Data-Centric Engineering (2026), 7: e30
doi:10.1017/dce.2026.10069

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Multi-fidelity autoencoders: RANS-LES jet flow predictions

Ettore Saetta[1], Michele Massa[1], Renato Tognaccini[1] and Gianluca Iaccarino[2]

[1]Industrial Engineering, University of Naples Federico II, Italy.
[2]Department of Mechanical Engineering, Stanford University, USA.
Corresponding author: Ettore Saetta; Email: ettore.saetta@unina.it

Received: 13 November 2025 UTC; **Revised:** 18 June 2026 UTC; **Accepted:** 28 June 2026 UTC

Keywords: autoencoders; large eddy simulations; multi-fidelity; nozzle jet; uncertainty quantification

Abstract

In Fluid Mechanics, developing deep learning models for relevant engineering configurations using exclusively high-fidelity data is often impractical due to the large computational cost of the corresponding simulations. We introduce a Machine Learning procedure that combines three algorithms to construct a multi-fidelity autoencoder capable of providing real-time high-fidelity predictions with quantified uncertainties while substantially reducing the computational cost of database construction. We demonstrate the method on a problem of practical interest: the prediction of jet flow fields emanating from a family of parametric nozzle configurations. The proposed multi-fidelity dataset comprises a limited number of computationally expensive Large Eddy Simulations augmented by a larger ensemble of computationally cheap Reynolds-Averaged Navier–Stokes simulations. The trained model learned the underlying relation between nozzle geometry and jet properties from low-fidelity data, and used high-fidelity ones to construct a correction term from low to high fidelity, thus providing a high-fidelity reduced-order model in the full design space.

Impact Statement

High-fidelity datasets for deep learning surrogates in Fluid Mechanics are prohibitively expensive. We introduce a multi-fidelity training algorithm for autoencoders that relies on few high-fidelity samples enriched by many low-fidelity simulations to produce accurate surrogates (with quantified uncertainties) while reducing database construction cost by 20x.

1. Introduction

Fluid flow simulations for configurations of engineering interest remain computationally taxing when using high-fidelity approaches. For this reason, most of the current industrial applications rely on low-fidelity models (e.g., Reynolds-Averaged Navier–Stokes simulations—RANS), especially when many configurations need to be investigated (e.g., shape design optimization). However, in many cases of engineering interest, low-fidelity methods fail to provide accurate predictions, thus limiting the usefulness of the results.

Recent advances in Machine Learning (ML) have enabled data-driven surrogate models capable of accurately predicting aerodynamic performance in real time, greatly accelerating optimization processes. Nevertheless, in fields such as Fluid Mechanics, generating high-fidelity training datasets remains computationally expensive and time-consuming. In recent years, autoencoders (AEs) (Hinton and

This research article was awarded Open Data badge for transparent practices. See the Data Availability Statement for details.

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

Salakhutdinov, 2006) have become popular for the construction of reduced-order models. They are Deep Learning algorithms typically employed for dimensionality reduction and feature extraction. AEs can be viewed as a nonlinear extension of the Proper Orthogonal Decomposition (POD), reducing complex data into a low-dimensional compressed representation (latent space), with compression ratio much higher than that of POD (Milano and Koumoutsakos, 2002). The ability of AEs to make accurate predictions has been extensively discussed in recent literature, and many different approaches and architectures have been proposed (Agostini, 2020; Racca et al., 2023; Solera-Rico et al., 2024). Many articles discuss the problem of data collection for training AEs (and, in general, deep learning algorithms) in Fluid Dynamics. For instance, Tangsali et al. (2020) showed that, in order to construct an AE capable of generalizing across many different flow regimes past an airfoil, a training database composed of thousands of simulations was required, which would be unfeasible for high-fidelity methods such as Large Eddy Simulations (LES). Training strategies that rely on a mixture of low- and high-fidelity results (multi-fidelity) have the potential to enable accurate deep learning approaches at a manageable computational cost.

Multi-fidelity applications for AEs remain an area with limited coverage in the literature. For instance, Shen et al. (2024) proposed a multi-fidelity transfer learning method using AEs to efficiently construct a surrogate model for airfoil aerodynamics, employing Euler simulations (low fidelity) and RANS simulations on coarse grid (medium fidelity) and fine grid (high fidelity). In their approach, the objective is constructing a model capable of making high-fidelity predictions providing low-fidelity input. Partin et al. (2023) proposed a multi-fidelity network architecture composed of convolutional encoders, decoders, and skip connections, resulting in a model where different fidelities are learned simultaneously. In this case, sparse high-fidelity data are enriched with a large amount of low-fidelity data. They applied their method to classical benchmark problems: one-dimensional functional approximation and the solution of the pressure Poisson equation.

In the present article, we propose a method for constructing AE models able to provide high-fidelity predictions with quantified uncertainties at low computational cost (taking advantage of multi-fidelity methods) on a problem of practical engineering interest. Specifically, in the present article, we introduce a multi-branch AE to treat two-dimensional fields that are not spatially related but correspond to the same input parameters. Specifically, each dataset case comprises channels containing physical variables from different flow sections; we introduce a multi-fidelity training algorithm for AEs that enables a drastic reduction in the computational cost of database construction. The novelty of the multi-fidelity method lies in the fact that multi-fidelity papers in the literature typically refer to a coarse/fine solution of the same problem and introduce a relatively complex architecture that enables merging of the datasets. Here, the approach is general, that is, the data can be from completely different descriptions of the solution (including experiments, provided that the low-fidelity data are pre-processed to mimic the experimental data), and the data fusion is based on the idea of latent space enrichment. Finally, we applied an ensemble technique to provide predictions with quantified model uncertainties. The proposed method is applied to the prediction of jet flows exhausting from nozzles of different geometries. The multi-fidelity dataset is constructed using RANS simulations for the low-fidelity data and LES for the high-fidelity data.

Section 2 focuses on the comparison between RANS and LES solutions and experimental results available in the literature, highlighting the necessity of high-fidelity calculations. Section 3 describes the computational setup used for constructing the training database (Saetta et al., 2025) and the parametric description of the nozzle geometries. Section 4 is dedicated to describing the ML algorithms and methods employed for training a multi-fidelity AE and for making flow predictions with quantified uncertainties. In Section 5, the results of the trained model are discussed, first presenting a practical example of the encoding and decoding procedures, and then assessing the accuracy of the model predictions against CFD reference simulations.

2. Validation of flow simulations

The validation of the CFD simulations used to construct the database was carried out by comparing the numerical results with two available experiments: a hot round jet (Bridges and Wernet, 2011) and a rectangular jet (Zaman, 2012).

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

2.1. Hot round jet

The open-source software SU2 (Economon et al., 2016), was used to perform RANS simulations, while the CharLES software (Goc et al., 2021) was used for the Wall Modeled Large Eddy Simulations (WM-LES). Two computational grids were used for both RANS and LES to analyze the grid convergence. For RANS, the grids were taken from the NASA Turbulence Modeling Resource (https://turbmodels.larc.nasa.gov/jetsubsonichot_grids.html), and they had resolution of 18×10^3 and 72×10^3 cells. For the LES, the grids were generated using the CharLES grid generator and had resolution of 11×10^6 and 30×10^6 cells.

The nozzle exit Reynolds number, non-dimensionalized by the exit diameter, is 6.15×10^5, while the nozzle exit Mach number is 0.376. The k–ω SST turbulence model described by Menter (1994) was employed for the steady RANS simulations, while the subgrid model proposed by Vreman (2004) was used for the LES. The computation of the statistics in the LES was performed after an initial transient of $\overline{t_{tr}} = t_{tr} U_{id}/D = 203$ over a simulation time of $\overline{t_{sim}} = t_{sim} U_{id}/D = 1354$, with $\overline{\Delta t} = \Delta t U_{id}/D = 6.77 \times 10^{-4}$, in which t_{tr} is the transient time, U_{id} is the jet velocity at the nozzle exit assuming fully expanded flow, D is the exit diameter, t_{sim} is the simulation time, and Δt is the simulation time step. The time step was chosen such that the simulation was carried out at $CFL \approx 2.0$. A co-flow at $M_\infty = 0.01$ was used outside the nozzle in both simulations.

The grid convergence and the comparison with the experiment of the mean streamwise velocity at the jet centerline are shown in Figure 1.

The RANS simulation significantly overpredicts the mean velocity near the nozzle exit, the region commonly referred to as the potential core. Meanwhile, across the entire axis, the LES show a more convincing agreement.

2.2. Rectangular jet

The capability of the LES to capture the flow features of rectangular jets was verified analyzing the jet with aspect ratio $AR = 8$ studied by Zaman (2012). For this analysis, only the LES were performed, with a grid generated using the CharLES grid generator composed of ≈ 30 million cells. The nozzle exit Mach and Reynolds numbers were 0.226 and 2.76×10^5, respectively. In this analysis, statistics were computed after an initial transient of $\overline{t_{tr}} = 96$ over a simulation time of $\overline{t_{sim}} = 640$, with a time step of $\overline{\Delta t} = 3.20 \times 10^{-4}$, where D_{eq} is the nozzle exit equivalent diameter. The mean streamwise velocity at the jet centerline, as obtained by the LES and the experiment, is shown in Figure 2.

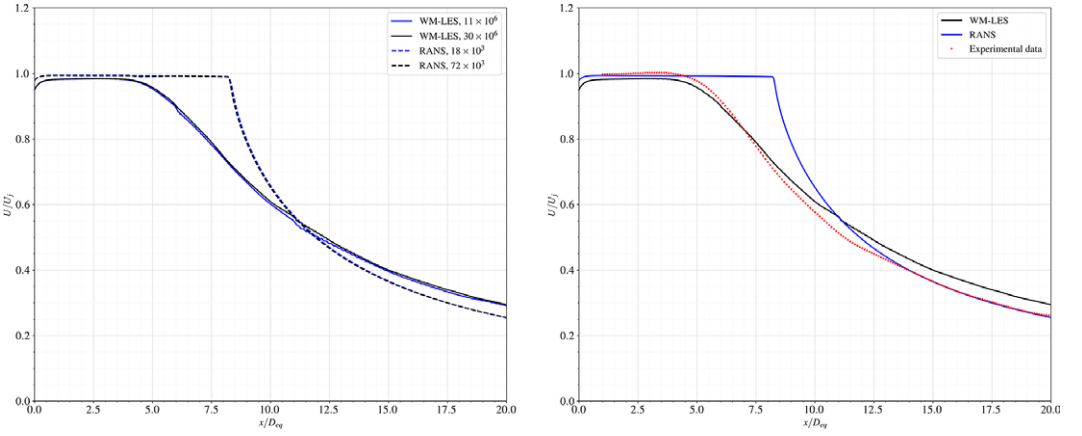


Figure 1. *Grid convergence (left) and comparison with the experiment (right) of the mean streamwise velocity at the jet centerline by CharLES and SU2 for the hot round jet.*

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

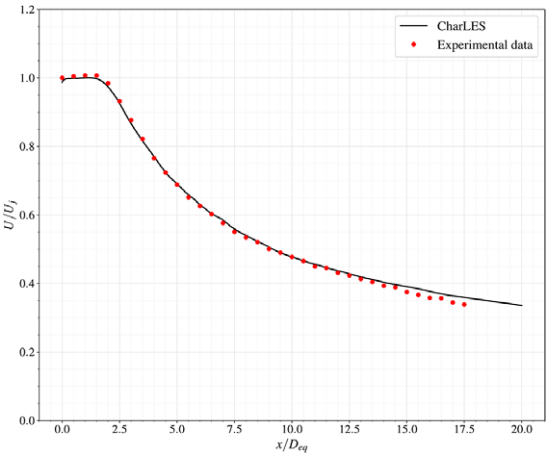


Figure 2. *Comparison of the mean streamwise velocity at the jet centerline obtained by the CharLES and the experiment for the rectangular jet.*

The agreement with the experiment further validated the LES methodology. Due to the highlighted limitations of the RANS model in capturing the jet potential core, in the following RANS will be referred to as low fidelity (LF), while LES will be referred to as high fidelity (HF).

3. Nozzle jet database

3.1. Parametric description

The adopted reference system is shown in Figure 3 for a sample nozzle.

The nozzle cross-section is described by a superellipse (Davis and Gessner, 1992). A database of nozzles (described in more detail in Section 5.1) was generated based on the following geometric design parameters:

- f_A^*: cross-section area distribution. A low f_A^* indicates that area variations are concentrated near the nozzle inlet, whereas a high f_A^* indicates that area variations are concentrated near the nozzle exit.
- AR_{out}: exit aspect ratio (width over height).
- n_{out}: exit superellipse factor. With $n_{out} = 2$, the exit is an ellipse, while increasing n_{out} makes it progressively more rectangular.

All nozzles maintain a circular inlet with the same area, while also keeping the same length L and outlet area. The analytical laws used to define the geometry are omitted for the sake of brevity. The influence of the parameters on the nozzle shape is shown in Figure 4.

