] or even non-linear ones in the physical space such as the KPOD approach [26,32–35], where the obtained projection coefficients are used to do reconstructions by a linear combination all along a POD reduced basis in the physical space V_h or in the feature space F_h . By analogy, we would like to take advantage of these KPOD coefficients in order to propose a new prior distribution over the latent variable of a VAE. From this data-targeted prior distribution over the latent variable, we would like to reconstruct the physical fields—this time thanks to the neural network's layers. This is a way to deal with the pre-image problem in the KPOD approach thanks to the decoder output in the physical input space V_h . Moreover, the choice of KPOD projection coefficients has already been motivated by the fact that for complex physical phenomena with transport behavior, such as the compressible turbulent fluid flows or the reacting fluid flows with a variable density, the linear dimensionality reduction is not possible within the physical input space where all the instantaneous velocity fields are linearly independent. Therefore, a reduced number of POD orthogonal coefficients that precisely describes the dynamics of the fluid flow does not exist. In [40], a deterministic autoencoder was used in order to approximate the manifold data over which the model's equations were applied to define a new projection-based reduced order model. The resolution of this reduced order model gives access to the dynamics of the latent variable. The final solution is then obtained from the application of the decoder layers on the physical latent variable. In this paper, we use VAEs and we do not solve the model's equations; however, we take advantage of the generative model of a VAE thanks to its latent space learned as a probability distribution.

Therefore, hereafter $p_\theta(z)$ denotes the prior distribution over z defined by the KPOD projection coefficient $(\alpha_n(t_i))_{i=1,\dots,M}$.

Remark 1. *The proposed prior distribution $p_\theta(z)$ can be approximated by performing a kernel density estimate over the total M realizations of the KPOD coefficients available for each dimension of the random latent space. We explain the details of this computation in the following section relative to the numerical experiments.*

Remark 2. *The proposed methodology combining from one side the prior distribution of the KPOD orthogonal projection coefficients with the available physical fields and from the other side, the genera-*

tive variational approaches by VAEs, can also be seen as a new method for computing reconstructions and new generations in the input physical space V_h from the feature high-dimensional space F_h , which is usually performed by pre-image approaches.

3.2. Framework for the Implementation of the VAE with a Data-Targeted Prior Distribution

It is important to implement this new approach in a way that ensures its use with all types of neural network architectures—the fully connected ones and the deep ones. We previously mentioned the work of Partaourides et al. [36], where for the proposition of a new class of asymmetric deep generative models, the neural network’s architecture was always considered linear. The practical difficulty in training VAEs is the reparametrization trick, so that the source of randomness in the reduced latent space does not impact the accuracy of the backward differentiations during the optimization of the network’s parameters Φ and θ . For multivariate normal distributions, the expression of the reparametrization trick is very simple. The separation in the latent variable z between the parametric functions and the source of randomness is ensured by sampling the mean and the logarithm of the variance within the layers, following a random vector satisfying a standard normal distribution. Therefore, when we need to change the prior distribution $p_\theta(z)$, as we are proposing in this paper to replace the standard normal prior with a new data-targeted one based on the KPOD coefficients, it is often very difficult to implement the reparametrization trick for deep neural networks.

We propose the following framework in Figure 2, in order to take into account the new latent distribution $p_\theta(z)$ without the reparametrization trick. As shown in Figure 2, we decided to keep, on the one hand, the sampling of the random latent vector z following a multivariate normal distribution, and on the other hand, the new prior distribution $p_\theta(z)$ for which we have M available realizations $\alpha_n(t_i), i = 1, \dots, M$ in each n -th ($n = 1, \dots, N$) dimension. The latter is imposed when the input of the encoder is the random output field $\mu_h(t_i)$. More precisely, the new prior is imposed by backward differentiation which also minimizes the squared error loss between $z' \sim q_\Phi(z'|\mu_h(t_i))$ and $\alpha(t_i): \|z'(t_i) - \alpha(t_i)\|_{\mathbb{R}^N}^2$. The new loss function of the VAE becomes as it is expressed in Equation (15). Henceforth, the backward differentiation during the learning phase is performed without the need for a new reparametrization trick as we are penalizing the random variable z' (following a normal distribution for which we apply the usual reparametrization trick) with fixed realizations of the prior $p_\theta(z)$ instead of sampling z' from $p_\theta(z)$ using the associated inverse cumulative distribution function for which the reparametrization trick is a function of $m(\mu_h(t_i), \Phi)$ and $s(\mu_h(t_i), \Phi)$ is unknown even in this case and then, computing $D_{KL}(q_\Phi(z|u_h(t_i))||p_\theta(z))$:

$$\mathcal{L}_{\text{data-targeted}}(\Phi, \theta, u_h(t_i)) = E_{z \sim q_\Phi(z|u(t))}[-\log(p_\theta(u_h(t_i)|z))] + D_{KL}(q_\Phi(z|u_h(t_i))||N(0, 1)) + \|z'(t_i) - \alpha(t_i)\|_{\mathbb{R}^N}^2. \tag{15}$$

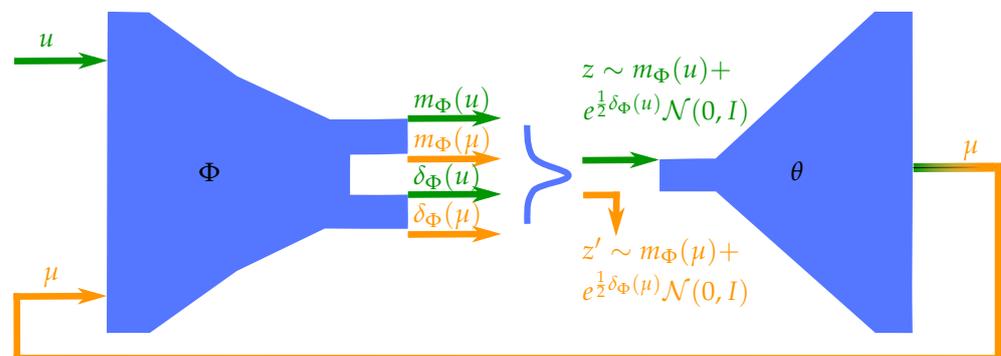


Figure 2. Schematical representation of the proposed data-targeted VAE.

Remark 3. In the framework proposed in Figure 2, we notice that applying the new penalization cost function $\|z'(t_i) - \alpha(t_i)\|_{\mathbb{R}^N}^2$ on the probability distribution over z' of the decoder outputs $\mu_h(t_i)$ considerably contributed to minimizing it, as illustrated in the following numerical experiments. However, by applying it on the probability distribution over z of the training data $u_h(t_i)$, $\|z(t_i) - \alpha(t_i)\|_{\mathbb{R}^N}^2$ decreased a little bit but it very quickly stagnated.

Remark 4. This framework can be seen as a penalization technique in order to force the random latent vector z' during the training stage to be close to the realizations of our prior distribution $p_\theta(z)$. A direct impact is then obtained on the parameterized mean and the logarithm of variance from the random input fields $\mu_h(t_i)$, in order to fulfill the new prior distribution. This impact is also seen on the parameterized mean and the logarithm of the variance from the training deterministic input fields $u_h(t_i)$, as long as the likelihood cost function $E_{z \sim q_\Phi(z|u(t))} - \log(p_\theta(u(t)|z))$ is minimized.

4. Numerical Experiments

4.1. Flow Solver

For the presented simulations, the low-Mach number solver YALES2 [41] for unstructured grids is retained. This flow solver was specifically tailored for the direct numerical simulation and large eddy simulation of turbulent reacting flows on large meshes counting several billion cells using massively parallel super-computers [42,43]. The Poisson equation that arises from the low-Mach formulation of the Navier–Stokes equations is solved with a highly efficient deflated preconditioned conjugated gradient method [43].

The following test case was chosen from a list of test cases available from the sources of the YALES2 code.

4.2. Governing Equations, Test-Case and Training Set

The governing equations are the Navier–Stokes equations with variable density. They are formulated as follows:

$$\begin{aligned} \frac{\partial \rho_h}{\partial t} + \text{div}(\rho_h(t)u_h(t)) &= 0 \\ \frac{\partial \rho_h u_h}{\partial t} + \text{div}(\rho_h u_h \otimes u_h) - \text{div}(\nu[\nabla u_h + (\nabla u_h)^T]) + \nabla p_h &= 0 \\ \text{div}u_h &= 0 \\ (\rho_h, u_h)_{(t=0)} &= (\rho_0, u_0) \end{aligned} \tag{16}$$

$\rho_h(t, x)$ is the density of a particle at a position x in \mathbb{R}^d and a time instant t , ν is the air viscosity, and p_h is the pressure field.

