

Predictive modelling of the dominant axisymmetric pressure mode of a high-speed turbulent jet flow using spectral proper orthogonal decomposition and machine learning

Cite as: Phys. Fluids **38**, 095126 (2026); doi: [10.1063/5.0341107](https://doi.org/10.1063/5.0341107)

Submitted: 29 April 2026 · Accepted: 13 July 2026 ·

Published Online: 9 September 2026



View Online



Export Citation



CrossMark

Vasily Gryazev,^{1,a)} Umberto Armani,¹ Bin Zang,¹ Amal Roy Murali,² Lidiya Kalathiparambil Kennedy,³ Zhuofei Wang,¹ Annabel P. Markesteijn,¹ Vassili Toropov,¹ Marc C. Jacob,² Elnaz Naghibi,⁴ Vladimir Ryabov,⁵ and Sergey A. Karabasov¹

AFFILIATIONS

¹Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom

²Ecole Centrale de Lyon, CNRS, Université Claude Bernard Lyon 1, INSA Lyon, LMFA, UMR5509, 69130 Ecully, France

³Institute of Structural Engineering, Eidgenössische Technische Hochschule Zürich, Stefano-Franscini-Platz 5, Zürich 8093, Switzerland

⁴University of East London, University Way, London E16 2RD, United Kingdom

⁵Future University Hakodate, 116 Kamedanakano-cho, Hakodate 041-8655, Hokkaido, Japan

^{a)}Author to whom correspondence should be addressed: vasily.gryazev@qmul.ac.uk

ABSTRACT

Analysis of dominant features associated with coherent structures in turbulent flows is essential for the development of Reduced-Order Models (ROMs) in many fluid dynamics problems. As the first step in this direction, the axisymmetric pressure mode is extracted from the Large Eddy Simulation (LES) of a high-speed round jet flow, analyzed, and reconstructed using flow decomposition and Machine Learning (ML) techniques. The motivation of these data-driven analyses is to ultimately develop robust methods for modeling turbulent flows using only a fraction of the LES time series for training purposes, thus minimizing computing costs. The considered LES solution is based on the high-resolution Compact Accurately Boundary Adjusting high-Resolution Technique accelerated on Graphics Processing Unit cards for conditions of the National Aeronautics and Space Administration Small Hot Jet Acoustic Rig experiment with a round jet at acoustic Mach number 0.9. A frequency-domain Spectral Proper Orthogonal Decomposition (SPOD) of the LES data is performed to extract the time-varying coefficients. The SPOD coefficients condense important unsteady features of the turbulent jet flow, while preserving its spatiotemporal coherence. The coefficients are reconstructed using four machine-learning approaches. The predictive ROMs developed here based on ML are specifically focused on reconstructing the leading SPOD time coefficient of the axisymmetric ($m = 0$) pressure mode at $St = 0.1$ that corresponds to the dominant hydrodynamic mode that is known to strongly contribute to the peak-noise radiation of high-speed round jets at small angles to the jet axis. The first ROM approach employs Feedforward Neural Networks, a technique previously recommended for modeling turbulent flows over a plunging airfoil [Lui and Wolf, “Construction of reduced-order models for fluid flows using deep feedforward neural networks,” *J. Fluid Mech.* **872**, 963–994 (2019)]. The second ROM uses Genetic Programming, a symbolic regression method widely recognized in the optimization literature, though it has not yet been applied to unsteady fluid dynamics problems. The third approach adopts Echo State Networks, a time-series prediction and reconstruction technique originating from the field of Reservoir Computing. Finally, the fourth approach leverages Transformer, one of the most advanced architectures within the family of Artificial Neural Networks. Although transformers have demonstrated efficiency in long-term reconstruction of turbulent wake flows, their application to high-speed jet modeling remains unexplored. The stability and accuracy of the considered ML methods for turbulent signal reconstruction are evaluated in terms of capturing its essential properties, such as the amplitude distribution and convection phase.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0341107>

I. INTRODUCTION

In many areas of fluid mechanics ranging from aerospace and wind engineering to geophysics, fast Reduced-Order Models (ROMs) of complex turbulent flow are required to enable large-scale parametric studies and support efficient design optimization and flow-control strategies.^{2–4} A common step in ROM development is the application of dimensionality reduction techniques to identify and extract the most dynamically significant flow structures. To this end, several modal decomposition methods have been developed, including Proper Orthogonal Decomposition (POD)^{5–7} and Dynamic Mode Decomposition (DMD) along with its variants.^{8,9} Because many turbulent flows exhibit both coherent and stochastic features in space and time, frequency-domain Spectral Proper Orthogonal Decomposition (SPOD) method¹⁰ has become an attractive tool for analyzing turbulent flows. SPOD enables a clear separation of stochastic realizations from the spatiotemporally coherent flow structures at a given frequency.¹¹

With the most dynamically important flow modes extracted, the second step is to capture the evolution of the temporal coefficients, which can then be used to reconstruct the flow dynamics and hence, predicting the development of the flow. Several methods, including the Galerkin projection and its variants,¹² as well as Sparse Identification of the Non-linear Dynamics (SINDy),¹³ have been developed to approximate the key dynamics of these temporal evolution. The challenge here is that ROMs based on such reconstructions may exhibit unstable behavior in attempts to reproduce the complex dynamics.¹⁴ Despite progress in the development of stabilized ROMs,^{15,16} the application of such techniques for turbulent flows at high Reynolds numbers remains a challenge.

Machine Learning (ML) methods based on Artificial Neural Networks (ANN), on the other hand, have gained more research attention recently. These ML methods have proven to perform well for reconstructing the dynamics of high-dimensional systems with good degree of efficiency.^{17,18} For example, there have been a number of studies that demonstrate the effectiveness of neural-network-based ML in predicting turbulent flow fields, e.g., using Convolutional Neural Network (CNN),^{19,20} Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM),²¹ and physics-informed neural networks (PINN).^{22,23} However, there is yet no consensus in terms of the choice and the expected performance for the different ML methods for complex turbulent flows, not to mention at relatively high Reynolds numbers. Therefore, the present work is inspired by the need to further explore and systematically examine the performance of the various ML methods in capturing the dynamics of the temporal coefficients, which can subsequently be used to reconstruct the turbulent flows via the reduced-order modeling approach.