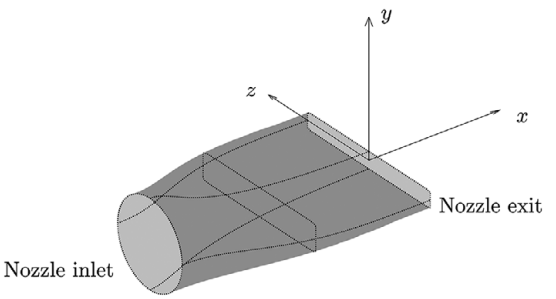


Figure 3. *A sample of nozzle database.*

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

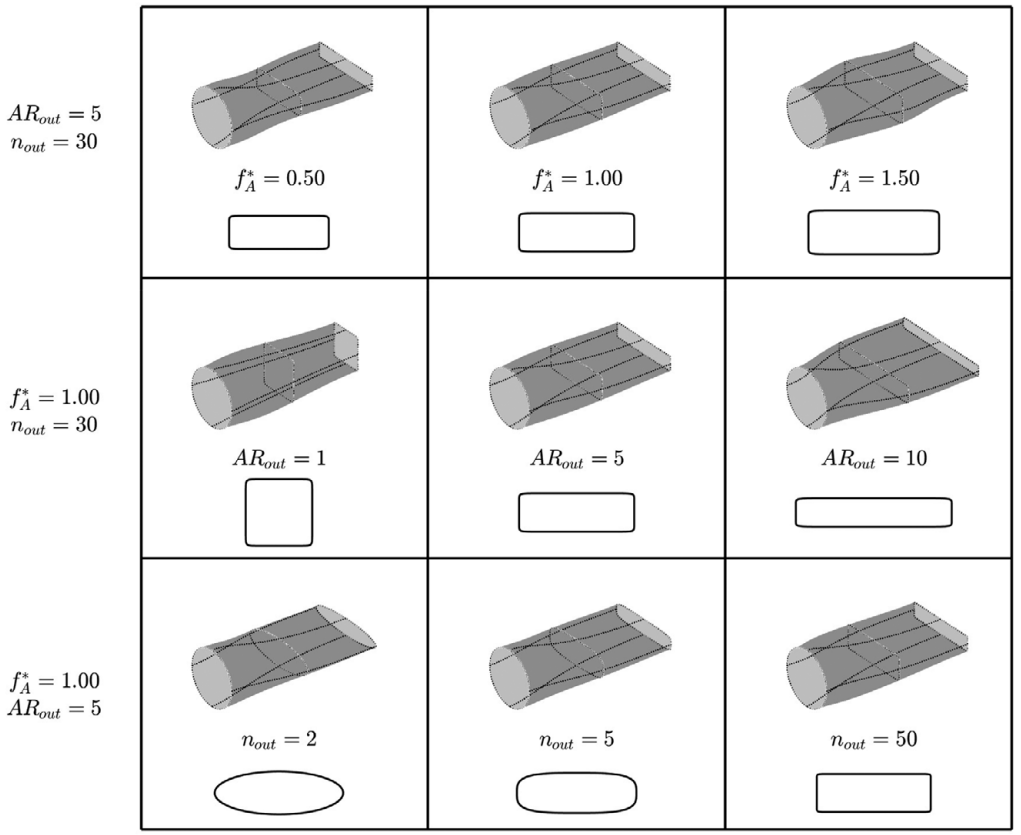


Figure 4. *Isometric view and section at $x/L = -0.5$ of sample geometries at different f_A^* (top), AR_{out} (middle), and n_{out} (bottom).*

3.2. Computational setup

Turbulence is deeply affected by nozzle geometric variations, because the jet exit velocity distribution determines the initial condition of the jet shear layers and its subsequent mixing. In addition, specific turbulence modeling choices (RANS vs. LES) induce differences in the results, with the RANS models being unable to represent the critical physical processes. RANS simulations fail to capture the generation of secondary flows in rectangular nozzles, which arise from the anisotropy of Reynolds stress (Bradshaw, 1987). Moreover, the breakdown of the coherent structures occurring in the jet can not be accurately captured by the present steady RANS model. On the other hand, LES can accurately capture stress-induced secondary flows and the instantaneous behavior of the coherent structures, and is therefore referred to as a high-fidelity physical representation of the nozzle jet flow. The LES resolve the instantaneous flow field, which is not symmetric even in a geometrically symmetric configuration. Meanwhile, the RANS equations solve the statistically averaged flow field, and the mean flow is symmetric under symmetric geometry and boundary conditions. Therefore, the flow resolved in the RANS simulation was defined on a quarter of the domain, while the flow resolved in the LES was defined as the full 360 degree field. In both cases, the highest refinement is near the walls and inside the jet shear layer.

The computational grids used for the RANS simulations were generated by an automatic multi-block structured grid generator developed in-house, while the grids used for the LES were generated following the same methodology used for the hot round jet and rectangular jet. A grid convergence analysis was carried out for both low- and high-fidelity simulations on a sample geometry ($f_A^* = 1.00$, $AR_{out} = 10$,

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

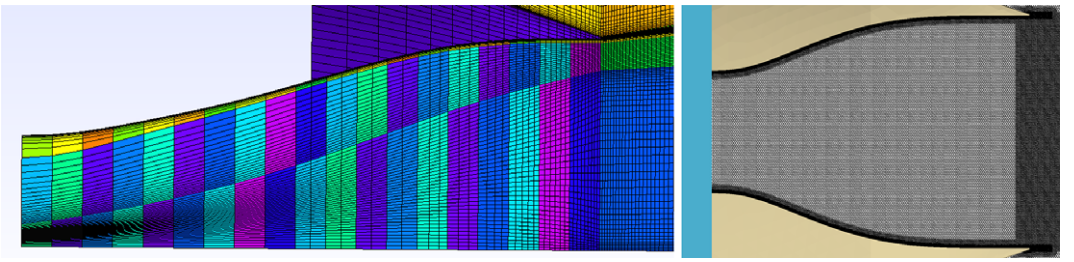


Figure 5. *Fine computational grid used for RANS (left) and LES (right) for the geometry* $f_A^* = 1.00$, $AR_{out} = 10$, $n_{out} = 30$.

$n_{out} = 30$), ensuring no influence of the grid resolution on the nozzle jet physics. The grid resolutions were $\approx 700 \times 10^3$ (fine) and $\approx 260 \times 10^3$ (coarse) for RANS, and $\approx 40 \times 10^6$ (fine) and $\approx 8 \times 10^6$ cells (coarse) for LES. Figure 5 shows the fine grids, and Figure 6 shows the streamwise velocity along the jet centerline for both low and high fidelity. The results were nearly equal, and the fine grids were chosen for both fidelities due to the higher quality of the flow field.

The nozzle inlet total pressure and temperature were set to $p_0/p_\infty = 1.79$ and $T_0/T_\infty = 2.69$ where p_∞ and T_∞ denote the ambient pressure and temperature. The nozzle exit Mach number is 0.953, and the nozzle exit Reynolds number, non-dimensionalized using the equivalent diameter, is 6.44×10^6. For each simulation, the following global metrics were computed:

- Potential core length L_{pc}.
- Half-jet spreading $y_{1/2}$ and $z_{1/2}$ at $x/D_{eq} = 2$.

In the present application, the database is composed of tensors obtained by interpolating the CFD data onto structured Cartesian grids. The channel groups are:

1. $\mathbf{b}_1$: x, z grid coordinates, U_x, U_z velocity components in the $x-z$ plane at $y/D_{eq} = 0$. The structured grid extended in $x \in [-L, 20D_{eq}]$ in the streamwise direction, while up to the half-width coordinate (see Mi et al., 2005) in the spanwise direction.
2. $\mathbf{b}_2$: y, z grid coordinates, U_x, U_z velocity components in the $y-z$ plane at $x/D_{eq} = 1$. The grid was rectangular in the transverse direction and was extended from the jet half-widths coordinates of the minor half-width in both directions.

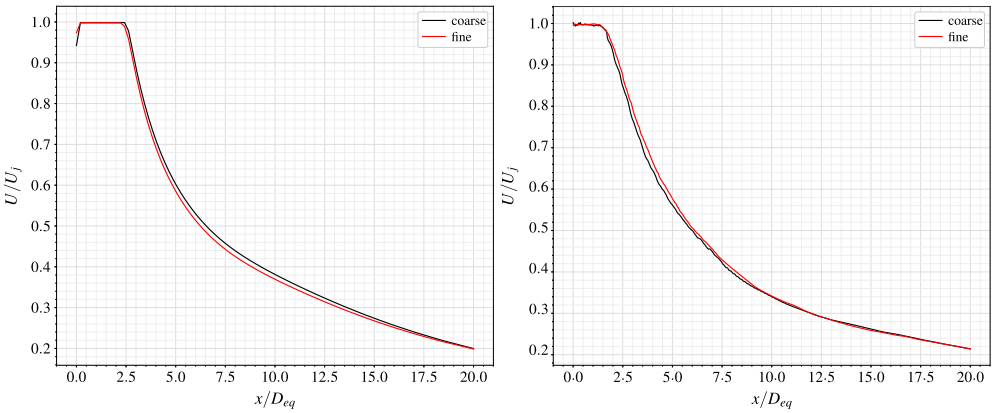


Figure 6. *Streamwise velocity along the jet centerline of coarse and fine grid for RANS (left) and LES (right) for the geometry* $f_A^* = 1.00$, $AR_{out} = 10$, $n_{out} = 30$.

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

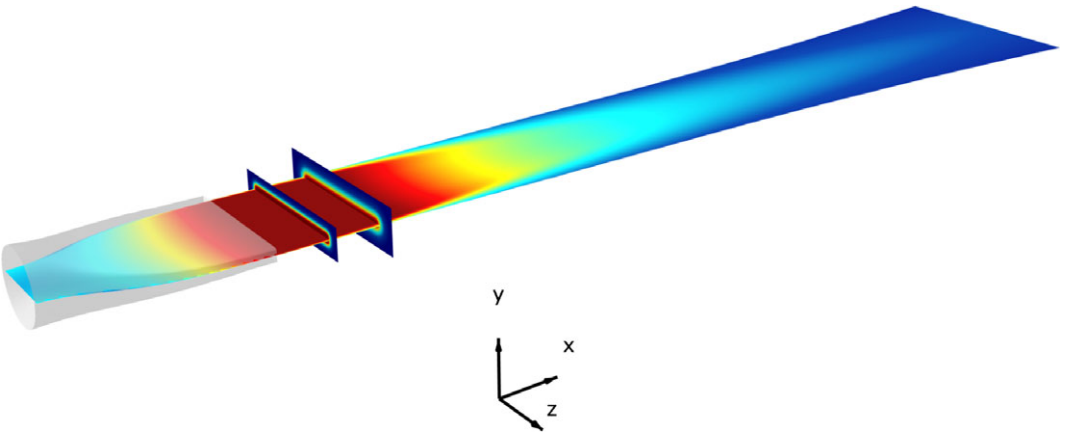


Figure 7. *Contour map of streamwise velocity for a sample nozzle.*

3. $\mathbf{b}_3$: y, z grid coordinates, U_x, U_z velocity components in the $y-z$ plane at $x/D_{eq}=2$. The grid definition methodology was the same as the one for $x/D_{eq}=1$.

A jet flow for a sample nozzle used as input to the AE is shown in Figure 7.

The RANS simulations were carried out using Intel Xeon Gold 6230R processors, with a mean time-to-solution of approximately 7 hours on 16 CPUs. The LES simulations were performed using A100 GPUs on the GPU-accelerated partition of Leonardo's exascale system for HPC and AI applications (Turisini et al., 2023). Flow statistics were computed after an initial transient of $\overline{t_{tr}}=200$ over $\overline{t_{sim}}=100$, with a time step of $\overline{\Delta t}=1.00\times10^{-3}$, resulting in a mean time-to-solution of approximately 3 hours on 16 GPUs. While we acknowledge that this simulation time is not sufficient to reach full statistical convergence, this choice will be justified in Section 5.3.

The computational cost per simulation for RANS (LF) and LES (HF) is shown in Table 1. The computational cost of a fully statistically converged LES, with $\overline{t_{sim}}=2000$, is also reported. The RANS simulations were carried out on 16 CPUs, while the LES on 16 GPUs.

3.3. Impact of geometry variation

Many authors have analyzed the effects of geometry on jet flows, including Tsuchiya et al. (1986), Mi and Nathan (2009), Zaman (2012). The results found in the present work are consistent with those reported in the literature. However, it is important to consider the impact of the geometrical parameters in the underlying physics, which is briefly discussed here. Figure 8 shows the variation of the streamwise velocity fields with respect to the input parameters. Each parameter affects different zones of the flow field: the cross-section area distribution affects the flow field inside the nozzle; the aspect ratio mostly impacts the jet; and the superellipse factor influences the transverse cross-section of the jet. The distribution of the cross-section area influences the location of higher gradients in the nozzle, thereby impacting the thrust, back pressure, and flow development. The exit aspect ratio significantly affects the

Table 1. *Computational cost of single low- and high-fidelity simulations*

	Device	Cost
LF	CPU Intel Xeon Gold 6230R	122 CPUh
HF ($\overline{t_{sim}}=100$)	GPU NVIDIA A100	45.7 GPUh
HF ($\overline{t_{sim}}=2000$)	GPU NVIDIA A100	335 GPUh

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

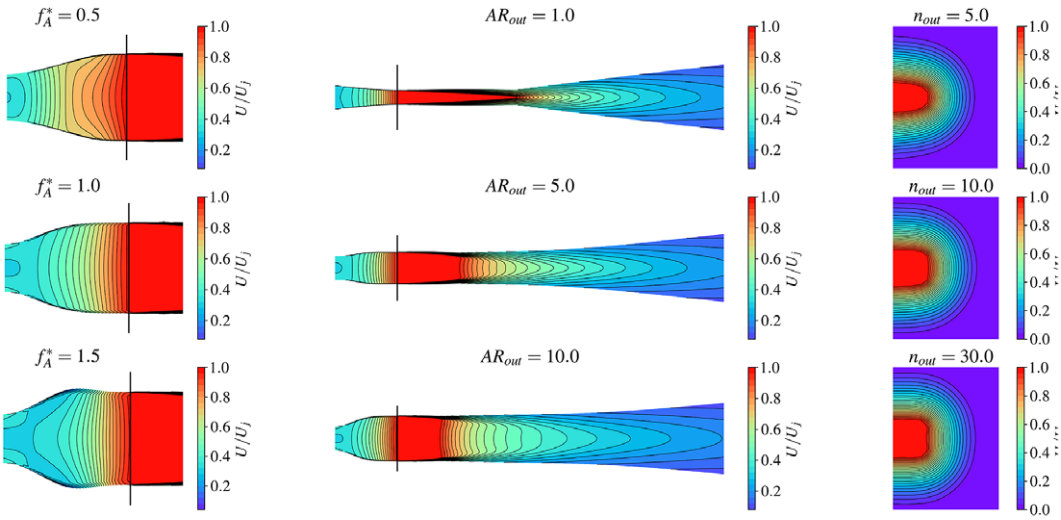


Figure 8. *Effects of cross-section area distribution (left), aspect ratio (middle), and superellipse factor (right) on the flow field. The black lines denote the nozzle exit.*

jet flow field, with the velocity decaying faster at higher aspect ratios due to the stronger coherent structures forming at the nozzle exit. Finally, the superellipse factor induces different velocity profiles in the jet cross-section, with the profile being more rounded at lower superellipse factors. This effect tends to disappear further away from the nozzle exit due to the stronger influence of the jet shear layers and turbulent mixing (Zaman, 1996). Although the aspect ratio is the most dominant parameter in the underlying physics, variations in the cross-section area distribution and the superellipse factor induce changes in the flow field significant in design optimization studies. Therefore, the ability of the AE to capture both large-scale and subtle variations in the underlying physics represents a significant advantage.