The test case is a simple 2D model of an aeronautical injector, for which an illustration of the unstructured mesh used for the computation and an example of the x-component of an instantaneous velocity field can be seen in Figure 3. A mixture of air and fuel enters the domain from the channel on the left, at inlet velocities of 0.1 m/s for the air and 0.05 m/s for the fuel. Wall boundary conditions are enforced at the top and lower boundaries, and an outlet boundary condition is enforced on the right extremity of the domain. The Mach number for this test case is equal to $3 \cdot 10^{-4}$ and the Reynolds number is equal to 50 based on the inlet fuel velocity.

The training set $u_h(t_i)$ and $p_h(t_i)$, $i = 1, \dots, M$, necessary for the optimization of the VAE parameters, is formed of 998 snapshots of the high-fidelity velocity and pressure extracted at each time step. These 998 snapshots are taken among 5000 time steps of the high-fidelity simulation corresponding to 500 ms.

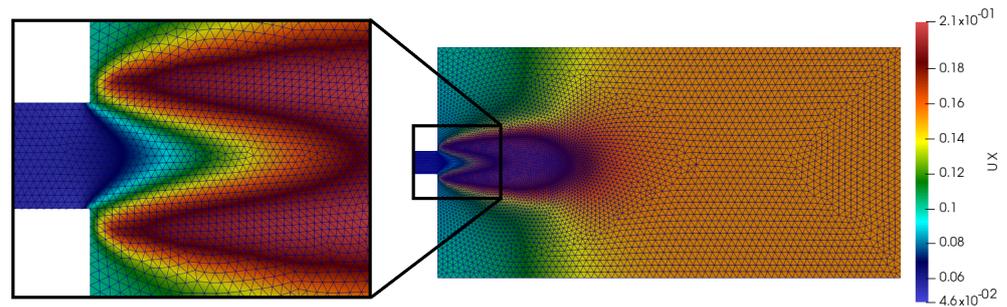


Figure 3. Illustration of the mesh and the example of the x-component of an instantaneous velocity field of the considered test-case.

4.3. KPOD Orthogonal Coefficients with the Training Realizations

We present in Figure 4 the evolution of the first two KPOD orthogonal coefficients, $\alpha_1(t_i)$ and $\alpha_2(t_i)$, $i = 1, \dots, 998$, associated with the training velocity and pressure fields (scaled between 0 and 1). The computation of these coefficients was previously detailed in (10) and in (12). We precise that we chose the user-defined parameter c , as can be seen in Equation (11), such that $\sigma^2 = 15$ for the Gaussian kernel computation (10).

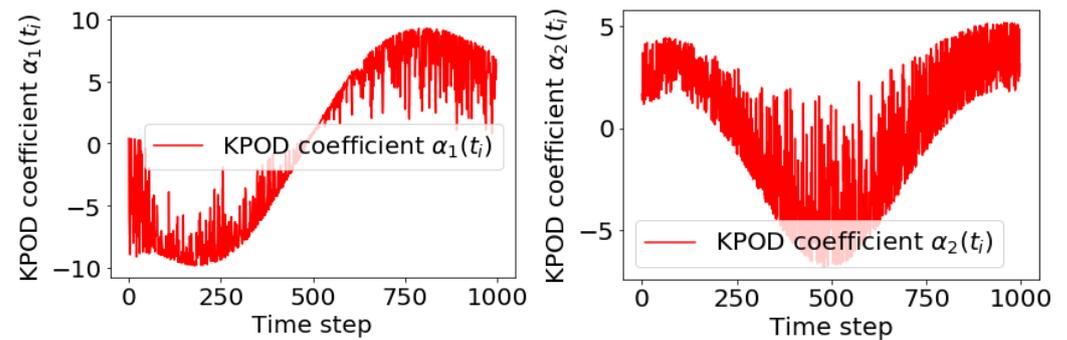


Figure 4. KPOD orthogonal coefficients $\alpha_1(t_i)$ (left) and $\alpha_2(t_i)$ (right), with respect to the time step $i = 1, \dots, M$.

4.4. Variational Autoencoder Architecture

The framework proposed in Figure 2 was presented under any choice of architecture for the encoder and the decoder parts of the VAE. Therefore, and without any loss of generality, we chose to apply our new inferential model on a classical deep convolutional architecture illustrated in Figure 5. The latent space dimension denoted by N is set as equal to 2.

In order to solve the optimization problem of the VAE parameters, we used the ADAM optimizer, introduced in [44], which is an algorithm for the first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of a lower-order moment. We set the learning rate as equal to 10^{-3} .

Remark 5 (Projection and inverse projection). *The vector containing the discretized solution on the unstructured mesh (illustrated in Figure 3) has no information on the geometrical proximity of its components without the connectivity of the mesh. The solution expressed as a two-dimensional array on a Cartesian mesh has such information: two values having close indices in this array are also close in the physical domain. We can apply convolutions on this array. Projections are needed to express the discretized YALES2 solution on a 48×48 Cartesian mesh, compatible with convolution neural networks, while inverse projections are needed to express the approximations generated by the neural network back onto the physical unstructured mesh. These operations were illustrated in Figure 6. The finite element interpolation was used for both the projection and inverse projection—the Cartesian mesh being converted to a quadrilateral-based unstructured mesh with the same vertices before applying the inverse projection.*

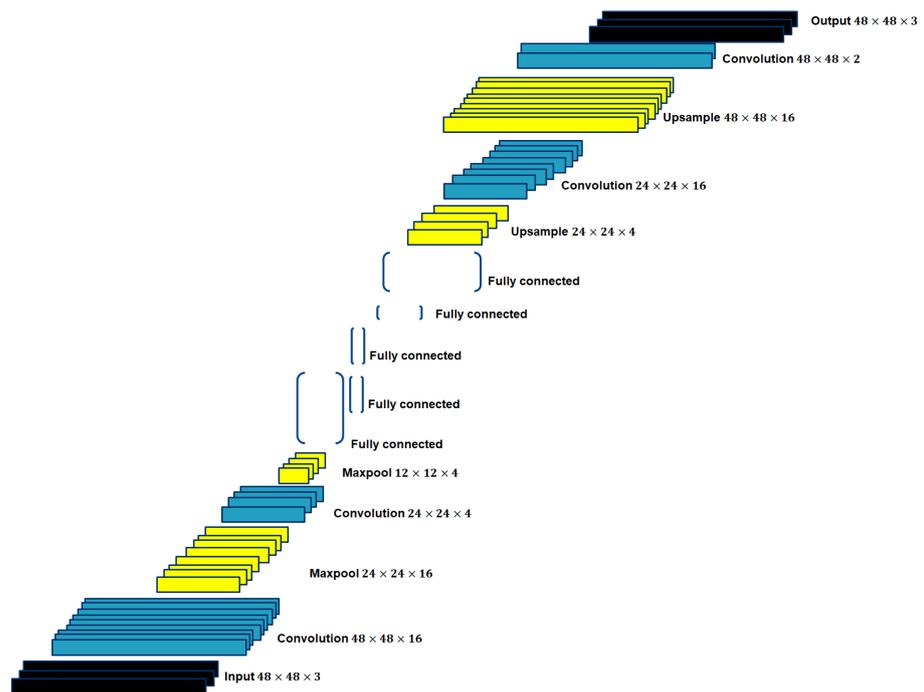


Figure 5. Illustration of the VAE architecture.

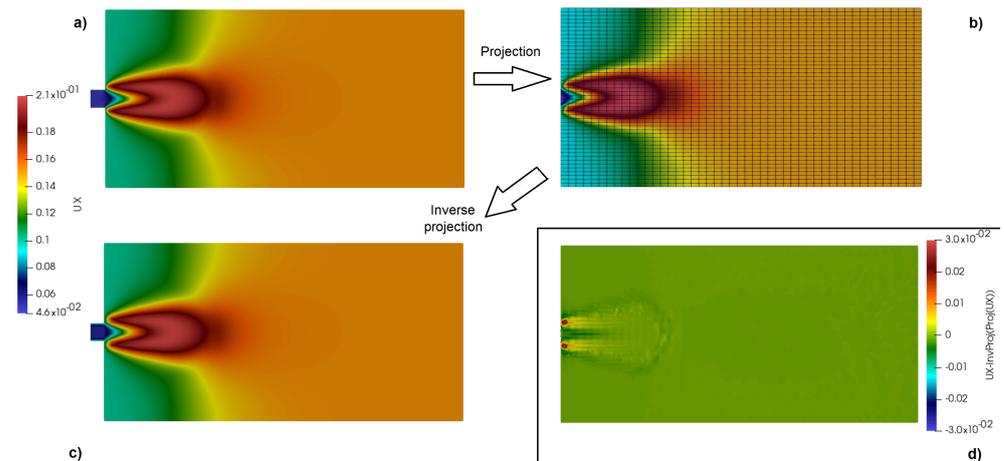


Figure 6. Illustration of the projection and inverse projection operations: (a) example of a YALES2 solution on the unstructured mesh; (b) projection of (a) on the 48×48 Cartesian mesh; (c) inverse projection of (b) on the unstructured mesh; (d) difference between (a,c), which corresponds to the errors introduced by both the projection and inverse projection steps.