An unforced (natural) high-speed jet flow carries a broad range of space-time scales, which does not have any distinct tonal frequency in contrast to the airfoil and cylinder flows previously considered,¹ and hence is regarded as a suitable turbulent flow scenario for the study. The configuration is an isolated, unheated, static jet from the NASA Small Hot Jet Acoustic Rig (SHJAR) experiment.²⁴ The acoustic Mach number, defined as $Ma = U_j/a_\infty$, is 0.9, where U_j is the jet exit velocity and a_∞ is the ambient speed of sound. The Reynolds number based on the nozzle diameter D_j is $Re \approx 10^6$, corresponding to the SHJAR Set Point (SP) seven jet case. The near-field flow solution is obtained using Wall-Modeled Large Eddy Simulation (LES). The solver is based on the Compact Accurately Boundary-Adjusting

high-Resolution Technique (CABARET) accelerated by Graphics Processing Units (GPUs), which was developed and validated for jet LES in a series of works.^{25–34}

Three neural-network-based ML methods are selected for learning the dynamics of the temporal coefficients and reconstructing the flow structures, namely, the Feedforward Neural Networks (FNNs), the Echo State Network (ESN), and the Transformer. Feedforward Neural Networks (FNN) was found to be very effective in modeling a range of dynamical systems.^{1,35} In Ref. 1, the FNN technique was applied to the reduced-order modeling of turbulent flow over a cylinder and a plunging airfoil, where the flow modal decomposition was performed using POD. When a sufficiently long time series of the flow field was used, the method demonstrated reliable flow field predictions beyond the training window. ESN, an approach belonging to the broader class of Reservoir Computing methods, exploits a Recurrent Neural Network (RNN) comprising interconnected neurons.³⁶ They were shown to overcome the problem of vanishing and exploding gradients predictions often encountered by RNNs.³⁷ In recent years, ESNs have become increasingly popular in data-driven fluid dynamics modeling.^{38–41} Another promising method is the Transformer Neural Network,⁴² which leverages on the self-attention mechanism and often provides an improved performance over RNNs and CNNs in inferring contextual information, i.e., the dynamics of the flow evolution from one to the next time instance. In recent studies,^{43,44} transformer models have been used to learn and predict wake flows behind a square cylinder and a surface-mounted tower and demonstrated the ability to accurately capture the dynamics of the wake beyond training datasets.

In addition, Genetic Programming (GP) is also employed in the current work as an alternative method. GP originates from the class of Genetic Algorithms (GAs) with the goal of addressing symbolic regression problems. Although it is well known in the optimization research,^{45,46} it also has potential for capturing the low-order dynamics of complex turbulent flows. To the best of authors' knowledge, the present study is among the first to apply GP-based algorithms, to reconstruct the temporal dynamics of a high-Reynolds-number jet flow, and to systematically compare their performance to those from neural-network-based ML methods used in fluid mechanics modeling. As a result, even though not exhaustive, the methods selected in the present work can represent a range of the popular and relatively well-established approaches used in the literature.

Following the previous work on low-rank dynamics of unforced turbulent jets,⁴⁷ the objective of the reduced-order modeling here is to reconstruct the axisymmetric pressure mode at low frequencies. This particular mode is known to be dominant for noise radiation from Kelvin–Helmholtz instability waves at small angles to the jet axis, which largely contribute to the peak of jet noise spectra.⁴⁸

From the Computational Aeroacoustics viewpoint, importance of the axisymmetric pressure mode for jet noise can be illustrated by using the pressure solution and its radial derivative as boundary conditions in the linear flow region outside of the jet flow to solve the Kirchhoff problem to propagate the pressure signal to the far-field. From the viewpoint of modeling of effective acoustic sources inside the jet, it can be noted that the fluctuating pressure is coupled with the turbulent Reynolds stresses in the momentum component of the Navier–Stokes equations. At the same time, the Reynolds stresses play the role of effective source terms in many linear jet noise models, one

of which is the Goldstein Generalized Acoustic Analogy (GGAA).⁴⁹ The effect of different azimuthal modes of fluctuating Reynolds stresses extracted from the CABARET LES solution of a high-speed round jet flow at similar Mach and Reynolds number conditions compared to the current work on the far-field noise spectra at different observer angles was previously investigated in Ref. 50. In particular, it was shown that the contribution of the axisymmetric mode of Reynolds stresses is dominant for a range of acoustic frequencies, $St \in [0.1, 0.3]$ corresponding to the peak noise spectra for observer angles 30° , 60° , and 90° to the jet axis. The picture at higher frequencies is qualitatively different. For example, at $St = 2$ and an observer angle of 90° , the first eight azimuthal modes were needed to reproduce the Rolls-Royce SILOET far-field spectra to within 2 dB [see Fig. 11(c) in Ref. 50]. Therefore, the focus of the current work on data-driven modeling of the low-frequency dynamics of the axisymmetric pressure mode can be viewed as a first step in the direction of developing a more comprehensive data-driven turbulent jet mixing noise model applicable over the entire range of noise frequencies. This motivates the present study, which assesses how well four state-of-the-art ML methods can reconstruct the dominant features of the low-frequency jet mode of interest. Of particular interest is to challenge robustness of the ROMs for reconstructing the turbulent pressure signals significantly outside of the ML training window. The most robust ML methods will, therefore, be recommended as suitable candidates for extending the current ROM framework to reconstructing the turbulent pressure and Reynolds stress fluctuations for higher-order azimuthal modes and higher frequencies—toward the ultimate goal of developing suitable data-driven ROMs to supplement short jet LES time series for use in acoustic optimization studies.