4. The Machine Learning algorithm

The ML algorithm employed combines AE architectures and training methods for multi-fidelity predictions with quantified uncertainties. Specifically, an AE **baseline** architecture is introduced (the Multi-Branch Autoencoder—MBA). This AE is then trained using a multi-fidelity algorithm for AEs (AutoEncoder with Sequential Training and Adaptive Resolution—AESTAR); finally, an ensemble of trained AEs is constructed to provide predictions with quantified uncertainties. The three algorithms are briefly described below.

4.1. Autoencoder baseline: multi-branch autoencoder

The proposed AE has a multi-branch structure and will be denoted as MBA. Due to the use of convolutional layers, the dataset must be converted first to a tensor of shape $N \times C \times I \times J$ (cases $\times$ channels $\times$ i-grid elements $\times$ j-grid elements). The channels of the training dataset are the U_x and U_z velocity components in different planes: x-z plane and y-z planes. Because the channels belonging to different planes contain information that is not spatially related, a multi-branch structure is required. However, they share the same global information about the design parameters of the dataset. For this reason, multiple convolutional branches are considered, all merging in a shared latent vector. Figure 9 describes the multi-branch architecture proposed here. In the figure, each tensor (case) in the database can be divided into three groups of channels (green, blue, and red) $\mathbf{x} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$. Each channel group is passed through a different encoder function obtaining three latent vectors: $\overline{\mathbf{V}}_i = \mathbf{e}_i(\mathbf{b}_i)$. The latent vectors coming from different branches are projected into one shared latent vector $\mathbf{V}$ (in purple) using a fully

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

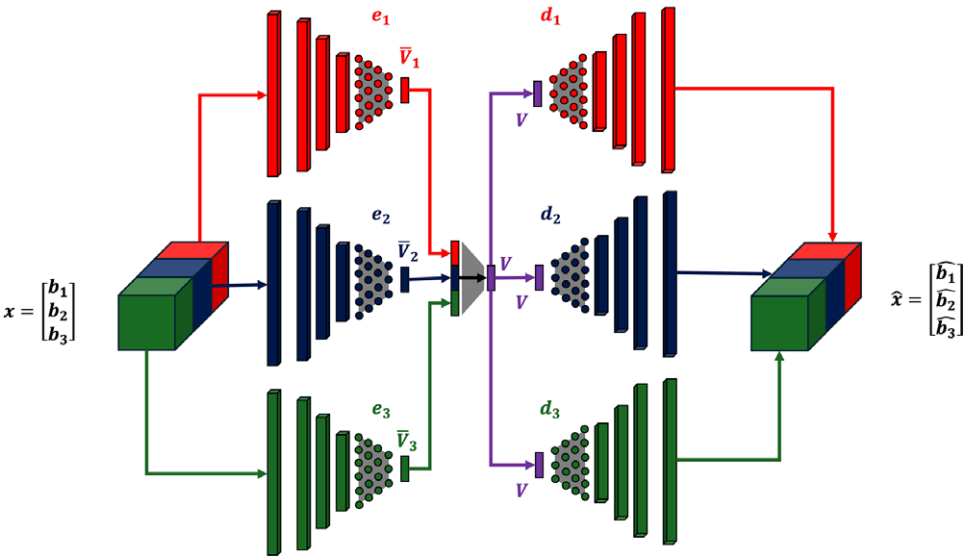


Figure 9. *Schematic representation of the Multi-Branch Autoencoder (MBA) architecture.*

connected network. The projected latent space is shared across different decoders to reconstruct the input tensor: $\widehat{\mathbf{b}}_i = \mathbf{d}_i(\mathbf{V})$.

4.2. The multi-fidelity training algorithm

For the treatment of multi-fidelity databases composed of simulations obtained with different physical models (e.g., Euler, RANS, DES, LES, etc.), we introduce the algorithm called AESTAR. It is a training algorithm for AEs naturally suited for multi-fidelity databases, based on a sequential training of AEs and an unsupervised conditioning of the latent space to construct successive latent representations. In the literature, latent space *conditioning* is typically achieved by introducing an additional term in the loss function, designed to enforce specific constraints on the latent variables. In other words, it guides the latent space toward a desired topological structure. Li et al. (2022) introduced a physics-based conditioning of the latent space, while a specific geometrical topology was enforced in An and Jeon (2024). This practice introduces additional information (bias) into the latent space during training. In contrast, the conditioning method proposed in the present work leverages knowledge extracted from the preceding training steps, and therefore operates without any *a priori* bias. The conditioning involves only the latent subspaces of the current and previous training instances. Consequently, the algorithm can be regarded as an *unsupervised* latent space conditioning technique.

The algorithm scheme is reported in Figure 10. $D^{(1)}$ and $D^{(2)}$ are, respectively, low- and high-fidelity databases. First, an AE^1 with a latent space dimension of p is trained on $D^{(1)}$. Then, in a second step, another AE^2 is trained with a database composed by $D^{(1)} \cup D^{(2)}$ and a latent space dimension of $P > p$. In the first step, the loss function is defined solely as the error between the input and its reconstruction. In the second step, the latent space conditioning term ($C(\mathbf{V})$) is added, thus the loss function employed becomes:

$$\mathcal{L}_{\text{AESTAR}} = \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) + \delta C(\boldsymbol{V}) \tag{4.1}$$

where $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$ is the loss term relative to the reconstruction error, x and $\hat{x}$ are the input data and its reconstruction, $\mathbf{V}$ is the latent vector, and δ is a weight introduced to balance the relative importance between the two terms in the loss function. $C(\boldsymbol{V})$ is the latent space conditioning term:

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

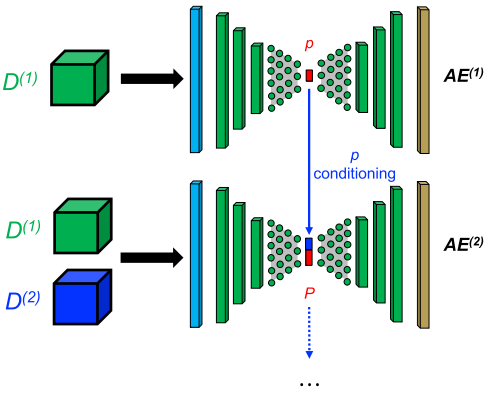


Figure 10. *Schematic representation of the training and architecture of the AESTAR algorithm for adaptive resolution databases.*

$$C(\mathbf{V}) = \begin{cases} \sum_{i=1}^{p} \left(V_i^{(I)} - V_i^{(I-1)} \right)^2, & \text{for D}^{(\text{I})} \\ \sum_{i=1}^{p} \left(V_i^{(I)} - V_i^{(I-1)} \right)^2 + \sum_{i=p+1}^{p} \left(V_i^{(I)} \right)^2, & \text{for D}^{(\text{I}-1)} \end{cases} \tag{4.2}$$

where I denotes the training step, V_i is the i-th latent variable, and p is the number of latent variables up to the I-th training step. The distinction between the formulations for databases $D^{(I)}$ and $D^{(I-1)}$ arise from the fact that the additional latent variables are left free to be learned in the higher-fidelity cases, whereas they are conditioned to zero for the lower-fidelity ones. This preserves consistency with the latent space identified at the previous fidelity level, where such additional latent variables were absent. In the present work, $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$ is

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N} \left(\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + a_1 \|\nabla \mathbf{x} - \nabla \hat{\mathbf{x}}\|_2^2 + a_2 \left\| \nabla^2 \mathbf{x} - \nabla^2 \hat{\mathbf{x}} \right\|_2^2 \right) \tag{4.3}$$

where N is the number of elements over which the loss is evaluated, the first term is the Mean Squared Error, ∇ is the 2D discrete gradient, and ∇^2 is the 2D discrete Laplacian. The former computed using convolutions with kernels $G_x = [-1 \ 1]$ along the width and $G_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ along the height; and the latter using the kernel $L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$. The coefficients a_i are hyper-parameters that weight the relative importance of the loss terms. The spatial regularization terms are based solely on the pixel representation of the fields. Meaning that, during training, the first and second derivatives are not computed in the physical space (x and y coordinates), but in the pixel space. For this reason, these terms can be used for any channel in the input data, even when information about spatial coordinates is unavailable. In our experiments, predictions exhibit some non-physical oscillations when using only the MSE. While both MSE and MSE + regularization losses capture the overall behavior, those computed using only the MSE are noisy.

This composition of the loss function, with spatial regularization terms, is inspired by other works in literature which employ gradient and Laplacian loss terms for enhance super-resolution techniques in convolutional neural networks (Mathieu et al., 2016; Lin et al., 2024).

This algorithm is motivated by the fact that, in fields such as Fluid Mechanics, the construction of high-fidelity databases can be computationally prohibitive. For this reason, we introduced the present algorithm to train a surrogate model capable of delivering high-fidelity predictions by exploiting multi-

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

fidelity datasets composed of many low-fidelity (inexpensive) and few high-fidelity (expensive) simulations. In the first training step, an initial AE is built to predict low-fidelity data. The second step is aimed at learning a correction term that maps low-fidelity information to high-fidelity data within the latent space. This stage is performed by initializing the AE weights with those learned in the previous step. The conditioning term then establishes a soft correspondence between low- and high-fidelity data associated with the same parameters.

Each successive phase of training can rely on a dataset with progressively higher-fidelity data. The resulting algorithm provides adaptive resolution, in the sense that the same trained AE can return reconstructions selectively associated with a specific fidelity level. Furthermore, the sequential increase in latent space dimension introduces information that can be directly attributed to the increased fidelity of the dataset. Beyond multi-fidelity training, the proposed procedure also enables the construction of datasets that sequentially incorporate additional physical processes, while clearly separating the contribution of each subset. For linear problems, this is analogous to a superposition principle. Since the AESTAR algorithm is a training strategy rather than an architecture, it can be naturally applied to any AE architecture.

4.3. Predictions with quantified uncertainties

To assess the confidence in the multi-fidelity AE predictions we use an *ensemble technique*, by constructing an ensemble of AEs and computing mean and standard deviation of their predictions (Saetta et al., 2024). In the present work, an ensemble of multi-fidelity MBAs is constructed. Following the guidelines of the previous article of Saetta et al. (2024), each model in the ensemble has randomly initialized trainable parameters, different AE's architectures (varying kernel sizes, kernel strides, number of convolutions, dropout probability, number of trainable parameters) and different random selection of subsets for training and validation sets. In this work, only model-form uncertainty is taken into account. Figure 11 shows a schematic representation of how the method works when using the decoder as generative network: latent variables are provided as input to an ensemble of N trained decoders, whose N outputs are averaged to compute the mean and the standard deviation of the predictions. Details of this method can be found in Saetta et al. (2024).

5. Nozzle jet flow: multi-fidelity MBA predictions with quantified uncertainties

The MBA, the AESTAR training algorithm, and the ensemble technique for uncertainty quantification (described in the previous section) are all methods that can be combined. Therefore, the results presented here are obtained using an ensemble of MBAs trained with the multi-fidelity AESTAR.

5.1. Training database

Once the CFD simulations were validated, a multi-fidelity database was constructed with the goal of combining many low-fidelity simulations (RANS) with a few high-fidelity ones (LES) (Saetta et al., 2025).

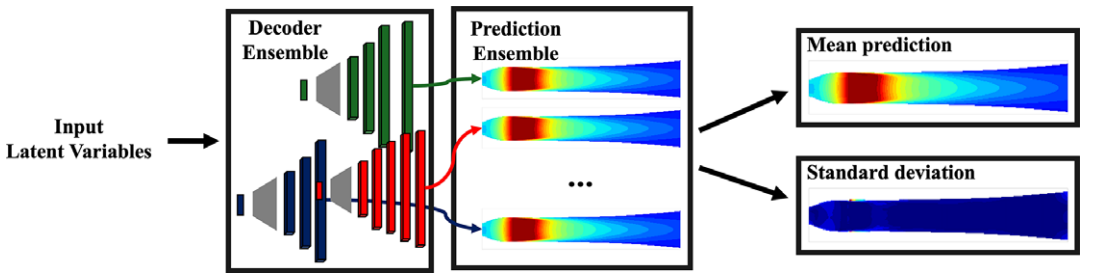


Figure 11. *Schematic representation of the ensemble technique of AE predictions for the quantification of model uncertainties.*

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

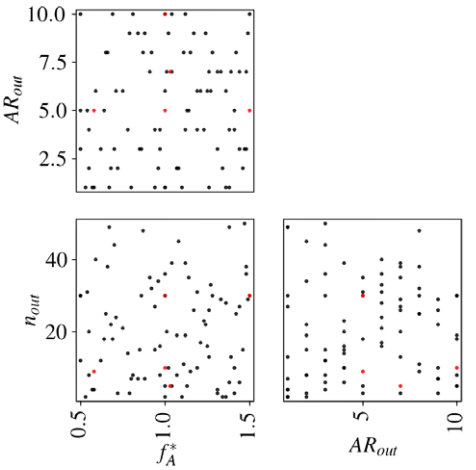


Figure 12. *Design parameter space of the multi-fidelity database. Black dots: RANS simulations; Red dots: LES.*

For this purpose, the database cases are selected using the Latin Hypercube Sampling (LHS, McKay et al., 1979) in the design parameter space f_A^*-AR_{out}-n_{out} as described in Section 3.1. For the parameters f_A^* and AR_{out}, a uniform distribution was used in the sampling. Meanwhile, as shown in Figure 4, the parameter n_{out} has a nonlinear effect on the nozzle surface shape; for example, the cross-section at $x/L = -0.5$ for $n_{out} = 5$ already resembles a rectangle more than an ellipse. For this reason, the sampling density for this parameter is higher at $n_{out} = 2$, and decreases as n_{out} increases. The final database consisted of 90 RANS simulations and 20 LES. Figure 12 shows the analyzed geometries in the design space, including both RANS and LES solutions.

To further reduce the computational cost of the high-fidelity database, a simulation time of only $\overline{t_{sim}} = 100$ were used to compute the time averages after the initial transient. On the contrary, as discussed in Section 3.2, a reasonable level of convergence for the time statistics requires a simulation time of $\overline{t_{sim}} = 2000$. This choice is further discussed in the results Section 5.3.