In what follows, when comparing the classical VAE (from Figure 1) with the proposed data-targeted VAE (from Figure 2), we keep the same encoder and decoder architectures and parameters for the learning task, namely the same batch sizes, numbers of epochs, learning rate and optimizer.

4.5. Reconstructions and New Generations

In what follows, all fields are expressed and represented on the unstructured mesh (used for the YALES2 computations), after applying the inverse projection procedure, as can be seen in Remark 5. Relative errors are also computed after inverse projection.

4.5.1. Comparison of Reconstructed Fields during the Training Phase

We investigated the results at the end of the training phase for the classical and data-targeted VAEs in Figures 7–12. Moreover, the logarithm of the instantaneous relative errors

with respect to LES solutions are shown in Figure 13. We precise that these results were taken from the last 4000-th training epoch.

In Figure 14, we show the Kullback–Leibler divergence during the training phase. We remark that the probability distribution of the encoder does not converge towards the standard normal prior, even though the reconstruction loss converges on the output data, as we can remark in Figure 13.

The penalization loss $\|z' - \alpha(t)\|^2$ in Figure 15 impacts the posterior approximation of the training data $u_h(t_i)$: this is performed by changing the posterior approximation of the decoder outputs $\mu_h(t_i)$, as can be seen in Figures 16 and 17. Following Remark 4, we show on Figures 18 and 19 the evolution of the parameterized mean and logarithm of the variance of the training data $u_h(t_i)$ for the data-targeted VAE. We remark that these latter are no longer the parameterized functions of a standard normal distribution. Instead, the parametric variance, illustrated in Figure 19, represents the square error with respect to the data-targeted prior. Hence, the parameterized mean of the training data in Figure 18 by the data-targeted VAE encoder is interpretable. Moreover, we remark that the parameterized mean of the training data is well contained in the prior latent space represented by the KPOD realizations, thanks to this new training algorithm. This result shows the good convergence of our algorithm and a good compression of the training data within the data-targeted latent space.

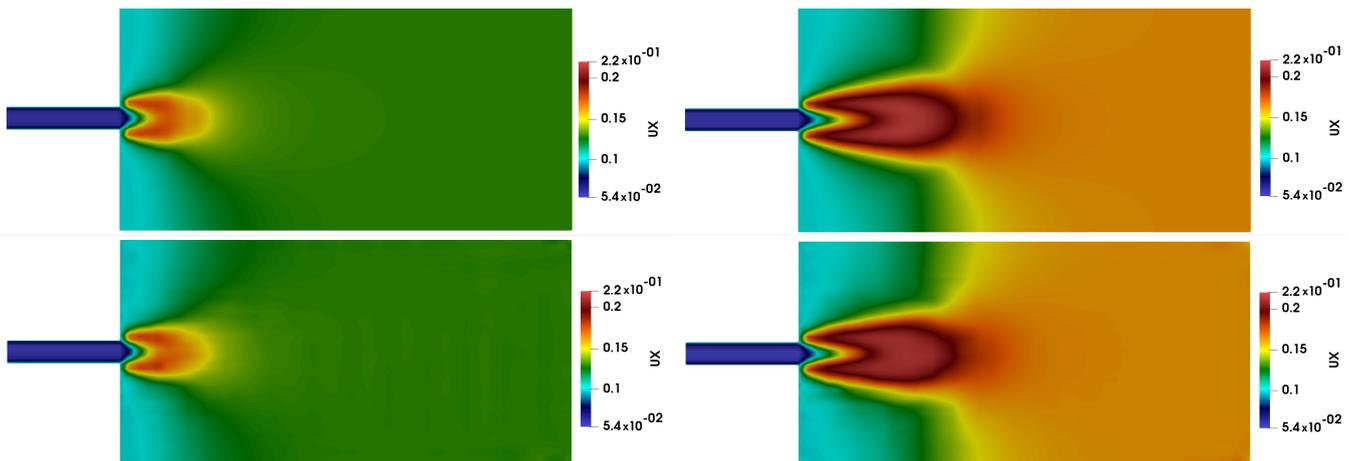


Figure 7. (Data-targeted VAE) On the top, the high-fidelity first component of the velocity fields for the two time steps 499 and 998. On the bottom, the decoder outputs for the two time steps 499 and 998 at the final 4000-th epoch of the training phase.

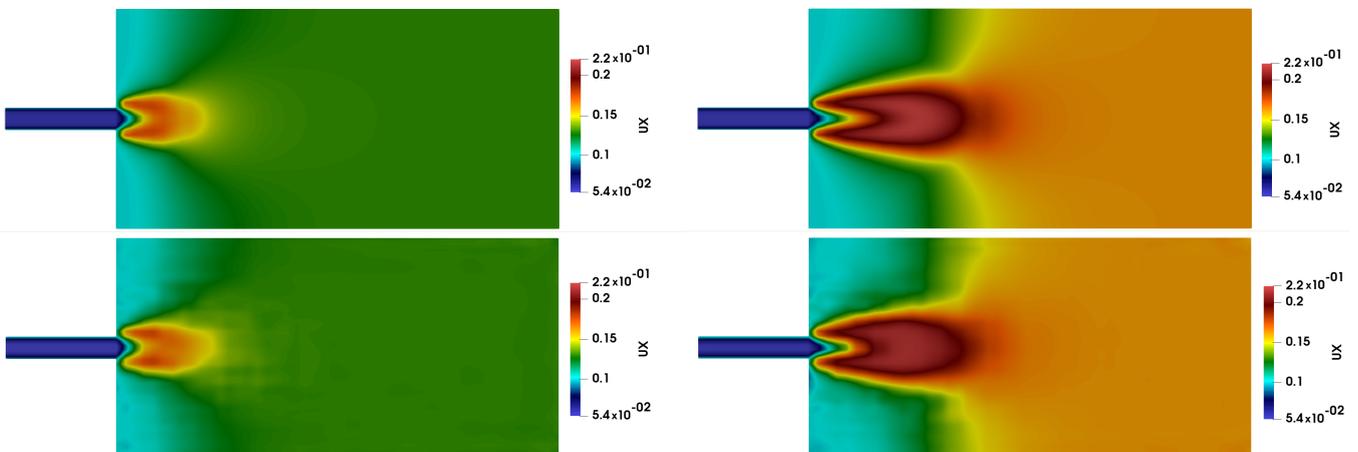


Figure 8. (Classical VAE) On the top, the high-fidelity first component of the velocity fields for the two time steps 499 and 998. On the bottom, the decoder outputs for the two time steps 499 and 998 at the final 4000-th epoch of the training phase.