The paper is organized as follows: Sec. II summarizes details of the Large Eddy Simulation for the NASA SHJAR jet at SP7 conditions and outlines the main features of the LES database used for the data-driven reduced-order modeling. Section III presents the main steps of the frequency-domain SPOD method,^{10,47} verifies its implementation for the LES data, and compares the modal coefficients extracted via the block and time reconstruction. The Hilbert transform used for the analysis of the computed time signals is also discussed in this section. In Sec. IV, background details of the ML algorithms that are trained using the reconstructed modal time series are outlined. The results of the application of these ML methods during training and reconstruction are presented in Sec. V. This is followed by a discussion on the robustness and accuracy of each ML method in reconstructing the key amplitude and phase characteristics of the turbulent time signals, which is followed by the conclusions in Sec. VI.

II. JET FLOW MODELLING

In the NASA Small Hot Jet Acoustic Rig (SHJAR) experiment performed at SP7 conditions, a single-stream unheated static jet issues from a profiled axisymmetric convergent nozzle with an exit diameter, $D_j = 0.05\text{m}$ at an acoustic Mach number $M_a = 0.9$. Following Ref. 51, wall modeled LES solutions were obtained using the GPU-CABARET method. An unstructured LES grid with snappy-hexahedral/hanging node mesh elements was generated with local refinements around the nozzle and across the shear layers, so that the grid resolution around the nozzle jet lip-line corresponds to $(dz/D_j, dr/D_j, r d\theta/D_j) = (0.16 \times 10^{-1}, 0.78 \times 10^{-2}, 0.78 \times 10^{-2})$ with a total grid count of 61.2×10^6 cells. The LES grid remains fine in the potential core region of the jet and then expands by a factor of

2 in terms of cell size at $z/D_j = 10, 15, 20$ downstream of the nozzle exit, where zero corresponds to the nozzle exit location. The computational domain spans $30D_j$ radially, $100D_j$ axially downstream of the nozzle exit, and $10D_j$ axially upstream of the nozzle exit.

Standard stagnation pressure and temperature boundary conditions were specified at the nozzle inlet. Despite leading to an initially laminar inflow condition at the nozzle exit, this is acceptable because of the sufficiently refined mesh in the early shear layers of the jet, allowing a fast transition of the LES solution to turbulence downstream of the nozzle. Characteristic nonreflecting boundary conditions were specified on all open-domain boundaries, and an algebraic wall shear stress model was used for all nozzle surfaces to account for a coarsely resolved boundary layer with approximately five CABARET discretization points per boundary layer thickness. Figures 1 and 2 show the resulting axial and radial distributions of the mean stream-wise velocity and its root mean square fluctuation obtained from the LES solution, which are in very good agreement with the NASA PIV measurements.²⁴

For use in reduced-order modeling, the LES pressure solution component was interpolated from the original unstructured grid onto a coarser structured cylindrical grid, $(n_z \times n_r \times n_\theta)$ spanning the axial, radial, and circumferential directions, $([z/D_j] \times [r/D_j] \times [\theta]) \in ([0, 10] \times [0, 3] \times [0, 2\pi])$, respectively. Table I summarizes the main parameters of jet flow and LES data sampling. In total, the LES dataset available for post-processing has 5348 time frames corresponding to more than 740 convective time units, D_j/U_j , where U_j is the jet exit velocity.

Using the LES solution interpolated onto the cylindrical grid, zero-mean pressure fluctuations p' are computed as

$$p'(z, r, \theta, t) = p(z, r, \theta, t) - \bar{p}(z, r, \theta), \quad (1)$$

where $\bar{(\cdot)}$ denotes the time averaged part and $(\cdot)'$ denotes the fluctuating part of the pressure field. An instantaneous snapshot of the resulting fluctuating pressure field, p' , in the nozzle symmetry plane ($r = z$) is shown in Fig. 3. These space-time pressure fields are used in the subsequent modal decomposition analysis.

The fluctuating pressure signals are subjected to a Fourier transform in the circumferential direction according to

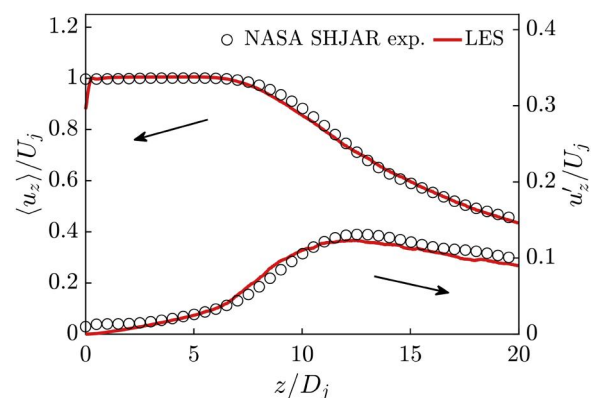


FIG. 1. Axial profiles of the mean axial velocity and fluctuating axial velocity compared to experimental results.

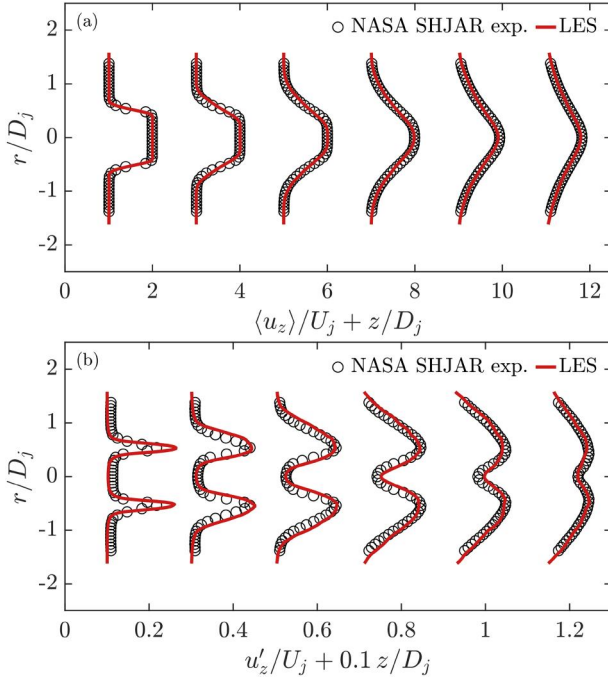


FIG. 2. Radial profiles of the mean axial velocity (a) and fluctuating axial velocity (b) compared to experimental results.