5.2. Multi-fidelity MBA ensemble

For the nozzle jet flow application, we trained the MBA using the AESTAR multi-fidelity method with two levels of data fidelity (i.e., two training steps):

- Step 1: RANS dataset with three latent variables. Since the number of independent design parameters is 3, training requires *at least* three latent variables.
- Step 2: RANS + LES dataset, adding two latent variables (five total latent variables).

A full sensitivity analysis of the latent space dimension is reported in Appendix A. An ensemble of 50 multi-fidelity MBAs is built in order to make multi-fidelity predictions with quantified uncertainties. Table 2 shows the training architecture variability to build the AE ensemble. Kernel size and stride for the i and j directions are selected independently. Additionally, they are selected independently for each convolutional layer, thus increasing the diversity of the architecture design space. For the architecture selection, some checks are provided to discard cases with kernel strides greater than size, and those which decoder output dimensions do not match the encoder inputs. In addition, for each item in the ensemble, the training database is randomly shuffled and the weights are randomly initialized, thus taking into account

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

Table 2. *Variability of the architecture ensemble*

Parameter	Range
Number of convolutions	4–8
Kernel size	2–5
Kernel stride	1–3
Max pooling size	2–4
Max pooling stride	2–3
Drop out probability	0%–25%

Table 3. *Training hyper-parameters*

Hyper-parameter	Value
Batch size	8
Validation set	15%
Optimizer	AdamW (Loshchilov and Hutter, 2019)
Learning rate (η)	5×10^{-4}
Learning rate policy	*Reduce on Plateau* (Paszke et al., 2019)
δ (of Eq.(4.1))	1.5

also the variability with batch construction and training initialization. A detailed analysis of the ensemble dimension is provided in Appendix B, where the choice of 50 MBAs to construct the ensemble is justified.

Some of the most important training hyper-parameters are reported in Table 3. Figure 13 shows the loss convergence for the two training steps. Figure 13(a) shows a smooth convergence behavior for the low-fidelity training step. The case RANS + LES is shown in Figure 13(b); it exhibits more noise, but still demonstrates steady convergence. The difference in magnitude of the two losses is simply due to the different definition of the loss function in the two steps (see description of Eq. (4.1)).

5.2.1. Encoding

Figure 14 shows the latent space obtained by encoding the multi-fidelity database. The black dots correspond to the low-fidelity data, while the red dots represent the high-fidelity data. A first observation is that for the RANS cases V_1-V_2-V_3 (the first latent subspace) is active, while V_4-V_5 is not (all zero). In contrast, for the LES encoding, all latent variables are active. In V_1-V_2-V_3, the red dots overlap with the

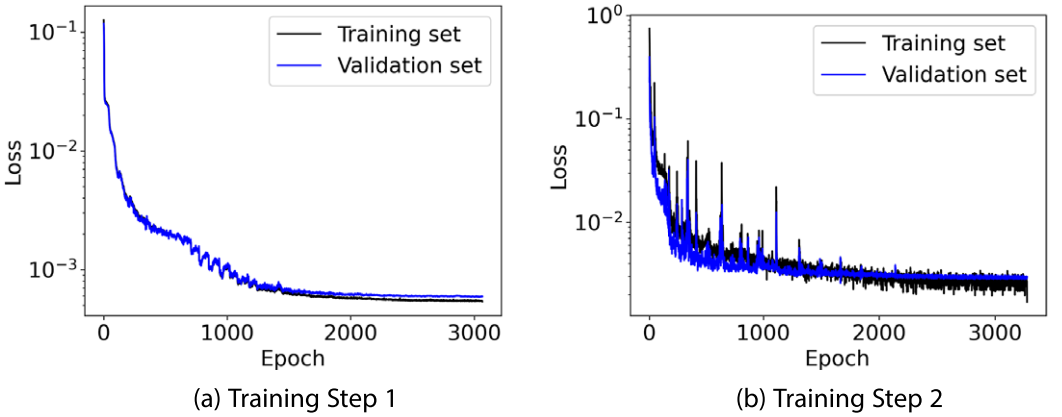


Figure 13. *Loss convergence for the two training steps. (a) step 1, RANS; (b) step 2, RANS+LES.*

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

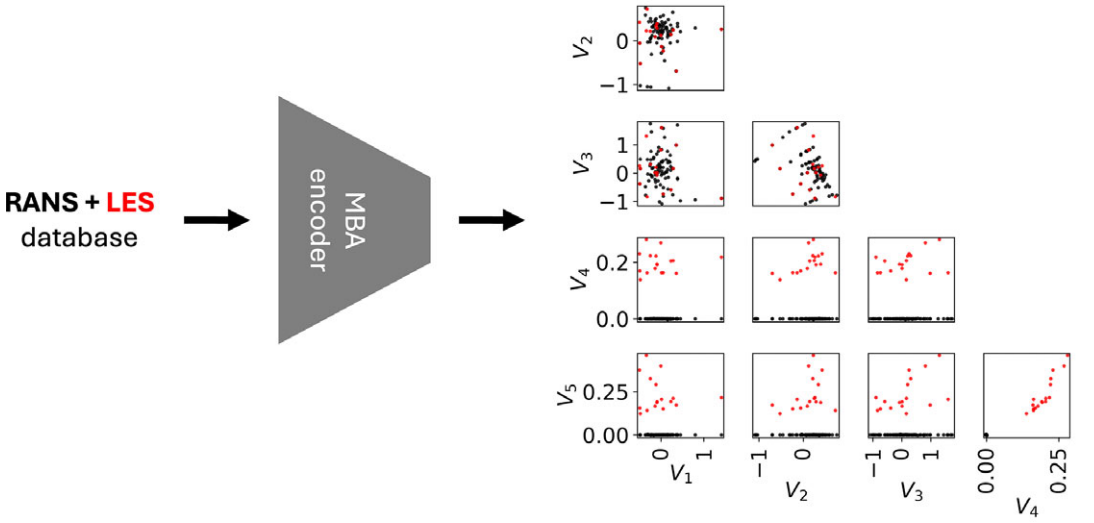


Figure 14. *Latent space projections of the multi-fidelity database embeddings. Black dots correspond to the RANS latent representations; red dots to the LES latent representations.*

black ones, indicating that this latent subspace is shared between the low- and high-fidelity datasets. To be precise, V_4-V_5 are not exactly zero for RANS samples, because the conditioning is enforced through a soft penalty rather than a hard constraint. In our experiments, their magnitude is nearly zero for RANS cases, on the order of $\mathcal{O}(10^{-5})$, while for LES cases it is typically on the order of $\mathcal{O}(10^{-1})$. This behavior can be interpreted as V_1-V_2-V_3 being responsible for predicting the nozzle jet flow phenomenon in low fidelity, while V_4-V_5 represent the correction term from low to high fidelity. In the present application, low- and high-fidelity data are two different representations of turbulence. RANS simulations are qualitatively correct and capture the physical phenomenon. Meanwhile, LES also capture the phenomenon quantitatively, as depicted in Figure 1 for the potential core region. For this reason, one can interpret the process as first learning a baseline (AE^1) using RANS data, and then correcting (AE^2) using LES data. It is expected that the structural, qualitative similarities between RANS and LES in this case enable a relatively small "latent space of corrections" (AE^2 vs. AE^1). If the differences between RANS and LES are more substantive, it is expected that the size of the latent correction would be considerably higher.

Additionally, Figure 15 shows the projections of the latent space colored by the database design parameters. Only three projections are shown for the sake of clarity. The design parameters vary smoothly along the latent space, indicating that the AE *learned* these parameters automatically (they were not provided during training).

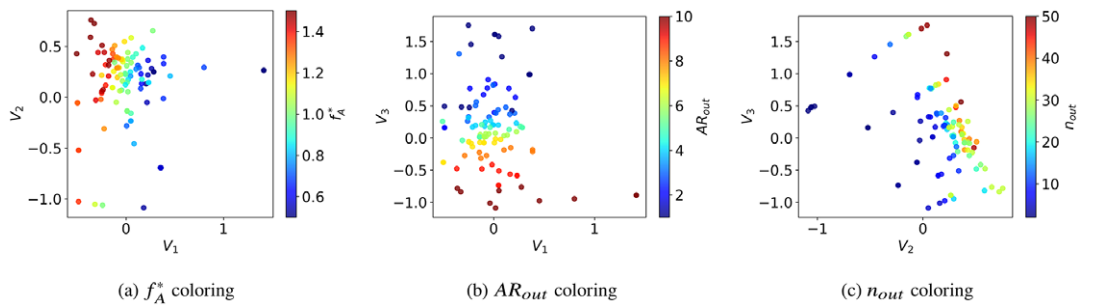


Figure 15. *Latent space projections colored by database design parameters. Blue: low parameter values; red: high parameter values.*

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

By navigating the latent space, it is possible to generate predictions as function of the database parameters. A closer look reveals that V_1 varies with f_A^*; V_2 with n_{out}, and V_3 with AR_{out}. Previous works in literature already found design parameters varying smoothly within the latent space of an AE for parametric datasets (Saetta et al., 2022; Fukami and Taira, 2023). Smoothness is not guaranteed in general but is promoted by the choice of continuous design variables and by inspecting that the resulting flow fields remain similar, so that the dataset maps a single regime of operations. It is very likely that in the presence of categorical design variables and extremely wide operating conditions, the resulting latent manifold would not be smooth.

5.2.2. Decoding

Figure 16 illustrates the decoding procedure of the generative model:

1. A new set of design parameters (not used in training) is selected (f_A^*, AR_{out} and n_{out}).
2. The latent vector corresponding to such parameters is computed. Different strategies can be employed, such as fully connected neural networks or interpolations; in the present article, we opted for Radial Basis Functions (RBF). Specifically, we are interpolating a scalar field (one for each database parameter) over sparse points (the latent variables in a three-dimensional space) using RBF with a "Thin-Plate Spline" kernel. No smoothing parameter for the RBF is applied, that is, the interpolant passes through the known data points.
3. Feeding the latent vector to the decoder with $V_4 = V_5 = 0$ yields a low-fidelity (RANS) prediction. On the contrary, setting $V_4 \neq 0$ and $V_5 \neq 0$ produces a high-fidelity (LES) prediction.

The decoder is, therefore, a multi-fidelity model capable of predicting both RANS and LES fields. From an engineering perspective, only high-fidelity predictions are of interest. However, low-fidelity data are also important during training to cover regions where high-fidelity data are scarce (see Appendix C). In addition, as will be shown in Section 5.3, they improve the prediction accuracy of the high-fidelity fields.

The capability of the AE to capture the differences between low- and high-fidelity representations of the phenomenon is highlighted in Figure 16. For simplicity, only the streamwise velocity component along the x central axis is shown (bottom right). The differences between AE and the corresponding CFD results (both low and high fidelity) are barely visible, with errors below 2%. The AE correctly captures the lack of accuracy of the low-fidelity representation in the potential core. The upper right plot shows the low-fidelity error (black for CFD, red for AE). The results are in agreement, indicating that the low-fidelity error reaches up to 10% in the potential core.

Global metrics (length of potential core L_{pc} and half-jet spreading at $x/D_{eq} = 2$ for the entire LES dataset) are reported in Figure 17. The scatter plots lie along the red line $x = y$ with negligible spreading. It

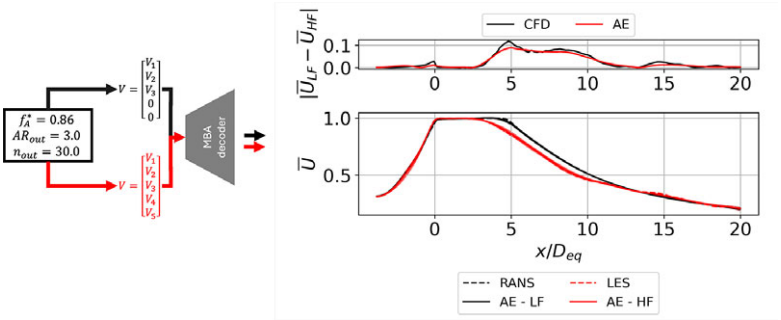
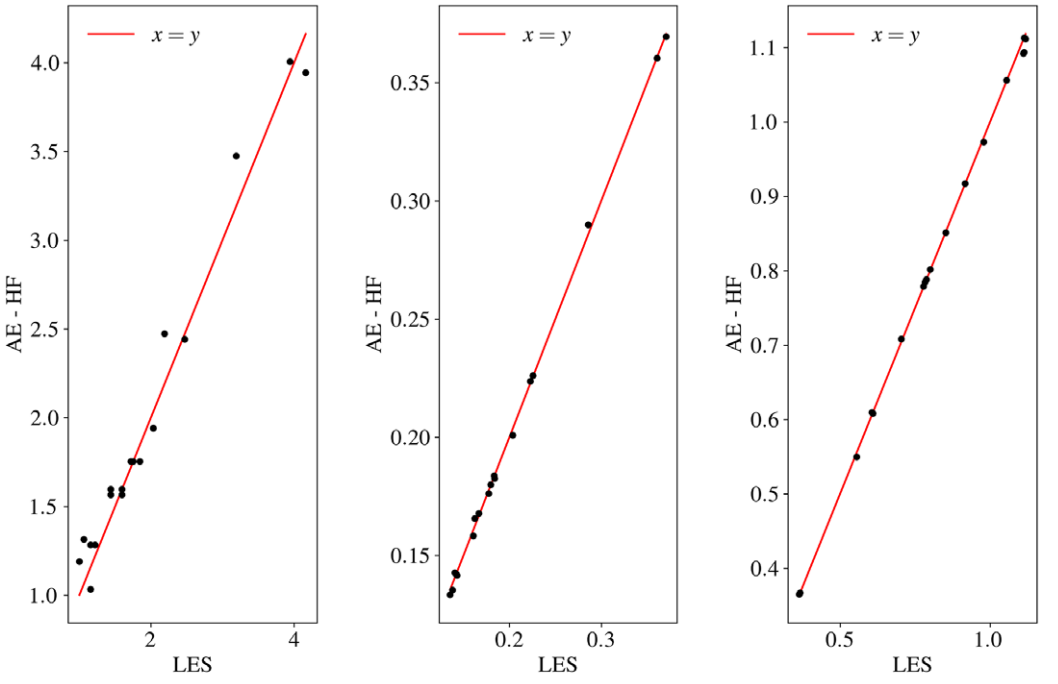


Figure 16. *Pipeline of multi-fidelity predictions using the trained ensemble of MBA decoders. Where $\overline{U}$ is the non-dimensional velocity defined as $\overline{U} = U/U_j$ and $\overline{U}_{LF}$ and $\overline{U}_{HF}$ are, respectively, referred to low and high fidelity.*

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press



(a) L_{pc} (b) $y_{1/2}$ (c) $z_{1/2}$

Figure 17. *Scatter plots of the predicted against the LES global metrics. (a) length of potential core L_{pc}; (b) jet spreading $y_{1/2}$; (c) jet spreading $z_{1/2}$ at $x/D_{eq} = 2$.*

indicates that the predicted fields preserve the accuracy of the global metrics which can be employed in an optimization objective function (see Appendix D).