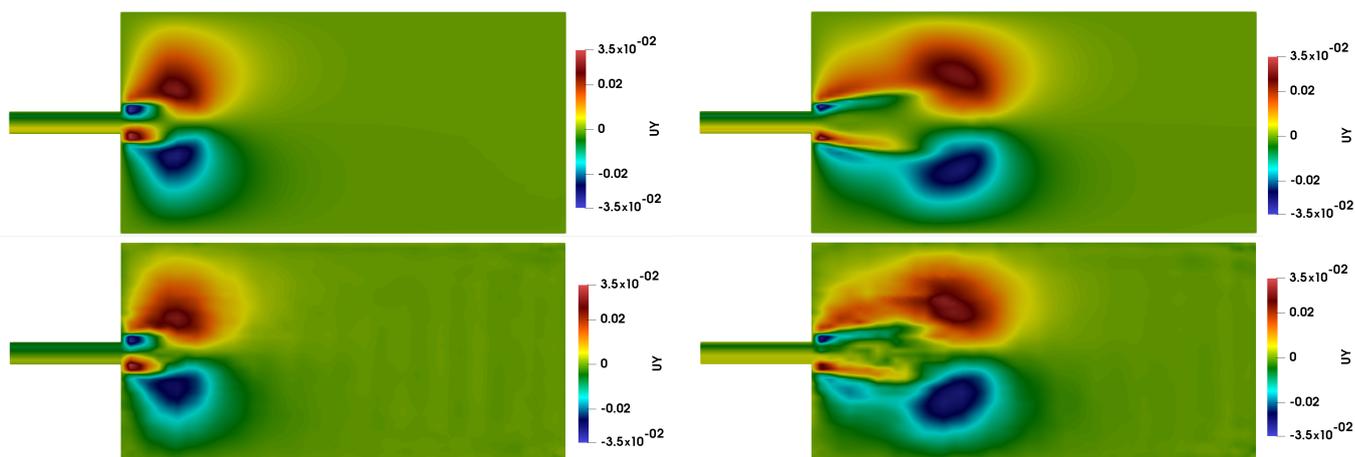


Figure 9. (Data-targeted VAE) On the top, the high-fidelity second component of the velocity fields for the two time steps 499 and 998. On the bottom, the decoder outputs for the two time steps 499 and 998 at the final 4000-th epoch of the training phase.

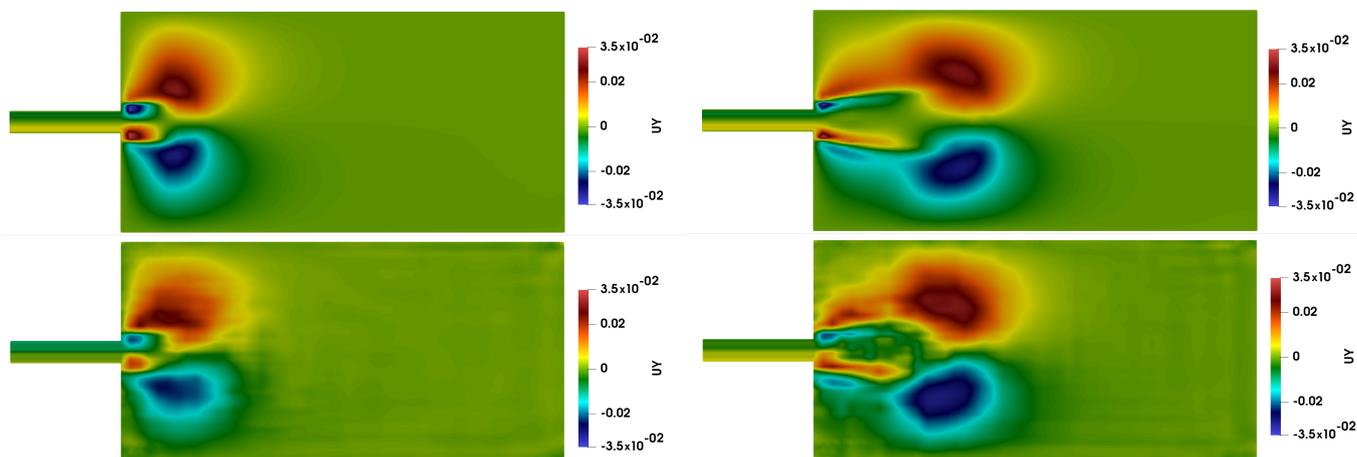


Figure 10. (Classical VAE) On the top, the high-fidelity second component of the velocity fields for the two time steps 499 and 998. On the bottom, the decoder outputs for the two time steps 499 and 998 at the final 4000-th epoch of the training phase.

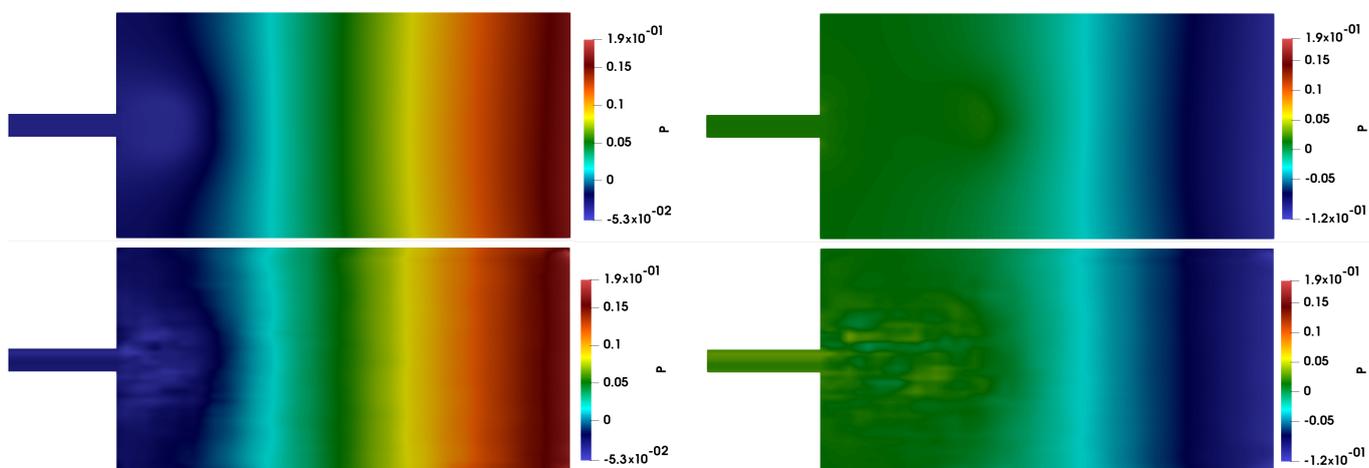


Figure 11. (Data-targeted VAE) On the top, the high-fidelity pressure of the velocity fields for the two time steps 499 and 998. On the bottom, the decoder outputs for the two time steps 499 and 998 at the final 4000-th epoch of the training phase.

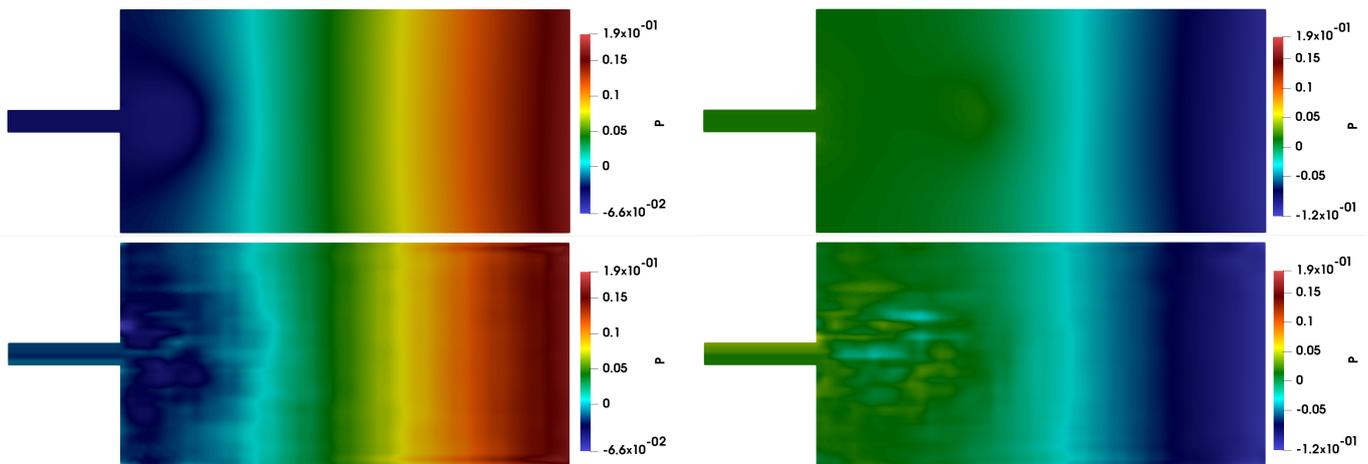


Figure 12. (Classical VAE) On the top, the high-fidelity pressure of the velocity fields for the two time steps 499 and 998. On the bottom, the decoder outputs for the two time steps 499 and 998 at the final 4000-th epoch of the training phase.