TABLE I. The operating conditions of NASA SHJAR SP7 jet and parameters of the space-time domain used for data analyses.

Parameter	Value
Set point (SP)	7
Acoustic Mach number, M_a	0.9
Jet Mach number, M_j	0.98
Nozzle temperature ratio (NTR)	0.835
Reynolds number based on jet diameter	10^6
Cylindrical grid spacing, $dz/D_j = dr/D_j, d\theta$	0.1, 1.4
Sampling frequency, $\times (U_j/D_j)$	8.33
Number of time samples	5348

$$p'(z, r, \theta, t) = \sum_m p'_m(z, r, t) e^{im\theta}. \quad (2)$$

The computed axisymmetric mode of the Fourier decomposition ($m = 0$) is of particular interest because of its importance for peak jet noise at small angles to the jet axis. A typical snapshot of the pressure fluctuation for this mode is shown in Fig. 4.

The resulting two-dimensional in space and one-dimensional in time pressure fluctuations are then decomposed into SPOD modes following Ref. 10 as discussed in Sec. III. The extracted temporal coefficient from the modal signals will then be used to train and test the accuracy and robustness of four ML methods used: FNN, GP, ESN, and Transformer.

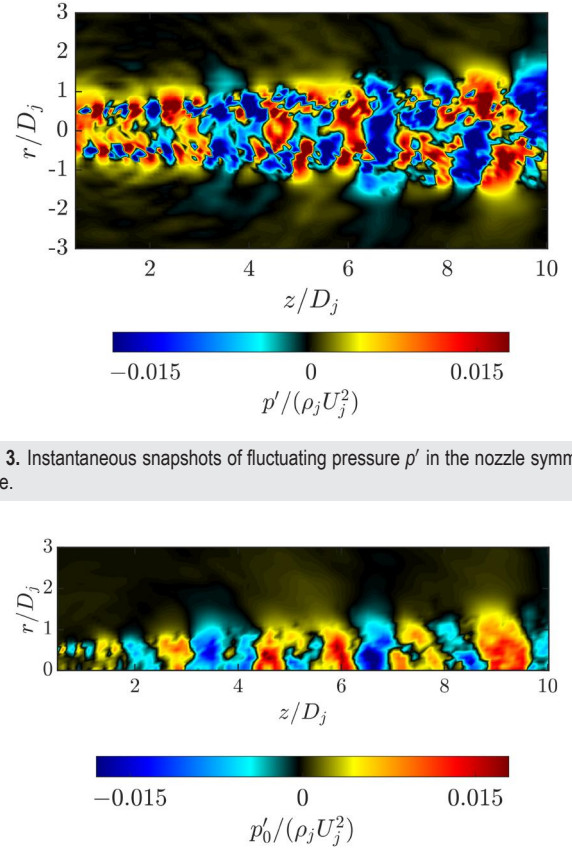


FIG. 3. Instantaneous snapshots of fluctuating pressure p' in the nozzle symmetry plane.

FIG. 4. Instantaneous snapshot of the fluctuating pressure p'_0 for the Fourier azimuthal mode $m = 0$.

III. SPECTRAL PROPER ORTHOGONAL DECOMPOSITION

Spectral Proper Orthogonal Decomposition (SPOD) is a data-driven modal decomposition technique designed for statistically stationary flows.¹⁰ It extends classical Proper Orthogonal Decomposition by incorporating temporal Fourier analysis to extract modes that are optimally ranked in energy and coherent in both space and frequency domain. Following Refs. 10 and 47, let us assume a fluctuating flow field $\mathbf{q}_i = \mathbf{q}(t_i)$ ($i = 1, \dots, n_t$) from which a snapshot matrix can be reconstructed as $\mathbf{Q} = [\mathbf{q}_1 \mathbf{q}_2 \dots \mathbf{q}_{n_t}]$. In this paper, the fluctuating flow field refers to the first azimuthal mode of fluctuating pressure $\mathbf{q}_i = p'_0(\mathbf{x}, t_i)$ and $\mathbf{x}$ represents the spatial coordinates (z, r) .

The SPOD method computes the cross-spectral density (CSD) matrix at each frequency by segmenting the time series into n_{blk} overlapping blocks, applying a temporal window function, and performing a Discrete Fourier Transform (DFT). At each frequency f , the cross-spectral density matrix is formed as

$$\mathbf{S}(\mathbf{x}, f) = \hat{\mathbf{Q}}^*(\mathbf{x}, f) \mathbf{W}(\mathbf{x}) \hat{\mathbf{Q}}(\mathbf{x}, f), \quad (3)$$

where $\hat{\mathbf{Q}}(\mathbf{x}, f)$ contains the Fourier-transformed data blocks

$$\hat{\mathbf{Q}} = [\hat{\mathbf{q}}_1 \hat{\mathbf{q}}_2 \dots \hat{\mathbf{q}}_{n_{\text{blk}}}], \quad (4)$$

and $\mathbf{W}(\mathbf{x})$ is a spatial inner-product weighting matrix, and $*$ denotes the conjugate transpose. The eigen-decomposition of $\mathbf{S}(\mathbf{x}, f)$ yields a set of orthonormal modes

$$\Psi(\mathbf{x}, f) = [\psi_1(\mathbf{x}, f), \psi_2(\mathbf{x}, f), \dots, \psi_{n_{\text{blk}}}(\mathbf{x}, f)], \quad (5)$$

and their corresponding modal energies

$$\Lambda = [\lambda_1(f), \lambda_2(f), \dots, \lambda_{n_{\text{blk}}}(f)] \quad (6)$$

according to

$$\frac{1}{n_{\text{blk}}} \mathbf{S}(\mathbf{x}, f) \Psi(\mathbf{x}, f) = \Psi(\mathbf{x}, f) \Lambda(\mathbf{x}, f). \quad (7)$$