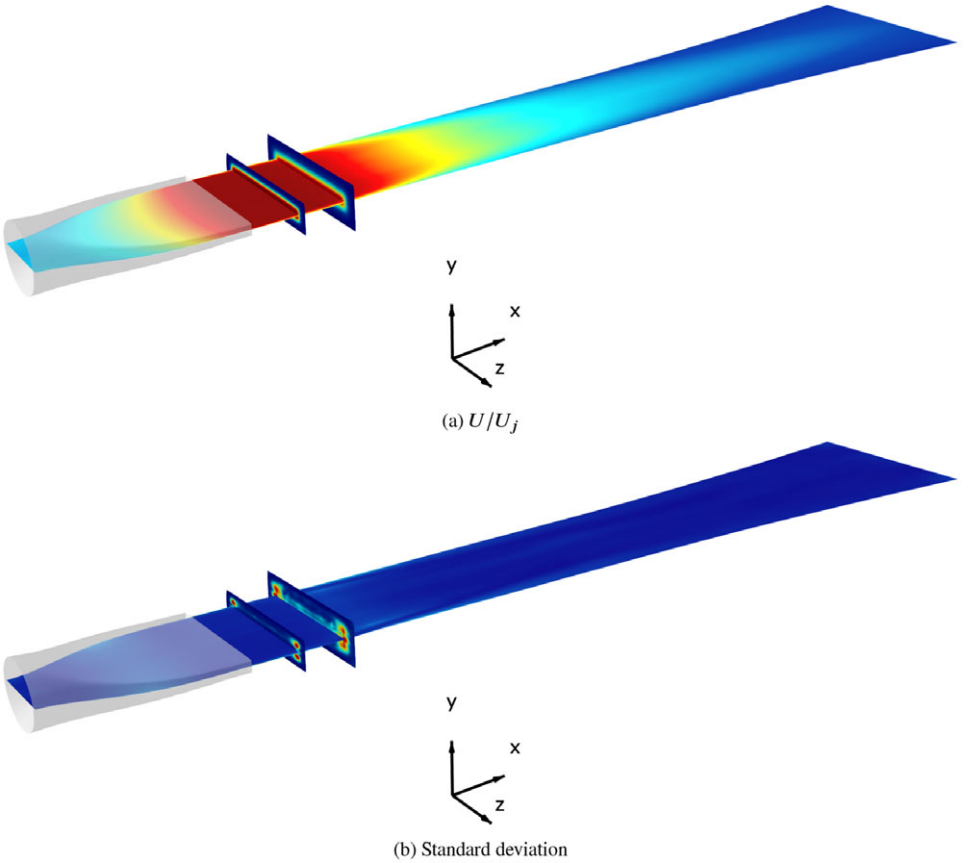
5.3. High-fidelity predictions comparison with CFD simulations

The assessment of AE accuracy is performed on a case that was not used during training or validation. The selected design parameters are: $f_A^* = 1.0$; $AR_{out} = 10.0$; $n_{out} = 30.0$. The AE predicts the velocity field (and its standard deviation) as shown in Figure 18(a) that shows the streamwise velocity predicted over the entire field. This represents the mean velocity obtained from the uncertainty quantification analysis. Figure 18(b) shows the corresponding standard deviation, highlighting that the regions with higher uncertainties of the AE are along the jet short-side boundaries.

For a clear and quantitative analysis of AE performance, the results in the *x*-*z* plane are considered for the comparison with CFD data. Despite the training database being constructed using LES time averaged over $\overline{t_{sim}} = 100$, a comparison of AE predictions against LES averaged over $\overline{t_{sim}} = 2000$ is also provided for this test case (not used in training). This comparison highlights an interesting feature of the proposed multi-fidelity method.

Figure 19 shows the mean (a) and standard deviation (b) of the prediction of the MBA ensemble. The mean appears smooth and symmetric, while the standard deviation highlights a small uncertain region around $x/D_{eq} = 2.0$, near the end of the potential core. Figure 20 shows the velocity field resulting from the LES averaged over $\overline{t_{sim}} = 100$ (Figure 20(a)), and $\overline{t_{sim}} = 2000$ (Figure 20(b)). The former exhibits a noisy and non-symmetric behavior, while the latter is more smooth. This difference is further illustrated in Figure 21, which shows the velocity profiles at different values of x/D_{eq}.

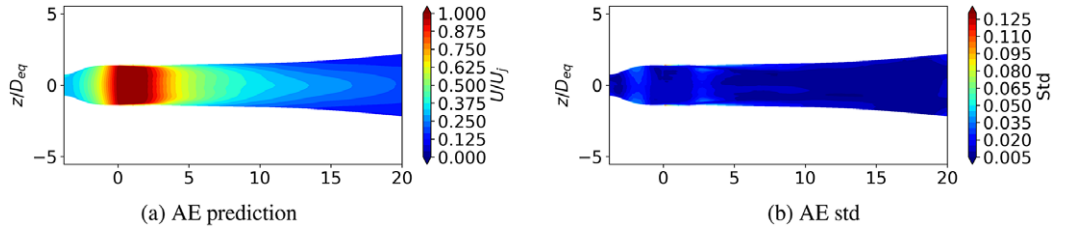
https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press



(a) U/U_j

(b) Standard deviation

Figure 18. *High-fidelity prediction of the MBA ensemble for* $f_A^* = 1.0$*;* $AR_{out} = 10.0$*;* $n_{out} = 30.0$*.*



(a) AE prediction (b) AE std

Figure 19. *High-fidelity prediction of the MBA ensemble: mean prediction (a) and its standard deviation (b).* $f_A^* = 1.0$*;* $AR_{out} = 10.0$*;* $n_{out} = 30.0$*. Streamwise velocity contour maps on the x-z plane.*

Figure 22 shows the absolute percentage error maps between the MBA ensemble predictions and the two LES (not statistically converged (a) and converged (b)). Remarkably, although the MBA ensemble was trained using non-converged data, its prediction in Figure 19(a) is closer to the converged solution of Figure 20(b), as highlighted by the error maps of Figure 22. This behavior arises from the presence of RANS data and from the conditioning term we employ. In particular, during the first training step, AE^1 is trained on RANS data, which are smooth and symmetric. As a result, the convolutional layers learn smooth feature maps and the symmetries of the flow. In addition, the RANS fields are qualitatively accurate enough to describe the underlying physics of the phenomenon. In the second step, AE^2 is trained on both RANS and LES data, starting from the previously learned weights. Through the conditioning

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

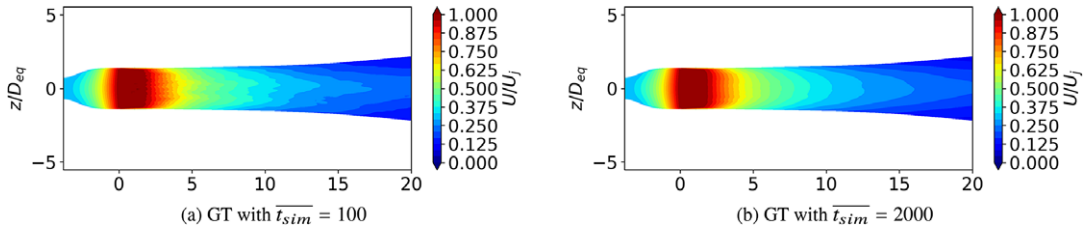


Figure 20. *Reference ground truth (GT): LES using* $\overline{t_{sim}} = 100$ *(a) and using* $\overline{t_{sim}} = 2000$ *(b) for the time averages.* $f_A^* = 1.0$*;* $AR_{out} = 10.0$*;* $n_{out} = 30.0$*. Velocity contour maps on the x-z plane.*

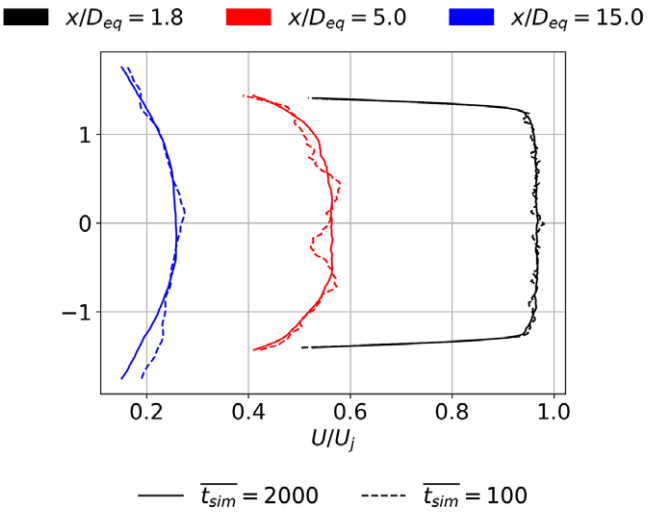


Figure 21. *Comparison of LES using* $\overline{t_{sim}} = 100$ *and* $\overline{t_{sim}} = 2000$ *for the time averages.* $f_A^* = 1.0$*;* $AR_{out} = 10.0$*;* $n_{out} = 30.0$*. Velocity profiles at different stations downstream the nozzle outlet.*

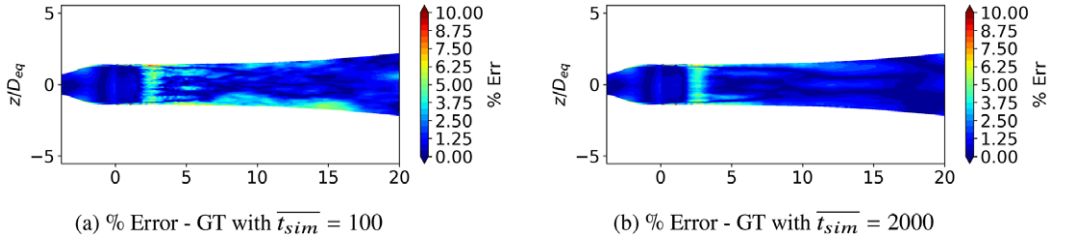


Figure 22. *MBA prediction errors with LES using* $\overline{t_{sim}} = 100$ *(a) and using* $\overline{t_{sim}} = 2000$ *(b) for the time averages.* $f_A^* = 1.0$*;* $AR_{out} = 10.0$*;* $n_{out} = 30.0$*. Velocity contour maps on the x-z plane.*

term, we guide the optimizer toward a solution that effectively corrects the RANS fields using LES data, also accounting for the symmetries still not completely captured by the not statistically converged LES simulations. Figure 23 presents a more detailed comparison between the MBA ensemble predictions and the LES averaged over $\overline{t_{sim}} = 2000$. The differences are within 2%. To complete the analysis, Figure 24 shows the velocity field on the *y*-*z* plane at $x/D_{eq} = 1$. Figures 24(a) and 24(b) show the LES results and the MBA predictions, respectively. Figures 24(c) and 24(d) present the local error (with respect to the

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

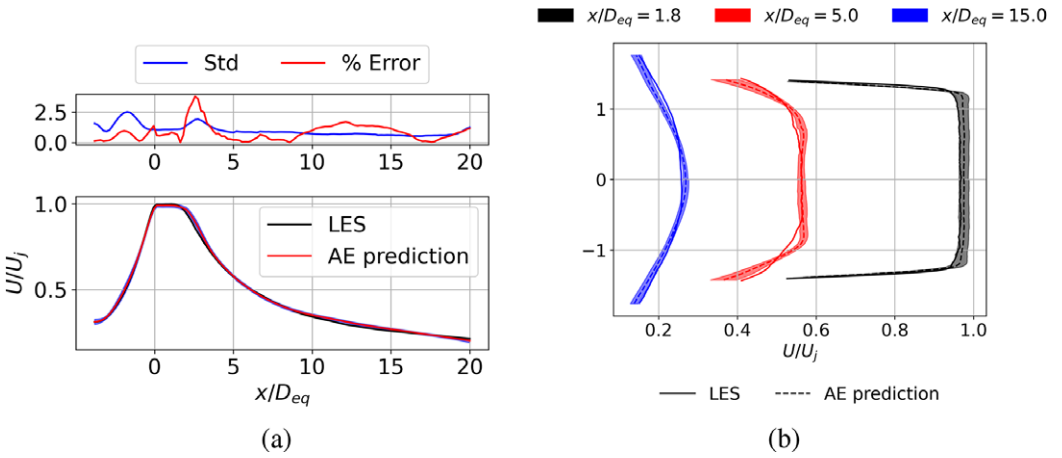


Figure 23. *High-fidelity prediction of the MBA ensemble and reference CFD simulation (LES using* $\overline{t_{sim}} = 2000$*) for* $f_A^* = 1.0$*;* $AR_{out} = 10.0$*;* $n_{out} = 30.0$*. (a) Velocity along the central nozzle axis; (b) velocity profiles at different stations downstream the nozzle outlet.*