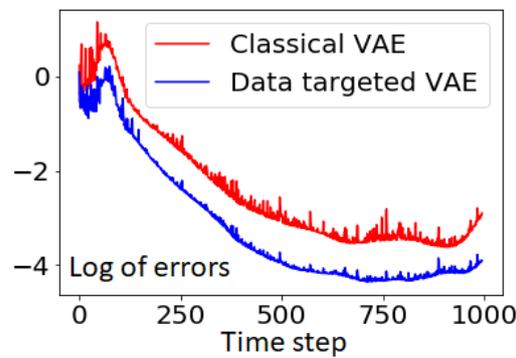


Figure 13. In red: $\log\left(\frac{\|\mu_h(t_i) - u_h(t_i)\|^2}{\|u_h(t_i)\|^2}\right)$ for the classical VAE; in blue: $\log\left(\frac{\|\mu_h(t_i) - u_h(t_i)\|^2}{\|u_h(t_i)\|^2}\right)$ for the data-targeted VAE.

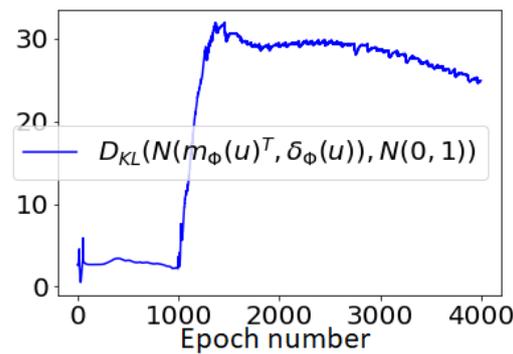


Figure 14. (Classical VAE) $D_{KL}(N((m_1, m_2)^T, \text{diag}(s_1^2, s_2^2)) || N(0, 1)) = \frac{1}{2} \sum_{i=1}^2 (s_i^2 + m_i^2 - 1 - \ln(s_i^2))$, with respect to the epoch number on the abscissa axis.

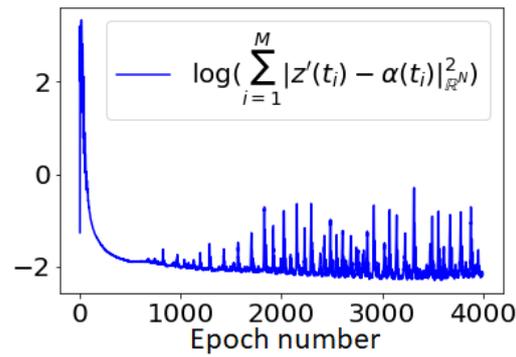


Figure 15. (Data-targeted VAE) $\log(\sum_{i=1}^M \|z'(t_i) - \alpha(t_i)\|_{\mathbb{R}^N}^2)$, $z' \sim q_{\Phi}(z'|\mu_h(t_i))$, with respect to the epoch number on the abscissa axis.

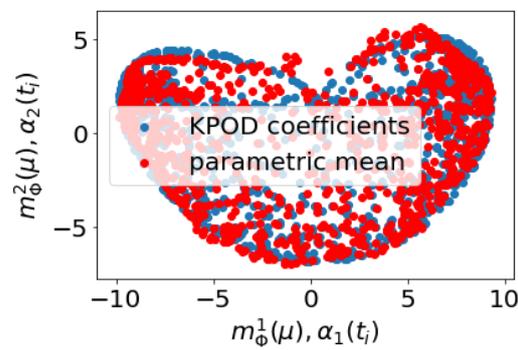


Figure 16. (Data-targeted VAE) Parametric plots of the two components of $m(\mu_h(t_i), \Phi)$ (red) and $\alpha(t_i)$ (blue), $i = 1, \dots, M$.

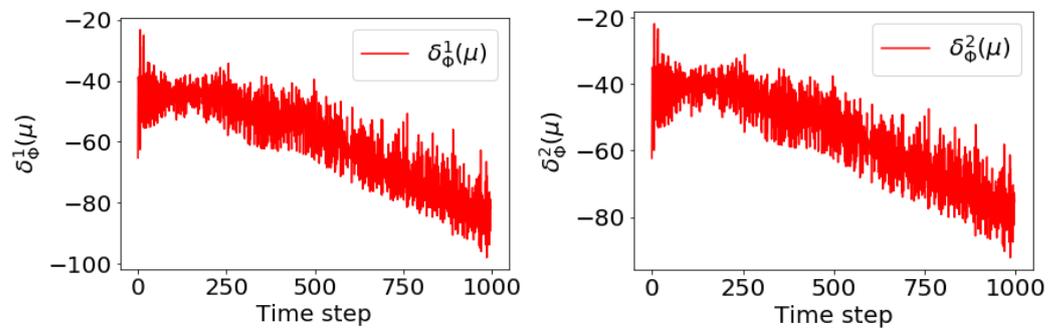


Figure 17. (Data-targeted VAE) First component (left) and second component (right) of $\log(s^2(\mu_h(t_i), \Phi))$ with respect to the time step $i = 1, \dots, M$.

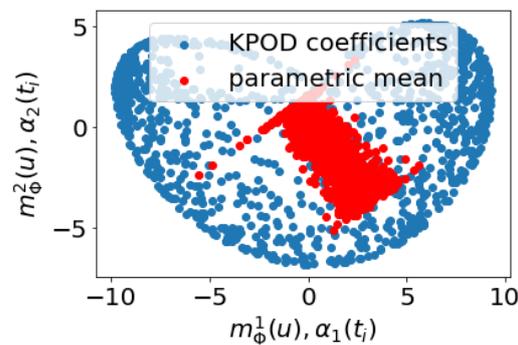


Figure 18. (Data-targeted VAE) In red, the parametric plot of the two components of the data-targeted $m(u_h(t_i), \Phi)$, $i = 1, \dots, M$. In blue, the parametric plot of the two components of $\alpha(t_i)$, $i = 1, \dots, M$.

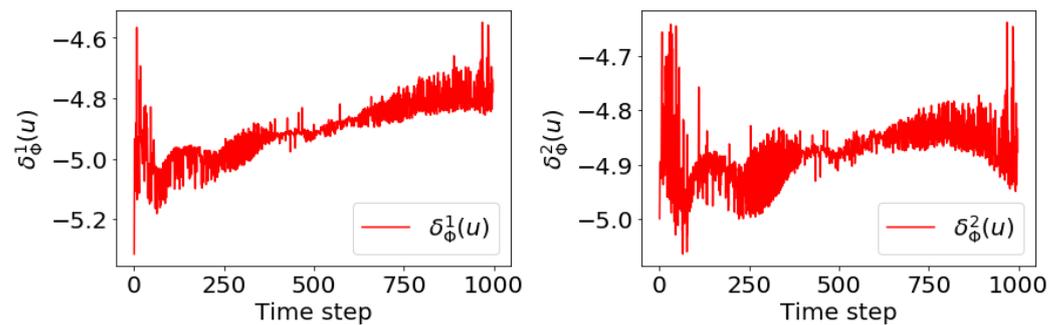


Figure 19. (Data-targeted VAE) First component (left) and second component (right) of $\log(s^2(u_h(t_i), \Phi))$ with respect to the time step $i = 1, \dots, M$.

4.5.2. Comparison of Generated Fields during the Exploitation Phase

Instantaneous velocity fields are generated by drawing samples of z following the prior distribution $p_{\theta}(z)$.

In the classical VAE, samples of z are drawn from the standard normal distribution by using available open source libraries, such as the Pytorch library. In the data-targeted VAE, $\alpha_n(t_i), i = 1, \dots, 998$ are realizations of the prior. A kernel density estimate is performed over the 998 training realizations of the two coefficients of the KPOD method, $\alpha_1(t_i)$ and $\alpha_2(t_i)$, using the module `stats.kde` of the library `Scipy` in Python. The cumulative density function (CDF) is obtained afterwards using this kernel density estimate. However, we need, respectively, the inverses of these CDFs in order to select a random variable and we need to be able to interpolate between the 998 realizations of the inverses of these CDFs. These realizations are fitted using an univariate spline model, from the module `interpolate` of the library `Scipy` in Python.

Finally, to draw samples of a variable following the distribution over the KPOD coefficients shown in Figure 4, we perform the inverse transformation method, i.e., we draw samples of a variable that follows a uniform distribution between 0 and 1 over which we apply the fitted univariate spline model of the inverses of the CDFs.