The SPOD modes

$$\Phi(\mathbf{x}, f) = [\phi_1(\mathbf{x}, f), \phi_2(\mathbf{x}, f), \dots, \phi_{n_{\text{blk}}}(\mathbf{x}, f)] \quad (8)$$

can then be expanded at frequency f as

$$\Phi(\mathbf{x}, f) = \frac{1}{\sqrt{n_{\text{blk}}}} \hat{\mathbf{Q}}(\mathbf{x}, f) \Psi(\mathbf{x}, f) \Lambda(\mathbf{x}, f)^{-1/2}. \quad (9)$$

Following the formulation proposed by Towne *et al.*,¹⁰ two types of SPOD expansion coefficients can be extracted to reconstruct the original flow field via inverse SPOD operations: (i) block-wise coefficients and (ii) time-resolved coefficients. Both represent the projection of the flow field onto the SPOD modes and encode how much energy each mode contributes over time. The primary distinction lies in the time resolution of the extracted coefficients and the type of reconstruction they enable.

A. Block-wise SPOD coefficients

The *block-wise coefficients* $a_j(f)$ are computed by projecting the discrete Fourier transform of the j th time block, $\hat{\mathbf{q}}_j(\mathbf{x}, f)$, onto the corresponding SPOD mode $\phi_j(\mathbf{x}, f)$, using a spatially weighted inner product,

$$a_j(f) = \langle \hat{\mathbf{q}}_j(\mathbf{x}, f), \phi_j(\mathbf{x}, f) \rangle_W, \quad (10)$$

where $\langle \cdot, \cdot \rangle_W$ denotes the inner product weighted by the spatial weight matrix W .

These coefficients represent the modal amplitudes within each block and are used to reconstruct frequency-domain realizations on a per-block basis. The inverse Fourier transform is then applied to each block, and the overlapping segments are combined using a window-based overlap-add procedure. This block-wise reconstruction yields a piecewise approximation of the original signal, with temporal resolution tied to the block spacing.

B. Time-resolved SPOD coefficients

For applications requiring finer temporal resolution or continuous-time analysis, time-resolved coefficients $a_j(f, t)$ can be computed. These coefficients are obtained by projecting windowed, Fourier-transformed snapshots of the signal onto the SPOD modes in a sliding fashion⁴⁷

$$a_j(f, t) = \int_{\Delta T} \int_{\Omega} \phi_j^*(\mathbf{x}, f) \mathbf{W}(\mathbf{x}) \mathbf{q}(\mathbf{x}, \tau + t) \omega(\tau) e^{-i2\pi f \tau} d\mathbf{x} d\tau, \quad (11)$$

where $\omega(\tau)$ is a temporal window function and $\phi_j(\mathbf{x}, f)$ is the SPOD mode at frequency f .

Compared to the block-wise formulation, time-resolved coefficients offer finer time sampling and enable a continuously updated reconstruction of the flow field. This is particularly useful when studying transient behavior, performing phase analysis, or training reduced-order models. However, the reconstruction also requires applying inverse Fourier transforms at each time step, followed by proper overlap handling and window correction. Both coefficient types ultimately allow accurate reconstruction of the original signal.

C. Block vs time reconstruction

Given the azimuthally averaged pressure field $p'(\mathbf{x}, t)$, the reconstruction precision is compared for the SPOD coefficients resolved in blocks and time. Block-wise coefficients supply one value per block, whereas time-resolved coefficients provide a value at each time step via a sliding-window projection. Figure 5(a) shows the original field with reconstructions at two representative time instants. To quantify differences, the normalized root mean square error (RMSE) at time t is calculated.

$$\text{RMSE}_{\text{norm}}(t) = \frac{1}{\sigma_{\text{ref}}} \left\{ \frac{1}{n_z n_r} \sum_{i=1}^{n_z} \sum_{j=1}^{n_r} \left[p_0^{\text{ref}}(z_i, r_j, t) - p_0^{\text{rec}}(z_i, r_j, t) \right]^2 \right\}^{1/2}, \quad (12)$$

where $\sigma_{\text{ref}} = \sqrt{\frac{1}{n_z n_r} \sum_{i=1}^{n_z} \sum_{j=1}^{n_r} [p_0^{\text{ref}}(z_i, r_j, t)]^2}$ is the spatial standard deviation of the azimuthally averaged pressure field. Figure 5(b) shows the difference $\Delta = \text{RMSE}_{\text{block}} - \text{RMSE}_{\text{time}}$.

D. Low-rank reconstruction using the dominant SPOD mode at $\text{St}=0.1$ and $m=0$

A low-rank reconstruction is obtained by retaining only the leading SPOD mode of the axisymmetric component ($m=0$) at $\text{St}=0.1$. Reconstructions are generated using both time-resolved and block-wise coefficients, and snapshots at selected time instants are compared in Fig. 6. Both approaches recover the dominant large-scale structures. The ML inputs are the real and imaginary parts of the leading time coefficient $a_1(\text{St}=0.1, t)$. The real part of the input signal is shown in Fig. 7, where two training windows are highlighted: T2000 (first 2000 samples) and T3000 (first 3000 samples). The remaining samples (out of $N=5348$) are used for testing.