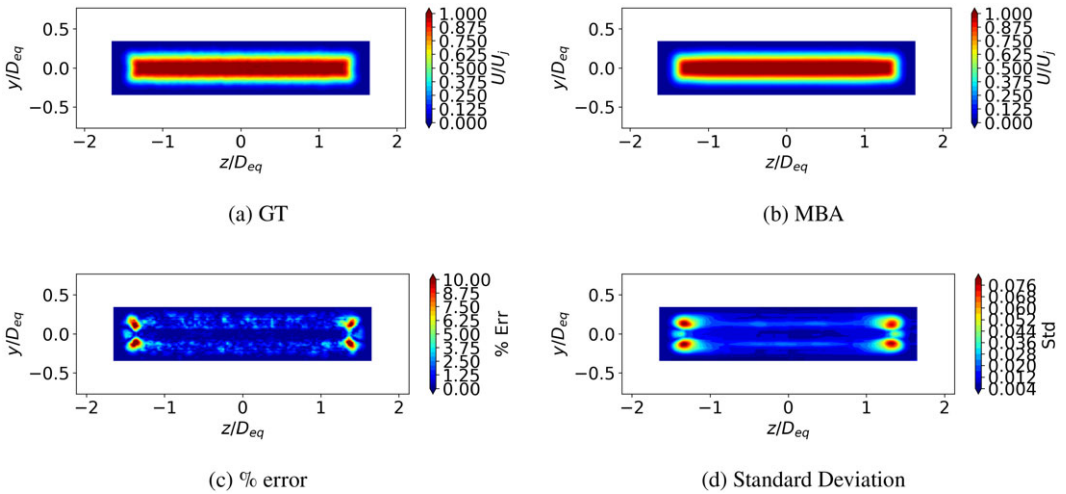


Figure 24. *High-fidelity prediction of the MBA ensemble and reference CFD simulation (LES using* $\overline{t_{sim}} = 2000$*) for* $f_A^* = 1.0$*;* $AR_{out} = 10.0$*;* $n_{out} = 30.0$*. Velocity contour map in the y-z plane at* $x/D_{eq} = 1$*.*

ground truth) and the standard deviation of the prediction, respectively. The largest errors are confined to the corners of the jet. The standard deviation clearly mirrors the error distribution, providing an effective way to identify regions of largest uncertainty when the ground truth is not available (see also Saetta et al. (2024)). Table 4 reports the global parameters computed from the LES and predicted fields, demonstrating the accuracy of the AE in predicting the control parameters. A larger error is present in the computation of L_{pc} (as confirmed by the error peak in Figure 23), but it remains negligible compared to the overall phenomenon (see Figure 17). The low-fidelity data guide the symmetry and the *smoothness* of the phenomenon, while the additional latent variables provide the high-fidelity correction. Indeed, the analysis using only a single-fidelity database with 20 LES for training, reported in Appendix C, shows that in the absence of low-fidelity data, the predictions are noisy and non-symmetric.

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

Table 4. *Control parameters computed in the predicted and LES fields for $f_A^* = 1.0$; $AR_{out} = 10.0$; $n_{out} = 30.0$*

	L_{pc}	$y_{1/2}$	$z_{1/2}$
LES	1.06	0.13	1.11
AE prediction	1.31	0.13	1.09

Table 5. *Computational cost and overall accuracy on the global parameters*

	Number of cases	Cost	MSE L_{pc}	MSE $y_{1/2}$	MSE $z_{1/2}$
LF	92	$\sim$ 11.2 kCPUh	8.16×10^{-1}	1.03×10^{-4}	5.73×10^{-3}
HF-$\overline{t_{sim}} = 100$	20	$\sim$ 906 kCPUh	$\mathbf{3.76 \times 10^{-2}}$	4.31×10^{-5}	1.01×10^{-3}
Multi-fidelity	112	$\sim$ 917 kCPUh	9.64×10^{-2}	$\mathbf{3.71 \times 10^{-6}}$	$\mathbf{5.20 \times 10^{-5}}$
HF-$\overline{t_{sim}} = 2000$	92	$\sim$ 30.6 MCPUh	–	–	–

In this case, the multi-fidelity method (compared to the standard single-fidelity approach) reduces the computational cost in two ways: by requiring a limited number of costly high-fidelity data and by allowing LES performed with a limited number of time steps (an order of magnitude lower). Table 5, which reports the computational cost for the entire setup (database construction and model training), highlights the savings achieved with this method. The accuracy of the predictive methods trained on single- and multi-fidelity shows that the best overall performance is achieved by the multi-fidelity approach. With only a small increase in computational cost compared to the HF-$\overline{t_{sim}} = 100$ case, the error in jet spreading is reduced by one order of magnitude for $y_{1/2}$ and by two orders of magnitude for $z_{1/2}$. The error in the potential core length L_{pc} is higher, but the difference is negligible (moreover, Table A3 shows that adding one latent variable on the multi-fidelity case improves the prediction of L_{pc}). The cost of constructing the HF-$\overline{t_{sim}} = 2000$ database is prohibitive and is therefore not considered in this analysis. Indeed, this is exactly the reason why multi-fidelity methods are necessary for these types of applications.

6. Using the latent space for shape optimization

As previously discussed, the trained multi-fidelity MBA ensemble can be used within an optimization loop to accelerate the design of nozzle shapes. By sampling the latent space, it is possible to generate a large number of flow predictions in real time and compute the quantities of interest (L_{pc}, $y_{1/2}$, and $z_{1/2}$), thereby mapping the latent space by assigning these quantities to each combination of latent variables. Employing this methodology enables the navigation of the latent space to find the combination of latent variables that minimizes jet extension (lowest values of L_{pc}) and maximizes jet spreading ($y_{1/2}$ and $z_{1/2}$), thereby enhancing jet mixing with the outer flow. The objective of this analysis is to illustrate the advantages of employing the multi-fidelity MBA ensemble as opposed to conducting a full nozzle optimization and should be seen as a proof-of-concept study. As illustrated in Figure 25, the latent space projections are mapped with the global parameters. This represents, in practice, the optimization landscape with the objective function $\min(L_{pc})$ and parameters V_i, $i = 1,\ldots,5$. It was obtained by performing 10000 flow predictions using the trained decoder, which would be unfeasible with CFD simulations. For clear visualization, an interpolation on a finer grid (5000×5000 points) was performed using RBF with a *thin-plate spline* kernel. For brevity, the same analysis for the other parameters of interest is not reported but exhibits similar behavior.

Considering the values of the generated latent vectors, the best combination for minimizing L_{pc} corresponds to a nozzle with $f_A^* = 0.500$, $AR_{out} = 9.501$, and $n_{out} = 15.554$. For this geometry,

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

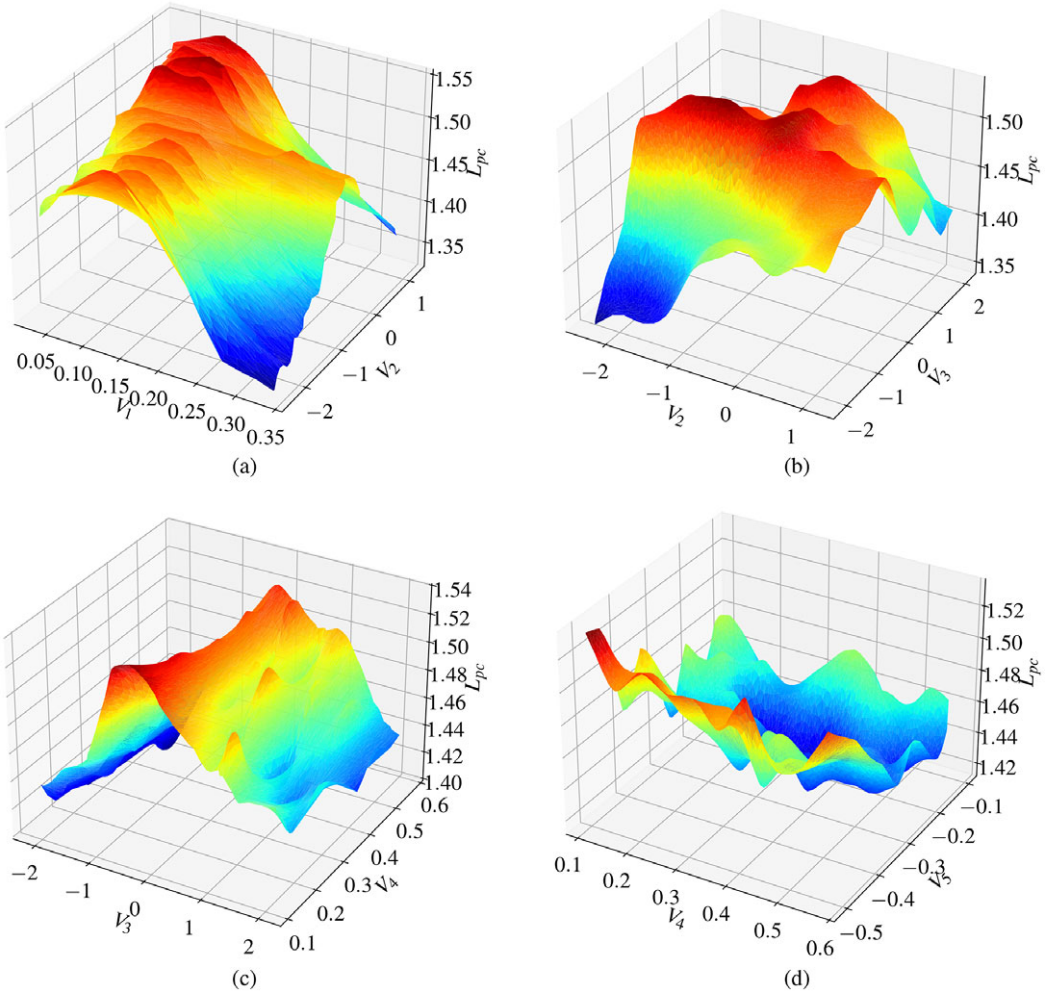


Figure 25. *Visualization of the global parameter L_{pc} along the latent space projections.*

$L_{pc} = 1.278$ m, $y_{1/2} = 0.123$ m, and $z_{1/2} = 1.079$ m. However, all the parameters of interest need to be optimized. To find the best compromise for minimizing overall jet extension, we performed a simple analysis: first, the parameters were scaled between 0 and 1; then, a weighted sum, $0.5 \cdot L_{pc} + 0.25 \cdot \frac{1}{y_{1/2}} + 0.25 \cdot \frac{1}{z_{1/2}}$, was computed, and the minimum was identified. The weighting is introduced to ensure that the potential core length and the jet spreading contribute equally to the optimization. Note that this objective function considers only the components of the jet along the y- and z-directions and does not account for the full jet cross-section or overall spreading. The corresponding nozzle geometry resulting from the conceptual optimization is $f_A^* = 0.977$, $AR_{out} = 3.093$, and $n_{out} = 2.018$, with global parameters $L_{pc} = 2.887$ m, $y_{1/2} = 0.214$ m, and $z_{1/2} = 0.675$ m. The velocity predictions for the minimum L_{pc} nozzle and the *best* nozzle geometry are shown in Figure 26. To better illustrate the selection of the best compromise between potential core length and jet spreading, Figure 27 shows the Pareto front associated with the two objective functions. The black dots represent the 10000 realizations obtained by sampling the latent space, whereas the red star denotes the selected case identified through the optimization procedure. This result does not represent an optimized nozzle, as it corresponds to the best among the 10000 generated samples. To identify the *best* geometry, a formal optimization procedure would be required.

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

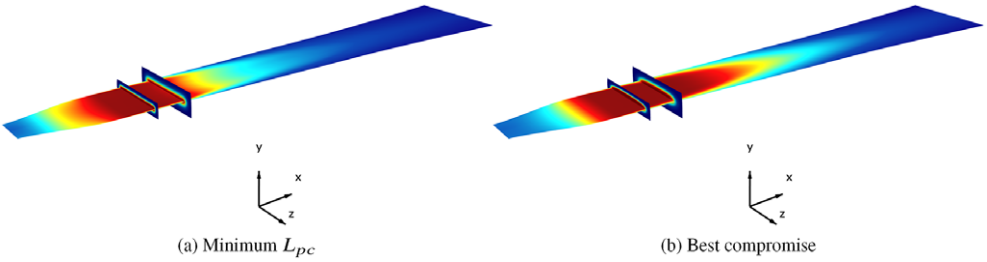


Figure 26. *Velocity contour map corresponding to (a) the nozzle providing the minimum L_{pc}: $f_A^* = 0.500$, $AR_{out} = 9.501$, and $n_{out} = 15.554$; (b) the nozzle geometry providing the best compromise between potential core length and jet spreading: $f_A^* = 0.977$, $AR_{out} = 3.093$, and $n_{out} = 2.018$.*

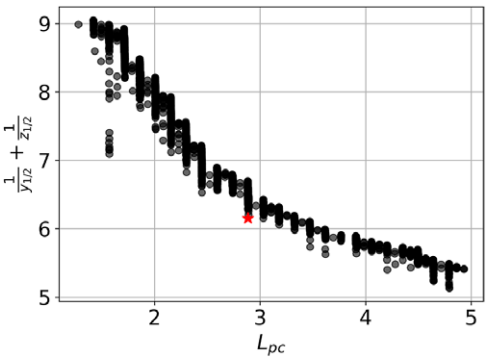


Figure 27. *Pareto front of the optimization procedure sampling 10000 latent vectors. Objective function minimization: potential core length L_{pc} and jet spreading $\frac{1}{y_{1/2}} + \frac{1}{z_{1/2}}$. Red star: best selected case.*

7. Conclusions

The present article proposes a set of multi-fidelity, multi-branch AEs for predicting high-fidelity flows with quantified uncertainties while limiting the computational effort required to construct the training database. Specifically, three ML methods (a multi-branch AE architecture, a multi-fidelity training algorithm for AEs, and an ensemble technique for quantifying the prediction uncertainties) have been combined to build a unified training pipeline.

The proposed method was applied to a problem of practical engineering interest: nozzle jet flow prediction. Specifically, the multi-fidelity model was constructed using a database composed of many RANS simulations (low fidelity) and only a few LES (high fidelity). The result is a multi-fidelity model capable of predicting both low- and high-fidelity data across the entire design space. High-fidelity predictions with quantified uncertainty were obtained in real time, with an average error of approximately 2%. The small number of high-fidelity samples enabled the construction of a correction term from low to high fidelity.

In contrast to standard single-fidelity training procedures, the present approach drastically reduces the computational cost associated with database construction (by more than 30 times in the present test). The computational savings from using the multi-fidelity algorithm are twofold: only a small fraction of LES are needed, and the LES time averages can be obtained by reducing the number of time steps required for the statistics computation by an order of magnitude.