We draw 998 samples of z for each case. We show the generated fields which are also denoted $(\mu_h(t), p_h(t)) \sim p_{\theta}((\mu_h(t), p_h(t))|z)$. Furthermore, we show at given time steps t_i of the high-fidelity density ρ_h the spatial residue of the high-fidelity mass conservation Equation (see (16)) with respect to the i -th generated velocity fields $\mu_h(t)$: $\frac{\partial \rho_h(t_i)}{\partial t} + \text{div}(\rho_h(t_i)\mu_h(t))$. This residue is computed on the Cartesian mesh by projecting the high-fidelity density field on the 48×48 Cartesian mesh. We compute this residue using a finite difference scheme for the time derivative of the density and second-order accurate central differences in the interior points and either first- or second-order accurate one-side (forward or backwards) differences at the boundaries for the divergence term $\text{div}(\rho_h(t)\mu_h(t))$. We compare this residue with respect to the one associated with the reconstructed and accurate velocity fields $u_h(t_i)$ on the Cartesian mesh: $\frac{\partial \rho_h(t_i)}{\partial t} + \text{div}(\rho_h(t_i)u_h(t_i))$. These results are summarized in Figures 20–22. We deduce that the velocity fields generated by the data-targeted VAE satisfy fairly well the mass conservation property, as the residue of the mass conservation equation by these latter is comparable to the one with the accurately reconstructed velocity fields $u_h(t)$. This result was explained by the good compression of the training data within the data-targeted latent space (see Figure 18), thanks to the proposed training algorithm of the data-targeted VAE. However, the generated velocity fields by the classical VAE do not satisfy the mass conservation; this result was not surprising because the encoder probability distribution over the training data did not converge to the standard normal prior (see Figure 14), leading to a bad compression of the training data within the latent space.

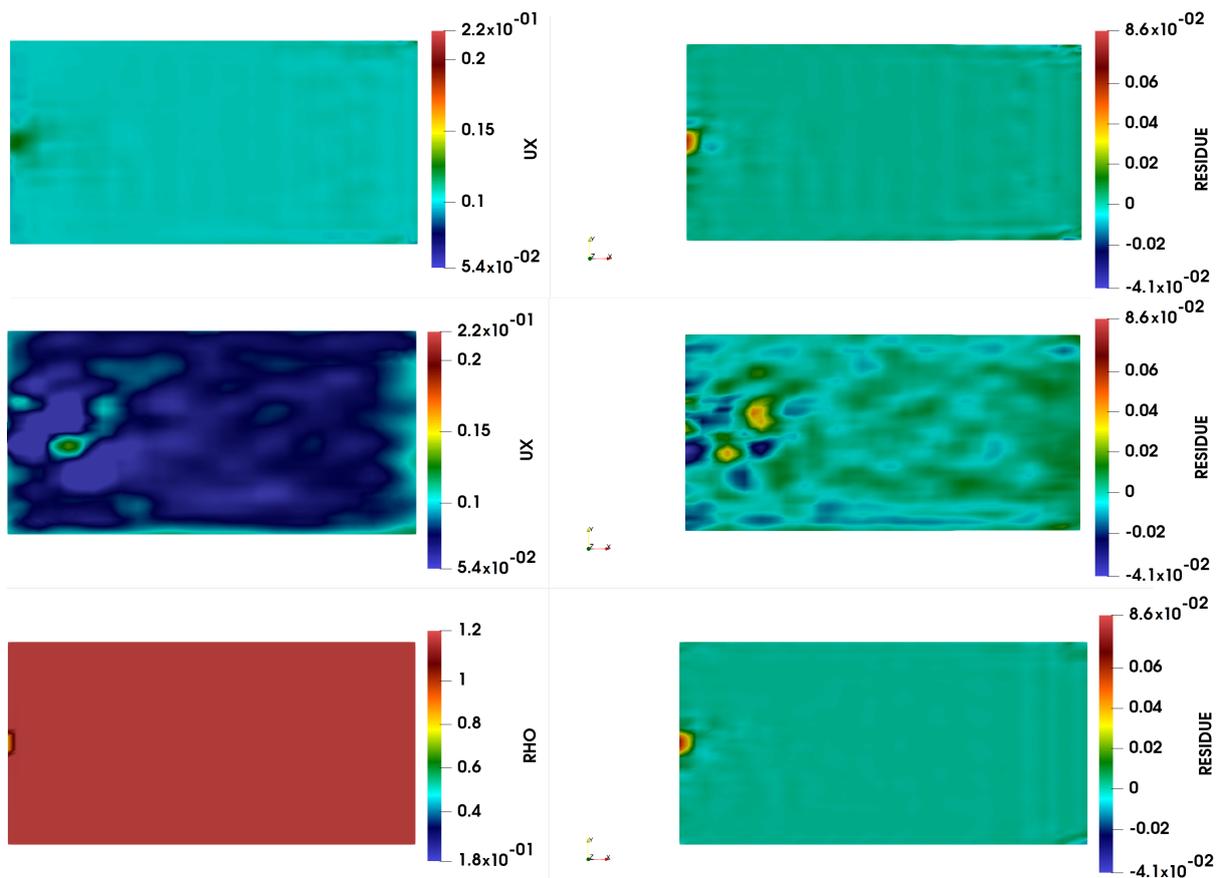


Figure 20. On the top left, the 32-th decoder generation $\mu_h(t) \sim p_\theta(\mu_h(t)|z)$ from a sample of the probability distribution $p_\theta(z)$ for the data-targeted VAE. On the top right, the residue of this generated sample, $\frac{\partial \rho_h(t_{32})}{\partial t} + \text{div}(\rho_h(t_{32})\mu_h(t))$. On the middle left, the 32-th decoder generation $\mu_h(t) \sim p_\theta(\mu_h(t)|z)$ from a sample of the probability distribution $N(0,1)$ for the classical VAE. On the middle right, the residue of this generated sample, $\frac{\partial \rho_h(t_{32})}{\partial t} + \text{div}(\rho_h(t_{32})\mu_h(t))$. On the bottom left, the high-fidelity density field $\rho_h(t_{32})$ projected on the Cartesian mesh. On the bottom right, the accurate residue $\frac{\partial \rho_h(t_{32})}{\partial t} + \text{div}(\rho_h(t_{32})u_h(t_{32}))$.