The two training window lengths, $T=2000$ and $T=3000$ samples, were selected to probe the sensitivity of the data-driven reconstructions to the amount of training data while keeping the partition of the LES record physically meaningful. At the sampling rate, $T=2000$ corresponds to approximately 240 convective time units (TUs, with one TU defined as D_j/U_j) or about 24 periods of the dominant $\text{St}=0.1$ mode. The larger sample size, $T=3000$, corresponds to approximately 360 TUs.

Based on the previous experience with statistical averaging of turbulent flow and noise signals obtained from jet LES, 240 TUs approximately correspond to the minimum time-averaging length required to obtain reasonably converged solutions at the peak jet noise frequencies, $\text{St} \in [0.1, 0.2]$. Accordingly, $T=2000$ was selected as the minimum physically meaningful training length, providing a

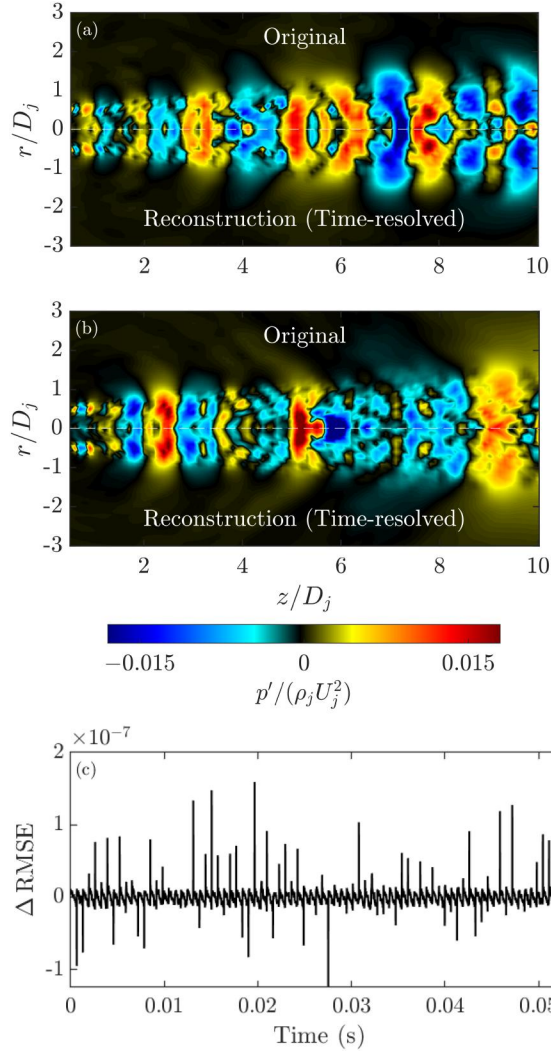


FIG. 5. Verification of the SPOD reconstruction at two selected time instants. The reconstruction is performed using all SPOD modes at all resolved frequencies, so the difference from the reference field measures only the error introduced by the choice of reconstruction approach (time-resolved vs block-wise) and not any modal or spectral truncation. Panels (a) and (b) show two selected time instants in which the upper half-plane ($r/D_j > 0$) is the reference LES pressure field and the lower half-plane ($r/D_j < 0$) is the corresponding time-resolved reconstruction. (c) Time evolution of the normalized RMSE difference $\Delta \text{RMSE} = (\text{RMSE}_{\text{block}} - \text{RMSE}_{\text{time}}) / (\rho_j U_j^2)$ between the block-wise and the time-resolved reconstructions. Values of order 10^{-7} throughout the record confirm that, when the full SPOD basis is retained, the block-wise and time-resolved reconstructions are statistically equivalent at this resolution.

reliable training signal for the leading SPOD coefficient that can be compared against the longer $T = 3000$ window.

E. Phase reconstruction of SPOD signals

A realization of the time-resolved SPOD coefficients obtained from the turbulent jet flow solution can be expressed as an analytical exponential function with amplitude A and frequency f ,

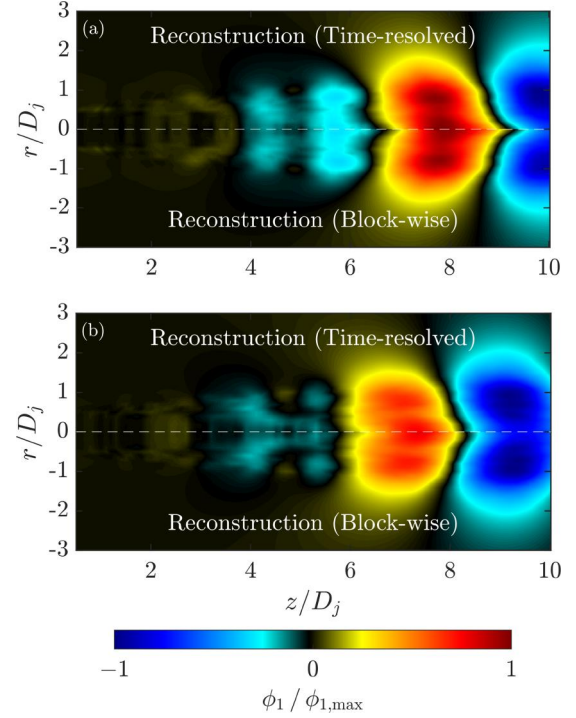


FIG. 6. Low-rank reconstruction of the fluctuating pressure field at two selected time instants [(a) and (b)]. In each panel, the upper half-plane ($r/D_j > 0$) shows the reconstruction obtained from the time-resolved SPOD coefficients and the lower half-plane ($r/D_j < 0$) shows the reconstruction obtained from the block-wise SPOD coefficients; the fields are normalized by $\phi_{1,\text{max}}$. In contrast to Figs. 5(a) and 5(b), where the full SPOD basis was retained, the reconstruction here is restricted to the leading SPOD mode at $\text{St} = 0.1$.