Finally, we showed that by predicting a large number of flow fields (10000) using the decoder and computing the parameters of interest for each field, it is possible to visualize them as a function of the latent variables. In this way, we can identify the optimized nozzle geometry through latent space analysis

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

with minimal computational effort. Performing the same analysis using CFD simulations would be computationally unfeasible.

Data availability statement. The data that support the findings are openly available in the "Multi-Fidelity Nozzle Jet Flows" repository on Hugging Face: https://huggingface.co/datasets/taarg/MultiFidelity_Nozzle_JetFlows.

Acknowledgements. We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy).

Author contribution. Conceptualization: E.S., M.M., R.T., G.I.; Data curation: E.S., M.M., R.T., G.I.; Methodology: E.S., M.M., R.T., G.I.

Funding statement. This work received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interests. The author(s) declare none.

Ethical standards. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

Agostini L (2020) Exploration and prediction of fluid dynamical systems using auto-encoder technology. *Physics of Fluids 32*(6), 067103. https://doi.org/10.1063/5.0012906.

An S and Jeon JJ (2024) Customization of latent space in semi-supervised variational autoencoder. *Pattern Recognition Letters 177*, 54–60. https://doi.org/10.1016/j.patrec.2023.11.018.

Baig Y, **Ma HR**, **Xu H and You L** (2023) Autoencoder neural networks enable low dimensional structure analyses of microbial growth dynamics. *Nature Communications 14*, 7937. https://doi.org/10.1038/s41467-023-43455-0.

Bradshaw P (1987) Turbulent secondary flows. *Annual Review of Fluid Mechanics 19*, 53–74.

Bridges J and Wernet M (2011) Technical Report. *The NASA Subsonic Jet Particle Image Velocimetry (PIV) Dataset*. Cleveland, OH: NASA Glenn Research Center.

Brindise MC and Vlachos PP (2017) Proper orthogonal decomposition truncation method for data denoising and order reduction. *Experiments in Fluids 58*(4), 28. https://doi.org/10.1007/s00348-017-2320-3.

Davis DO and Gessner FB (1992) Experimental investigation of turbulent flow through a circular-to-rectangular transition duct. *AIAA Journal 30*(2), 367–375. https://doi.org/10.2514/3.10927.

Economon T, **Palacios F**, **Copeland S**, **Lukaczyk T and Alonso J** (2016) SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal 54*. https://doi.org/10.2514/1.J053813.

Freedman D, **Pisani R and Purves R** (2007) *Statistics (international student edition)*, 4th Edn. New York: WW Norton & Company.

Fukami K and Taira K (2023) Grasping extreme aerodynamics on a low-dimensional manifold. *Nature Communications 14*, 6480. https://doi.org/10.1038/s41467-023-42213-6.

Goc KA, **Lehmkuhl O**, **Park GI**, **Bose ST and Moin P** (2021) Large eddy simulation of aircraft at affordable cost: a milestone in computational fluid dynamics. *Flow 1*, E14. https://doi.org/10.1017/flo.2021.17.

Hinton GE and Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science 313*(5786), 504–507. https://doi.org/10.1126/science.1127647.

Le Gratiet L (2013) Multi-fidelity Gaussian Process Regression for Computer Experiments. PhD dissertation, Université Paris-Diderot - Paris VII.

Li R, **Zhang Y and Chen H** (2022) Physically interpretable feature learning of supercritical airfoils based on variational autoencoders. *AIAA Journal 60*(11), 6168–6182. https://doi.org/10.2514/1.J061673.

Lin S, **Zhang C and Yang Y** (2024) Second-order gradient loss guided single-image super-resolution. In Cruz C, Zhang Y and Gao W (eds), *Intelligent Computers, Algorithms, and Applications*. vol *2036*. Communications in Computer and Information Science. Singapore: Springer, pp. 17–28.

Loshchilov I and Hutter F (2019) Decoupled weight decay regularization. In *International Conference on Learning Representations*. Available at https://arxiv.org/abs/1711.05101.

Mathieu M, **Couprie C and LeCun Y** (2016) Deep Multi-Scale Video Prediction Beyond Mean Square Error. *arXiv preprint arXiv:1511.05440*.

McKay MD, **Beckman RJ and Conover WJ** (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics 21*(2), 239–245.

Menter FR (1994) Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal 32*(8), 1598–1605. https://doi.org/10.2514/3.12149.

Mi J and Nathan G (2009) Statistical properties of turbulent free jets issuing from nine differently-shaped nozzles. *Flow, Turbulence and Combustion 84*, 583–606.

Mi J, **Deo RC and Nathan GJ** (2005) Characterization of turbulent jets from high-aspect-ratio rectangular nozzles. *Physics of Fluids 17*(6), 068102. https://doi.org/10.1063/1.1928667.

Milano M and Koumoutsakos P (2002) Neural network modeling for near wall turbulent flow. *Journal of Computational Physics 182*(1), 1–26. https://doi.org/10.1006/jcph.2002.7146.

Ngoc KM and Hwang M (2020) Finding the best k for the dimension of the latent space in autoencoders. In *Computational Collective Intelligence*. vol *12496*. Lecture Notes in Computer Science. Cham: Springer, pp. 453–464.

Partin L, **Geraci G**, **Rushdi AA**, **Eldred MS and Schiavazzi DE** (2023) Multifidelity data fusion in convolutional encoder/decoder networks. *Journal of Computational Physics 472*, 111666. https://doi.org/10.1016/j.jcp.2022.111666.

Paszke A, **Gross S**, **Massa F**, **Lerer A**, **Bradbury J**, **Chanan G**, **Killeen T**, **Lin Z**, **Gimelshein N**, **Antiga L**, **Desmaison A**, **Kopf A**, **Yang E**, **DeVito Z**, **Raison M**, **Tejani A**, **Chilamkurthy S**, **Steiner B**, **Fang L**, **Bai J and Chintala S** (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems* 32. https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf (accessed 15 July 2026).

Racca A, **Doan NAK and Magri L** (2023) Predicting turbulent dynamics with the convolutional autoencoder echo state network. *Journal of Fluid Mechanics 975*, A2. https://doi.org/10.1017/jfm.2023.716.

Saetta E, **Tognaccini R and Iaccarino G** (2022) Machine learning to predict aerodynamic stall. *International Journal of Computational Fluid Dynamics 36*(7), 641–654. https://doi.org/10.1080/10618562.2023.2171021.

Saetta E, **Tognaccini R and Iaccarino G** (2024) Uncertainty quantification in autoencoders predictions: Applications in aerodynamics. *Journal of Computational Physics 506*, 112951. https://doi.org/10.1016/j.jcp.2024.112951.

Saetta E, **Massa M**, **Tognaccini R and Iaccarino G** (2025) *Multi-Fidelity Nozzle Jet Flows.* Hugging Face. Available at https://huggingface.co/datasets/taarg/MultiFidelity_Nozzle_JetFlows (accessed 15 July 2026).

Shen Y, **Patel HC**, **Xu Z and Alonso JJ** (2024) Application of Multi-Fidelity Transfer Learning with Autoencoders for Efficient Construction of Surrogate Models. In *AIAA SCITECH 2024 Forum.* https://doi.org/10.2514/6.2024-0013. Available at https://arc.aiaa.org/doi/abs/10.2514/6.2024-0013 (accessed 15 July 2026).

Solera-Rico A, **Sanmiguel Vila C**, **Gómez-López M**, **Wang Y**, **Almashjary A**, **Dawson STM and Vinuesa R** (2024) ?-Variational autoencoders and transformers for reduced-order. *Nature Communications 15*, 1361. https://doi.org/10.1038/s41467-024-45578-4.

Tang Y, **Fang WZ**, **Zheng CY and Tao WQ** (2025) Applications of POD-based reduced order model to the rapid prediction of velocity and temperature in data centers. *Applied Thermal Engineering 263*, 125310. https://doi.org/10.1016/j.applthermaleng.2024.125310.

Tangsali K, **Krishnamurthy VR and Hasnain Z** (2020) Generalizability of convolutional encoder–decoder networks for aerodynamic flow-field prediction across geometric and physical-fluidic variations. *Journal of Mechanical Design 143*(5), 051704. https://doi.org/10.1115/1.4048221.

Tsuchiya Y, **Horikoshi C and Sato T** (1986) On the spread of rectangular jets. *Experiments in Fluids 4*, 197–204.

Turisini M, **Amati G and Cestari M** (2023) *LEONARDO: A Pan-European Pre-Exascale Supercomputer for HPC and AI Applications.* arXiv: 2307.16885 [cs.DC]. Available at https://arxiv.org/abs/2307.16885 (accessed 15 July 2026).

Vreman AW (2004) An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids 16*(10), 3670–3681. https://doi.org/10.1063/1.1785131.

Zaman K (1996) Axis switching and spreading of an asymmetric jet: the role of coherent structure dynamics. *Journal of Fluid Mechanics 316*, 1–27.

Zaman K (2012) Flow-field surveys for rectangular nozzles. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. https://doi.org/10.2514/6.2012-69.

A. Sensitivity analysis of the latent space dimension

The size of the latent variable is a critical hyper-parameter in this problem. We choose to treat it as a control parameter, performing training with an increasing size and stopping when a further increase in latent dimensions does not lead to better loss reduction. The resulting minimum dimension effectively represents the dimensionality of the latent manifold.

The present database was generated using three precise independent design parameters for the geometry of the nozzle. Therefore, the expectation is that the *minimum* dimensionality of the latent space is 3 (it could be lower if the design parameters are not independent). In more complex situations and if the resulting flow fields span different regimes, it is possible that the latent manifold requires a larger number of dimensions. In general, keeping the latent space small can facilitate interpretation.

We performed a sensitivity analysis on the latent space dimension for the first training step; the results are reported in Table A1. The analysis included the R^2 regression score, the explained variance, and the Structural Similarity Index Measure (SSIM). To define a criterion for selecting the minimum number of required latent variables, we rely on other works in literature: for POD, this corresponds to retaining the smallest number of modes that capture at least 99% of the explained variance, which is a standard criterion in the literature (Brindise and Vlachos, 2017; Tang et al., 2025). Similarly, in our case, sufficiency was defined as the smallest latent dimension for which the reconstruction metric considered here (i.e., R^2) reached 0.99 and showed no substantial further improvement with additional latent variables. While this convergence criterion with increasing number of modes is mathematically guaranteed for POD in terms of reconstruction error reduction, it is not generally guaranteed for deep learning

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

Table A1. *AE^1 training*

l.v.	Avg. training time	Avg. inference time	R^2	Exp. Var.	SSIM
1	6.317 min	0.0368 sec	0.984	0.984	0.993
2	6.264 min	0.0358 sec	0.994	0.994	0.996
3	6.257 min	0.0350 sec	0.995	0.995	0.997
4	6.260 min	0.0346 sec	0.996	0.996	0.997
5	6.282 min	0.0336 sec	0.996	0.996	0.997

Note: Metrics for RANS data predictions (test dataset) using an MBA ensemble with 50 elements. Hardware: NVIDIA RTX 6000 Ada. Times referred to wall clock time. "Avg. Training Time" is referred to a single MBA in the ensemble, while "Avg. Inference Time" is referred to the inference time for one prediction using the whole trained ensemble model.

algorithms. However, similar analyses are commonly adopted in autoencoder-based reduced-order models and should be interpreted solely as a convergence study with respect to the latent space dimension (Ngoc and Hwang, 2020; Baig et al., 2023).

As shown in Table A1, with only two latent variables, we already reach $R^2 = 0.994$. This is because the n_{out} design parameter has a minimal influence on the solution field. However, neglecting this parameter prevents us from distinguishing between rectangular and elliptical outlet sections. For this reason, we introduced a third latent variable to also capture the small effects of n_{out} (see Figure 15). Thus, the third latent variable can be seen as a *service variable* which is not introduced to improve the aggregate error, but to preserve the full parametric description of the problem. Adding more than three latent variables would not significantly improve the model performance.

For the analysis of the latent space dimension of the second training step, the number of latent variables for AE^1 (i.e., using only LF data) is fixed at 3.

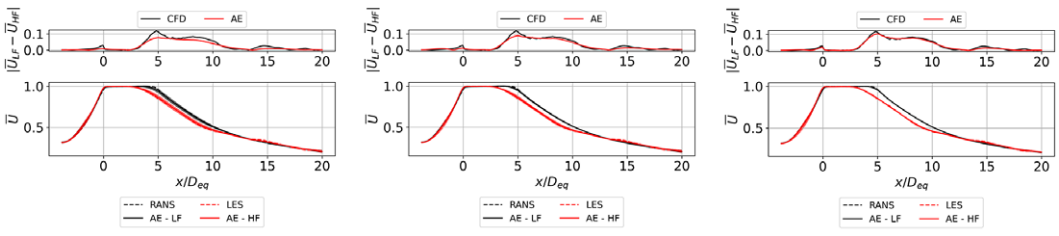
We performed three different trainings of the multi-fidelity MBA ensemble, considering 1, 2, and 3 additional latent variables to map the correction term from LF to HF. A first analysis of the computational cost and prediction accuracy metrics is presented in Table A2. The results indicate that one additional latent variable is sufficient to map the correction between RANS and LES: R^2 and explained variance are both >99%.

To better assess the predictive performance, an analysis of the physical variables and quantities of interest ($L_{pc}, y_{1/2}$, and $z_{1/2}$) is required. The prediction of axial velocity along the central nozzle axis is shown in Figure A1. The predictions obtained by adding 1

Table A2. *AE^2 training*

l.v.	Avg. training time	Avg. inference time	R^2	Exp. Var.	SSIM
+1	19.0240 min	0.0352 sec	0.993	0.993	0.962
+2	16.805 min	0.0345 sec	0.995	0.995	0.967
+3	18.710 min	0.0348 sec	0.997	0.997	0.971

Note: Metrics for multi-fidelity RANS–LES model, LES (test dataset) predictions using an MBA ensemble with 50 elements and 3 latent variables for AE^1. Hardware: NVIDIA RTX 6000 Ada. "Avg. Training Time" includes both AE^1 and AE^2 trainings. Times referred to wall clock time. "Avg. Training Time" is referred to a single multi-fidelity MBA in the ensemble, while "Avg. Inference Time" is referred to the inference time for one prediction using the whole trained ensemble model.