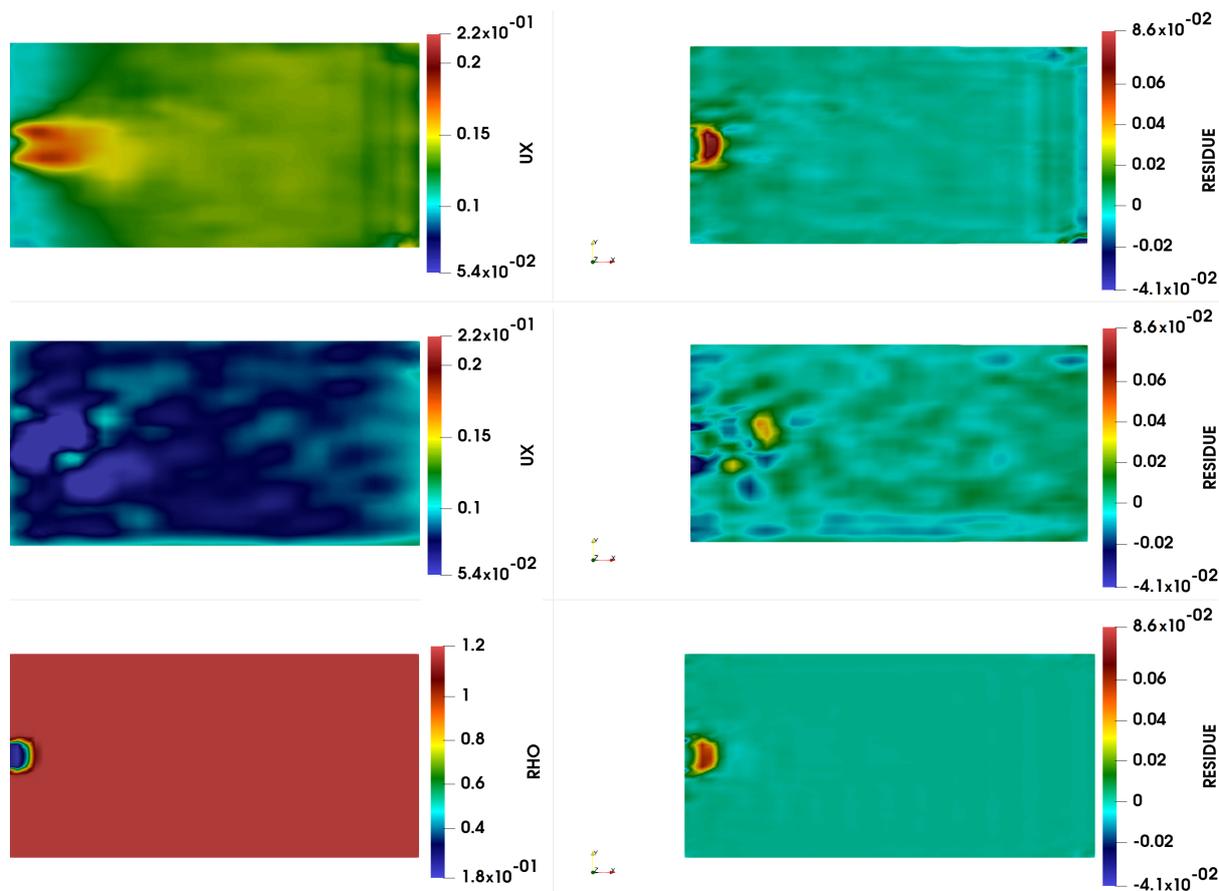


Figure 21. On the top left, the 202-th decoder generation $\mu_h(t) \sim p_\theta(\mu_h(t)|z)$ from a sample of the probability distribution $p_\theta(z)$ for the data-targeted VAE. On the top right, the residue of this generated sample, $\frac{\partial \rho_h(t_{202})}{\partial t} + \text{div}(\rho_h(t_{202})\mu_h(t))$. On the middle left, the 202-th decoder generation $\mu_h(t) \sim p_\theta(\mu_h(t)|z)$ from a sample of the probability distribution $N(0,1)$ for the classical VAE. On the middle right, the residue of this generated sample, $\frac{\partial \rho_h(t_{202})}{\partial t} + \text{div}(\rho_h(t_{202})\mu_h(t))$. On the bottom left, the high-fidelity density field $\rho_h(t_{202})$ projected on the Cartesian mesh. On the bottom right, the accurate residue $\frac{\partial \rho_h(t_{202})}{\partial t} + \text{div}(\rho_h(t_{202})u_h(t_{202}))$.

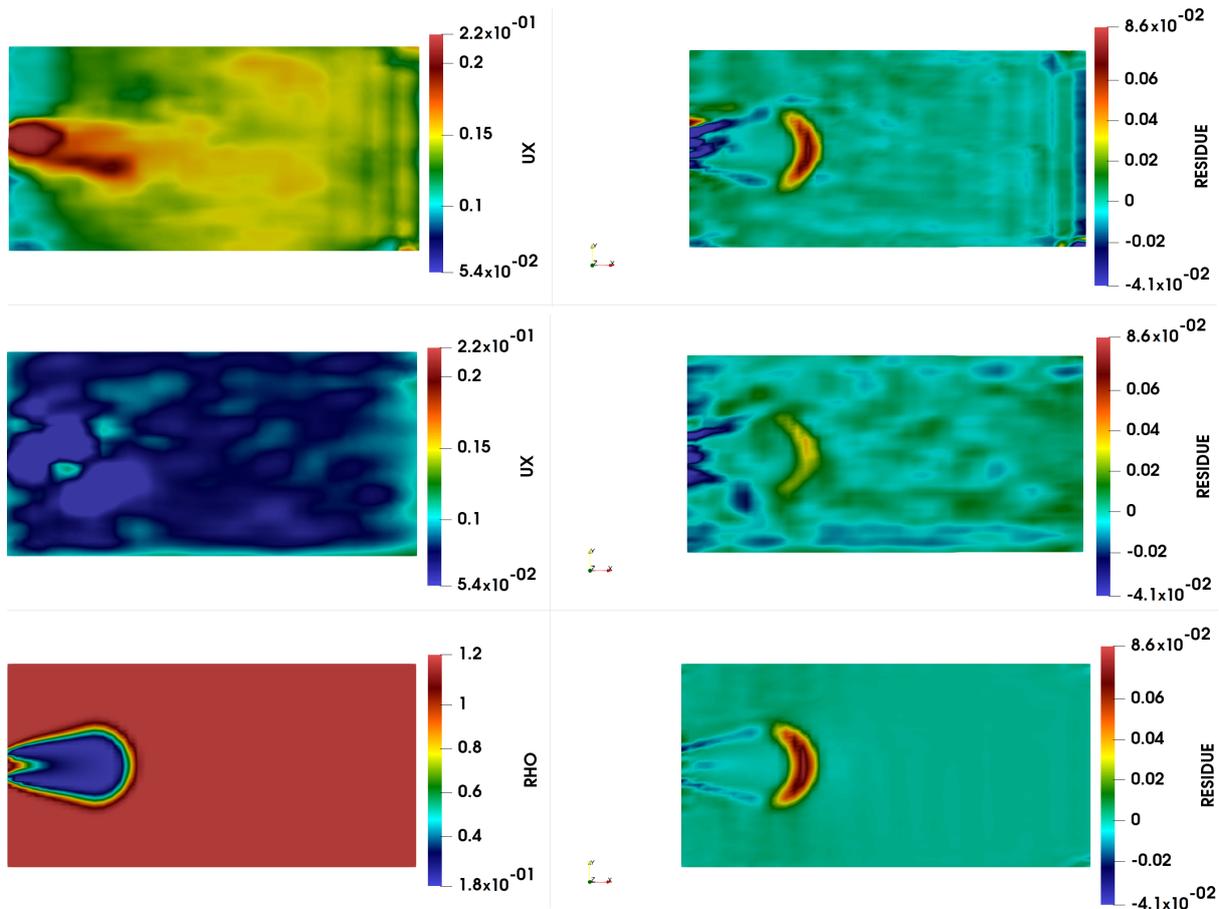


Figure 22. On the top left, the 805-th decoder generation $\mu_h(t) \sim p_\theta(\mu_h(t)|z)$ from a sample of the probability distribution $p_\theta(z)$ for the data-targeted VAE. On the top right, the residue of this generated sample, $\frac{\partial \rho_h(t_{805})}{\partial t} + \text{div}(\rho_h(t_{805})\mu_h(t))$. On the middle left, the 805-th decoder generation $\mu_h(t) \sim p_\theta(\mu_h(t)|z)$ from a sample of the probability distribution $N(0, 1)$ for the classical VAE. On the middle right, the residue of this generated sample, $\frac{\partial \rho_h(t_{805})}{\partial t} + \text{div}(\rho_h(t_{805})\mu_h(t))$. On the bottom left, the high-fidelity density field $\rho_h(t_{805})$ projected on the Cartesian mesh. On the bottom right, the accurate residue $\frac{\partial \rho_h(t_{805})}{\partial t} + \text{div}(\rho_h(t_{805})u_h(t_{805}))$.

5. Conclusions and Prospects

In this paper, we proposed a new framework for variational autoencoders in order to tackle a new data-targeted prior distribution without the need for the reparametrization trick. The approach gives encouraging results concerning the approximation of the true posterior of the observable training data over the random variable. In fact, we were able to impact the encoder’s posterior approximation thanks to a simple framework based on the backward differentiation of a squared error loss on the latent variable with respect to the KPOD orthogonal coefficients. The posterior approximation and its parameterized mean function can be interpreted with respect to a data-targeted prior. The associated numerical results illustrate that the generated fields by the decoder were plausible compared to the ones obtained by a VAE based on a standard normal prior. As motivated in the introduction of this paper, the future work consists of the direct application of this new methodology for the turbulence injection problem for large eddy simulations.