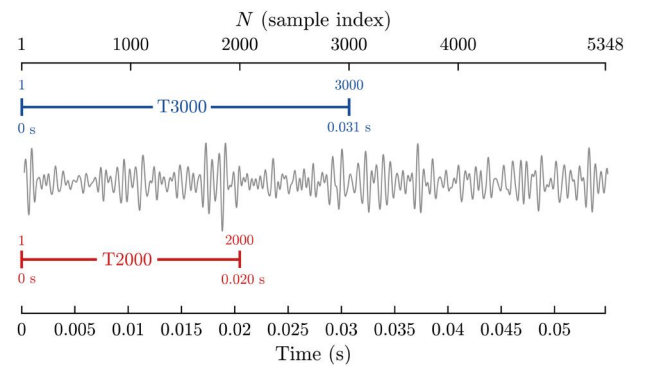


FIG. 7. Time evolution of the real part for the input SPOD time coefficient at $\text{St} = 0.1$ and $m = 0$. Brackets indicate the two training windows, T2000 (2000 samples) and T3000 (3000 samples).

$$a_j(f, t) = Ae^{i2\pi ft} = A \cos 2\pi ft + iA \sin 2\pi ft. \quad (13)$$

From the above-mentioned equation, a realization of the phase can be reconstructed, for example, given by

$$\angle a_j(f, t) = \tan^{-1}(\Im(a_j(f, t)) / \Re(a_j(f, t))), \quad (14)$$

where $\Re(a_j(f, t)) = A \cos 2\pi ft$ and $\Im(a_j(f, t)) = A \sin 2\pi ft$. For stochastic processes representative of turbulence, phase realizations can be ensemble averaged, thereby computing the effective convection phase and the corresponding convection velocity. This ensemble averaged convection velocity has the physical meaning of the propagation speed for turbulent eddies in jet flow and for a turbulent jet it is approximately equal to the local mean flow velocity.^{52–54} In the present analysis, the unwrapped phase of the leading SPOD time coefficient evolves linearly, and the ensemble-averaged SPOD reconstructed phase is in excellent agreement with the analytical prediction $\varphi(t) = 2\pi ft$ (Fig. 8). Using $f = St U_j / D_j$ for the selected Strouhal number $St = 0.1$ and estimating the streamwise wavelength λ_z from the corresponding spatial SPOD mode in the vicinity of the potential core end, the obtained phase speed yields $U_{\text{phase}} = 0.63 U_j$ (Fig. 8). This value is consistent with the convection speed of coherent structures in the jet and supports the interpretation of the leading SPOD mode as a predominantly convecting component whose propagation speed is set by the local mean flow.

However, when reconstructing realizations of the complex phase from the modal coefficients using reduced-order methods, additional errors may arise if the real and imaginary parts of the complex signal are treated as two independent time series. A convenient way to address this inconsistency is to use the Hilbert transform, which can mitigate this error. Under the assumption, that the realization of the modal SPOD coefficient is analytical, Eq. (13) can also be rewritten as

$$a_j(f, t) = Ae^{i2\pi ft} = A \cos 2\pi ft + i\mathcal{H}\{A \cos 2\pi ft\}. \quad (15)$$

Here, $\mathcal{H}$ is the Hilbert transform which for a real valued signal $x(t)$ is defined as its convolution with $1/(\pi t)$,

$$\hat{x}(t) = \mathcal{H}\{x(t)\} = \frac{1}{\pi} \text{p.v.} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau \quad (16)$$

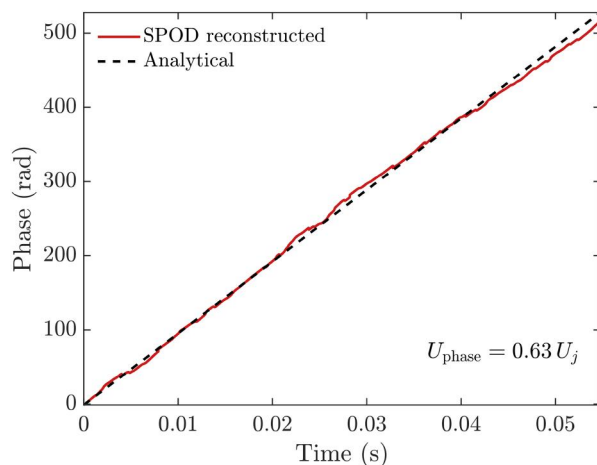


FIG. 8. Temporal evolution of the phase of the leading SPOD mode. Comparison of the reconstructed phase from the complex SPOD time coefficients with analytical prediction $\varphi(t) = 2\pi ft$, where $f = (St = 0.1) U_j / D_j$. The phase speed $U_{\text{phase}} = 2\pi f / k$ is obtained from $k = 2\pi / \lambda_z$, where the streamwise wavelength λ_z is estimated from the spatial SPOD mode distribution in the region around the end of the jet potential core.

and p.v. is the Cauchy principal value.

Therefore, for example, the entire complex signal can be represented as a function of the real part of the SPOD coefficient, according to the following equation:

$$a_j(f, t) = \Re(a_j(f, t)) + i\mathcal{H}\{\Re(a_j(f, t))\}, \quad (17)$$

which only requires a reconstruction of the real part of the turbulent signal, hence, is less demanding for accuracy of the data-driven reduced-order models, which will be considered in Sec. IV.