(a) +1 latent variable (b) +2 latent variables (c) +3 latent variables

Figure A1. *Streamwise velocity on the nozzle central axis. Multi-fidelity predictions using 1, 2, and 3 additional latent dimensions for the second (HF) training step. $f_A^* = 0.86$, $AR_{out} = 3.0$, $n_{oout} = 30.0$.*

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

Table A3. *Control parameters computed in the predicted and LES fields using 1, 2, and 3 additional latent dimensions*

	L_{pc}	$y_{1/2}$	$z_{1/2}$
LES	1.06	0.13	1.11
AE - +1 l.v.	1.42	**0.13**	1.07
AE - +2 l.v.	1.31	**0.13**	**1.09**
AE - +3 l.v.	**1.13**	**0.13**	**1.09**

Note: $f_A^* = 1.0$; $AR_{out} = 10.0$; $n_{out} = 30.0$. The best results are reported in bold.

latent variable are already accurate. However, the upper plots of Figure A1, which show the difference between HF and LF data ($|\overline{U}_{LF} - \overline{U}_{HF}|$, which is interpretable as the correction term), indicate a convergent behavior toward the LES data as the latent space dimension increases. The result is confirmed by the trend of the global parameters monitored in Table A3. The results obtained by adding two latent variables are already satisfactory, and this is the one considered in the article to balance the accuracy and the compression rate.

Regarding the training and inference costs when varying the number of latent variables, these are reported in Tables A1 and A2. It is clear that they are not significantly affected by the size of the latent space and remain negligible compared to the CFD cost (Table 5).

B. Sensitivity analysis on the MBA ensemble dimension

To quantify the model-form uncertainties in the predictions, we adopted an ensemble technique involving an ensemble of 50 MBAs. Here, we illustrate a sensitivity analysis of the ensemble dimension, as well as an analysis of the individual MBA realizations within the ensemble.

Figure B1 shows the standard deviation (i.e., uncertainty) of the predicted velocity along the central nozzle axis for an LES prediction with $f_A^* = 1.0$, $AR_{out} = 10.0$, and $n_{out} = 30.0$, for different ensemble dimensions. Using only 10 items is insufficient to compute meaningful statistics on the model uncertainty as this leads to an underestimation of variability. By contrast, using 50, 100, and 200 items exhibits a convergent behavior. An ensemble of 50 MBAs is already sufficient to capture the uncertainty induced by model variability.

Figure B2 shows the predictions of each ensemble member for the same case (LES with $f_A^* = 1.0$, $AR_{out} = 10.0$, and $n_{out} = 30.0$). The mean and median predictions are also reported and overlap, proving that the final predictions we reported are robust to the presence of outliers. Figure B2 also shows that the ensemble members are representative of the variability in the model. The region of the jet core decay exhibits the highest variability, due to the notorious tendency of RANS predictions to overestimate turbulent mixing in the jet shear layers, thus increasing the length of the jet core (see Figure 1).

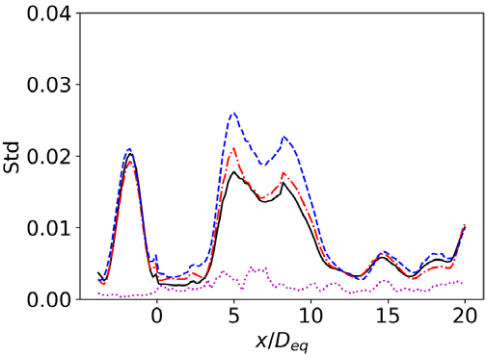


Figure B1. *Standard deviation of the prediction of velocity along the central nozzle axis LES for* $f_A^* = 1.0$*;* $AR_{out} = 10.0$*;* $n_{out} = 30.0$ *and different dimensions of the MBA ensembles. Dotted magenta: 10; dashed blue: 50; dot-dash red: 100; continuous black: 200.*

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

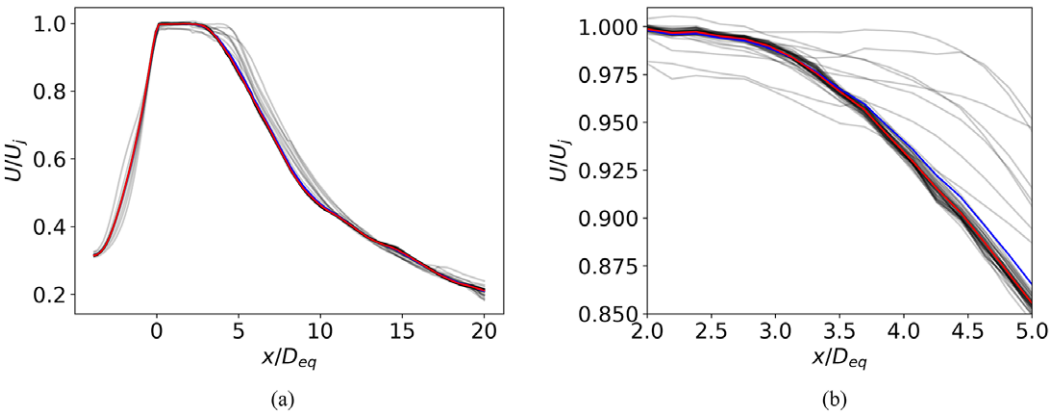


Figure B2. *Velocity along the central nozzle axis. Mean prediction (blue), median prediction (red), and item prediction (gray) for LES of $f_A^* = 1.0$; $AR_{out} = 10.0$; $n_{out} = 30.0$. (a) whole prediction; (b) zoom around the ending of the potential core jet.*

C. Multi-fidelity vs. LES single-fidelity

In this Appendix, results are reported using a standard single-fidelity approach with the available LES dataset, highlighting the difficulties of training an AE with a very limited high-fidelity database. The AE model employed here is an ensemble of MBAs, as described in Section 4. Figure C1 shows the latent space projections colored by database design parameters. For AR_{out}, the latent variables are smooth and well organized, whereas the other two parameters were not effectively *captured* by the training. This can be explained by Table C1, which reports the correlation between the design parameters and the L^2 norm of the discretized velocity field (a measure of energy content) using the Pearson correlation index r (Freedman et al., 2007). Values of $|r| \to 1$ indicate high correlation, while $|r| \to 0$ indicates low correlation. It shows that the most important parameter is AR_{out}, followed by a very low influence of f_A^* and n_{out}. One way to capture parameters with lower influence on the flow fields is to include more data representing this variability. However, this would require additional LES samples, further increasing the computational cost.

The velocity contour map and its standard deviation on the x-z plane are reported in Figure C2. The single-fidelity prediction is noisy and non-symmetric. Moreover, Figure C2(d) shows large errors inside the nozzle, caused by the AE's inability to identify the f_A^* parameter. The UQ analysis (standard deviation in Figure C2(b)) highlights the difficulty of predicting this region, with high uncertainty inside the nozzle. The prediction of the global parameters reported in Table C2 shows good accuracy, even though only the AR_{out} design parameter is captured indicative of the fact that the leading design parameter is sufficient for the assessment of the global parameters.

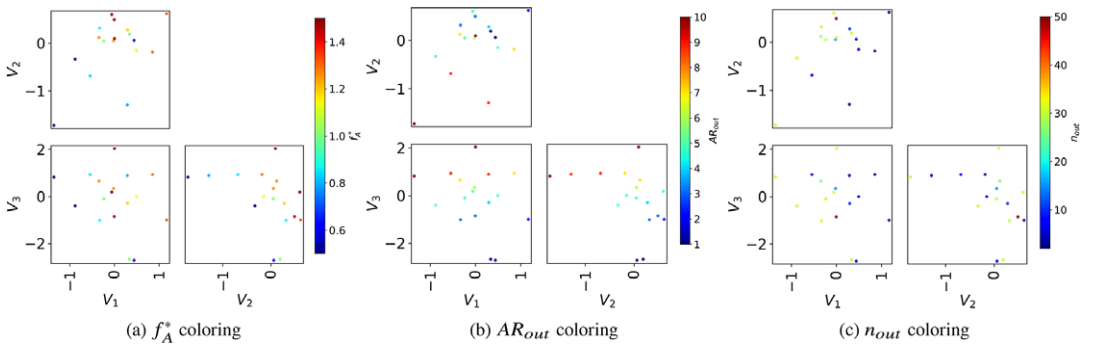


Figure C1. *Latent space projections colored by database design parameters. Blue: low parameter values; red: high parameter values. Single-fidelity method using the LES database.*

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press

Table C1. *Pearson correlation index* r *describing the correlation between the design parameters and the L^2 norm of the velocity field (a measure of the energy content)*

	f_A^*	AR_{out}	n_{out}
$\|r\|$	0.173	0.980	0.020

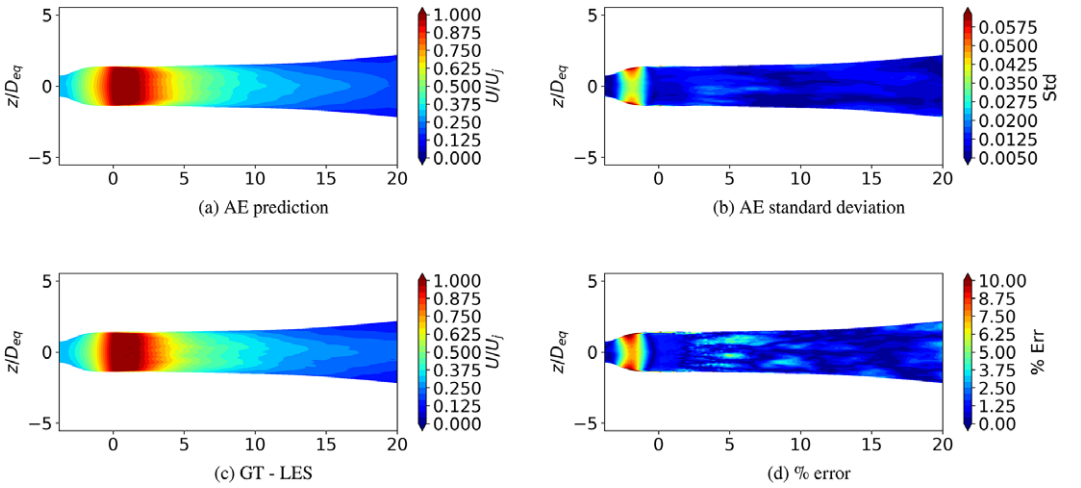


Figure C2. *Prediction of the MBA ensemble (single-fidelity model) and reference ground truth (GT) for $f_A^* = 1.0$; $AR_{out} = 10.0$; $n_{out} = 30.0$. Velocity contour maps on the* x-z *plane.*

Table C2. *Control parameters computed in the predicted and LES fields for $f_A^* = 1.0$; $AR_{out} = 10.0$; $n_{out} = 30.0$*

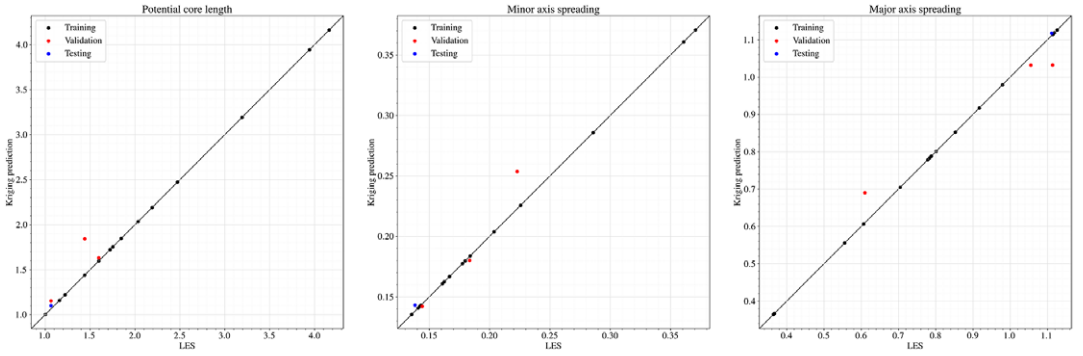
	L_{pc}	$y_{1/2}$	$z_{1/2}$
LES	1.06	0.13	1.11
AE prediction	1.13	0.13	1.05

D. Comparison with classical optimization methods

A typical procedure in optimization problems is to interpolate the objective functions across the design space and then use this interpolant to carry out the optimization process. In this section, the AE predictions are compared to Multi-Fidelity Kriging (MFK) interpolation (Le Gratiet, 2013) applied to the objective functions of the multi-fidelity dataset. Figure D1 shows the interpolated values plotted against the LES results.

The accuracy, in terms of mean squared error on the high-fidelity dataset, is shown in Table D1. The multi-fidelity AE shows higher accuracy in predicting jet spreading. Although the MFK interpolation achieves higher accuracy in predicting the potential core length, the advantages of predicting the entire flow field, rather than only the objective functions, should not be underestimated. Additionally, the errors reported in Table D1 refer to the entire training dataset, where the Kriging interpolant is constructed to pass through the training data. Moreover, Figure D1 shows that MFK fails to accurately predict the objective functions for unseen geometries, whereas the autoencoder maintains consistent accuracy across the entire dataset, as illustrated in Figure 17.

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press



(a) L_{pc} (b) $y_{1/2}$ (c) $z_{1/2}$

Figure D1. *Scatter plots of the global metrics predicted by MFK against the LES results; (a) potential core length, (b) jet spreading $y_{1/2}$ at $x/D_{eq} = 2$, (c) jet spreading $z_{1/2}$ at $x/D_{eq} = 2$.*

Table D1. *Control parameters computed in the predicted and LES fields for $f_A^* = 1.0$; $AR_{out} = 10.0$; $n_{out} = 30.0$*

	MSE L_{pc}	MSE $y_{1/2}$	MSE $z_{1/2}$
Multi-fidelity AE	9.64×10^{-2}	$\mathbf{3.71 \times 10^{-6}}$	$\mathbf{5.20 \times 10^{-5}}$
Multi-fidelity Kriging	$\mathbf{8.64 \times 10^{-3}}$	4.93×10^{-5}	6.84×10^{-4}

Cite this article: Saetta E, Massa M, Tognaccini R and Iaccarino G (2026). Multi-fidelity autoencoders: RANS-LES jet flow predictions. *Data-Centric Engineering*, 7, e30. doi:10.1017/dce.2026.10069

https://doi.org/10.1017/dce.2026.10069 Published online by Cambridge University Press