Author Contributions: Conceptualization, N.A.; methodology, N.A.; software, N.A., F.C.; validation, N.A., F.C. and T.D.; formal analysis, N.A.; investigation, N.A., F.C. and T.D.; resources, N.A.; data curation, N.A., F.C.; writing—original draft preparation, N.A.; writing—review and editing, F.C., T.D. and D.R.; visualization, N.A., F.C. and T.D.; supervision, D.R. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by Safran.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Similar data can be reproduced using fluid dynamics software.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chatelier, A.; Fiorina, B.; Moureau, V.; Bertier, N. Large Eddy simulation of a turbulent spray jet flame using filtered tabulated chemistry. *J. Combust.* **2020**, *2020*, 2764523. [[CrossRef](#)]
2. Akkari, N. A velocity potential preserving reduced order approach for the incompressible and unsteady Navier–Stokes equations. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020. [[CrossRef](#)]
3. Holmes, P.; Lumley, J.; Berkooz, G.; Rowley, C. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2012; pp. 358–377.
4. Sirovich, L. Turbulence and the dynamics of coherent structures. III. Dynamics and scaling. *Q. Appl. Math.* **1987**, *45*, 583–590. [[CrossRef](#)]
5. Lassila, T.; Manzoni, A.; Quarteroni, A.; Rozza, G. Model order reduction in fluid dynamics: Challenges and perspectives. In *Reduced Order Methods for Modeling and Computational Reduction*; Quarteroni, A., Rozza, G., Eds.; Springer: Berlin, Germany, 2014; Volume 9, pp. 235–274.
6. Balajewicz, M.; Tezaur, I.; Dowell, E. Minimal subspace rotation on the Stiefel manifold for stabilization and enhancement of projection-based reduced order models for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **2016**, *321*, 224–241. [[CrossRef](#)]
7. Baiges, J.; Codina, R.; Idelsohn, S. Reduced-order subscales for POD models. *Comput. Methods Appl. Mech. Eng.* **2015**, *291*, 173–196. [[CrossRef](#)]
8. Akkari, N.; Casenave, F.; Moureau, V. Time Stable Reduced Order Modeling by an Enhanced Reduced Order Basis of the Turbulent and Incompressible 3D Navier Stokes Equations. *Math. Comput. Appl.* **2019**, *24*, 45. [[CrossRef](#)]
9. Karatzas, E.N.; Stabile, G.; Nouveau, L.; Scovazzi, G.; Rozza, G. A reduced basis approach for PDEs on parametrized geometries based on the shifted boundary finite element method and application to a Stokes flow. *Comput. Methods Appl. Mech. Eng.* **2019**, *347*, 568–587. [[CrossRef](#)]
10. Hay, A.; Borggaard, J.; Akhtar, I.; Pelletier, D. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis. *J. Comput. Phys.* **2010**, *229*, 1327–1352. [[CrossRef](#)]
11. Wahba, G. *Spline Models for Observational Data*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1990. [[CrossRef](#)]
12. Williams, C.K.I. Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond. In *Learning in Graphical Models*; Jordan, M.I., Ed.; Springer: Dordrecht, The Netherlands, 1998; pp. 599–621. [[CrossRef](#)]
13. Kawai, S.; Shimoyama, K. Kriging-model-based uncertainty quantification in computational fluid dynamics. In Proceedings of the 32nd AIAA Applied Aerodynamics Conference, Atlanta, GA, USA, 16–20 June 2014; p. 2737.
14. Duchaine, F.; Morel, T.; Gicquel, L. Computational-fluid-dynamics-based kriging optimization tool for aeronautical combustion chambers. *AIAA J.* **2009**, *47*, 631–645. [[CrossRef](#)]
15. Margheri, L.; Sagaut, P. A hybrid anchored-ANOVA–POD/Kriging method for uncertainty quantification in unsteady high-fidelity CFD simulations. *J. Comput. Phys.* **2016**, *324*, 137–173. [[CrossRef](#)]
16. Gundersen, K.; Oleynik, A.; Blaser, N.; Alendal, G. Semi-conditional variational auto-encoder for flow reconstruction and uncertainty quantification from limited observations. *Phys. Fluids* **2021**, *33*, 017119. [[CrossRef](#)]
17. Kingma, D.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2014**, arXiv:1312.6114.
18. Jimenez Rezende, D.; Mohamed, S.; Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *arXiv* **2014**, arXiv:1401.4082.
19. Everson, R.; Sirovich, L. Karhunen Loeve procedure for gappy data. *J. Opt. Soc. Am. A* **1995**, *8*, 1657–1664. [[CrossRef](#)]
20. Bui-Thanh, T.; Damodaran, M.; Willcox, K. Aerodynamic data reconstruction and inverse design using Proper Orthogonal Decomposition. *AIAA J.* **2004**, *42*, 1505–1516. [[CrossRef](#)]
21. Liu, B.; Tang, J.; Huang, H.; Lu, X.Y. Deep learning methods for super-resolution reconstruction of turbulent flows. *Phys. Fluids* **2020**, *32*, 025105. [[CrossRef](#)]
22. Beck, A.; Flad, D.; Munz, C.D. Deep neural networks for data-driven LES closure models. *J. Comput. Phys.* **2019**, *398*, 108910. [[CrossRef](#)]
23. De Nayer, G.; Schmidt, S.; Wood, J.N.; Breuer, M. Enhanced injection method for synthetically generated turbulence within the flow domain of eddy-resolving simulations. *Comput. Math. Appl.* **2018**, *75*, 2338–2355. [[CrossRef](#)]
24. Metropolis, N.; Ulam, S. The Monte Carlo Method. *J. Am. Stat. Assoc.* **1949**, *44*, 335–341. [[CrossRef](#)]
25. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: <http://www.deeplearningbook.org> (accessed on 25 September 2021).
26. Xiang, M.; Zabarar, N. Kernel principal component analysis for stochastic input model generation. *J. Comput. Phys.* **2011**, *230*, 7311–7331.

27. Soize, C. Random matrix models and nonparametric method for uncertainty quantification. In *Handbook for Uncertainty Quantification*; Ghanem, R., Higdon, D., Owhadi, H., Eds.; Springer International Publishing: Cham, Switzerland, 2017; Volume 1, pp. 219–287. [[CrossRef](#)]
28. Farhat, C.; Tezaur, R.; Chapman, T.; Avery, P.; Soize, C. Feasible Probabilistic Learning Method for Model-Form Uncertainty Quantification in Vibration Analysis. *AIAA J.* **2019**, *57*, 1–14. [[CrossRef](#)]
29. Rezende, D.; Mohamed, S. Variational Inference with Normalizing Flows. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; Bach, F., Blei, D., Eds.; PMLR: Lille, France, 2015; Volume 37, pp. 1530–1538.
30. Caterini, A.; Doucet, A.; Sejdinovic, D. Hamiltonian Variational Auto-Encoder. *arXiv* **2018**, arXiv:1805.11328.
31. Neal, R. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*; CRC Press: Boca Raton, FL, USA, 2012. [[CrossRef](#)]
32. Rathi, Y.; Dambreville, S.; Tannenbaum, A. Statistical Shape Analysis using Kernel PCA. *Proc. SPIE—Int. Soc. Opt. Eng.* **2006**, *6064*. [[CrossRef](#)]
33. Kwok, J.Y.; Tsang, I.H. The pre-image problem in kernel methods. *IEEE Trans. Neural Netw.* **2004**, *15*, 1517–1525. [[CrossRef](#)] [[PubMed](#)]
34. Bakir, G.; Weston, J.; Schölkopf, B.; Thrun, S.; Saul, L. Learning to Find Pre-Images. *Adv. Neural Inf. Process. Syst.* **2004**, *16*, 449–456.
35. Mika, S.; Schölkopf, B.; Smola, A.; Müller, K.R.; Scholz, M.; Rätsch, G. Kernel PCA and De-Noising in Feature Spaces. In Proceedings of the 12th Annual Conference on Neural Information Processing Systems, NIPS 1998, Denver, CO, USA, 30 November–5 December 1998.
36. Partaourides, H.; Chatzis, S. Asymmetric Deep Generative Models. *Neurocomputing* **2017**, *241*, 90–96. [[CrossRef](#)]
37. Berger, V.; Sebag, M. Variational Auto-Encoder: Not all failures are equal. *hal-02497248* **2020**. Available online: <https://hal.inria.fr/hal-02497248> (accessed on 25 September 2021).
38. Amsallem, D.; Farhat, C. Stabilization of projection based reduced order models. *Int. J. Numer. Methods Eng.* **2012**, *91*, 358–377. [[CrossRef](#)]
39. Rowley, C.; Colonius, T.; Murray, R. Model Reduction for compressible flows using POD and Galerkin projection. *Phys. D Nonlinear Phenom.* **2004**, *189*, 115–129. [[CrossRef](#)]
40. Lee, K.; Carlberg, K. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.* **2020**, *404*, 108973. [[CrossRef](#)]
41. Moureau, V.; Domingo, P.; Vervisch, L. Design of a massively parallel CFD code for complex geometries. *Comptes Rendus Mécanique* **2011**, *339*, 141–148. [[CrossRef](#)]
42. Moureau, V.; Domingo, P.; Vervisch, L. From Large-Eddy Simulation to Direct Numerical Simulation of a lean premixed swirl flame: Filtered laminar flame-PDF modeling. *Combust. Flame* **2011**, *158*, 1340–1357. [[CrossRef](#)]
43. Malandain, M.; Maheu, N.; Moureau, V. Optimization of the deflated conjugate gradient algorithm for the solving of elliptic equations on massively parallel machines. *J. Comput. Phys.* **2013**, *238*, 32–47. [[CrossRef](#)]
44. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.