IV. MACHINE LEARNING METHODS

A. Feedforward neural networks

Feedforward neural networks (FNNs) are a class of artificial neural networks, which possess the properties of universal function approximators. By construction, it is a network of interconnected nodes where each node represents a set of weights and biases, used to obtain a linear combination of the inputs to that node. An activation function then introduces non-linearity to the output and the network as a whole represents a parameterized nonlinear function. The weights and biases are optimized (trained) enabling the network to be fitted onto a dataset of arbitrary dimensionality. Once trained, the network then acts as a smooth differentiable function whose computational cost for forward calculation is much smaller than the training steps. The FNN framework developed in Ref. 1 is applied to reconstruct the modal coefficients of SPOD, which are treated as a solution of a time-domain Ordinary Differential Equation (ODE) problem. In this framework, the ODE plays the role of a dynamical system model implicit in the Navier–Stokes equations. The reconstruction is done by learning the time-gradients of the SPOD mode signals. Following Ref. 1, the input to FNN at each time step is the SPOD time coefficient and the output is the reconstructed time gradient, which is then used to march the signal in time using numerical integration based on a high-order Runge–Kutta scheme. The updated reference solution values are then used for training in the FNN-based regression module, where a random search algorithm was implemented to explore the possible network configurations and find a suitable set of hyperparameters. The random search algorithm was designed to find an optimal network structure that best represents the derivative field in the domain. The degrees of freedom space for the search algorithm was selected to limit the total computational time. For example, the random search algorithm was limited to 100 random models. The range of values allowed for the number of hidden layers was from 6 to 10, the number of elements per layer (layer width) was in the range of 10–64, and the regularization parameter was set in a range of 10^2 – 10^6 .

A total of 2000 time-steps were used to train the model. For each of the random configurations generated, the mean absolute training error in the last training iteration was used as the performance metric. Once the network structure is finalized by the search algorithm by minimizing the mean absolute training error, the chosen network reconstructs the time-gradient values. The latter values are used as the right-hand side of the ODE, which is solved with an explicit fourth-order Runge–Kutta method to advance the signal in time. The above numerical framework, called the Neural ODE, is equivalent to solving a time evolving system,

$$\frac{da_i(f, t)}{dt} = \text{NN}(a_i(f, t)), \quad (18)$$

where $\text{NN}(\cdot)$ is the neural network prediction for the time gradient. Hence, the solution process is equivalent to time integration along a high-dimensional trajectory of the signal coordinates corresponding to the POD mode coefficients.

The stability criterion for this integration is that the system should approach a limit cycle, equivalent to a multi-dimensional dynamical system, which should balance two conflicting requirements as follows: on one hand, to avoid intersection points of the trajectory, which are detrimental to the quality of the training dataset, due to target multiplicity for the same collocation points, the domain should be sufficiently sparse. On the other hand, a largely sparse domain will also leave the network untrained with spurious gradients in the regions that are not explored by the optimizer. These random gradients divert the reconstructed signal from a stable cycle and lead to a divergent signal. In the previous studies, including the application of the FNN model for the reconstruction of bandwidth-filtered time signals from the same jet LES,⁵⁵ a system of Ordinary Differential Equations (ODE) was considered corresponding to the first eight SPOD modes, $a_j(f, t)$, $j = 1, \dots, 8$ as a compromise between the complexity of the system and time-integration stability. In the current work, the same approach is implemented to solve an ODE system for the first eight SPOD modes to reconstruct the original broadband signals from the jet LES at a frequency of $\text{St} = 0.1$.

B. Genetic Programming

Genetic Programming (GP)^{45,46,56,57} is an approach that originates in the class of Genetic Algorithms (GA) with the goal of addressing symbolic regression problems. While GA typically uses a string of numbers to represent the solution, GP creates a population of models, usually in the shape of graphs or trees, which are iteratively evolving and improving through the application of the genetic operators. The ultimate goal of GP is to solve an approximation problem by searching through highly fit models in the space of all possible approximations that can be generated by combining variables and functions provided by the user, called “primitives” or “building blocks.” If random sampling and clustering of primitives result in poor exploration of the space of all possible approximations, GP relies on an iterative process inspired by natural evolution to steer the assembling of new models (so-called search in the model space) toward expressions that might minimize the selected error metrics, e.g., RMSE, on the training dataset. By definition, models returned by GP are symbolic expressions and are global, i.e., they apply on the whole variable range defined by the training dataset. The initial step of a GP run is the creation of the initial population of approximations, which is performed either randomly or deterministically by exploiting preexisting knowledge.⁵⁶ In the former case, the generation of GP models (or “trees”) varying in shape and size is based on random or pseudo-random events to extensively explore the space of models defined by the building blocks/primitives provided by the user. In the latter, existing approximations are used to “seed” the initial population with the intention that the GP might further refine them. Once the initial population is generated, the evolution of a highly fit model proceeds through the iterative action of genetic operators and model evaluation by a fitness function, which determines the quality of the evolved

model and selects the set of models that undergo the next round of genetic operations. This procedure leads to a stable approximation provided that the primitives are selected correctly. For example, addition, subtraction, sine, and cosine allow for a bounded model to be extrapolated.

The fitness of models, which is a quality score typically based on RMSE or Mean Square Error (MSE), has a dominant role in steering the evolution toward better and better solutions. It depends directly and critically on the numerical coefficients of the models. Therefore, it is of special importance to distinguish the mathematical structure of a model as a separate object, defined as the model tree stripped of the numerical coefficients. As an example, Fig. 9(a) shows the mathematical structure of the complete model

$$c_3 + (c_1 \cdot z_1 / z_2 + c_2 \cdot z_3)^2, \quad (19)$$

where c_1 , c_2 , and c_3 are the numerical coefficients of the model.

The Genetic Programming method used in this work is the HyGP implementation developed in Refs. 46 and 57. Still founded on the basic principles and guidelines provided by Ref. 45, this HyGP implementation can be classified into the subcategory of “hybrid GP” or “memetic GP” as a result of its additional features, such as the use of Baldwinian learning.^{58,59} In accordance with this approach, classic genetic operators, such as reproduction, mutation, and crossover are still used, but are applied on “parameterless models,” i.e., the mathematical structures of the approximations, such as in the example shown in Fig. 9(b), as opposed to the standard GP methods which apply genetic operators to full models as the one in Fig. 9(a). Consequently, the numerical coefficients are not modified by genetic operators and are instead optimized independently. Optimization of the numerical coefficients of HyGP is performed through a separate two-step process: first by using a population-based optimization

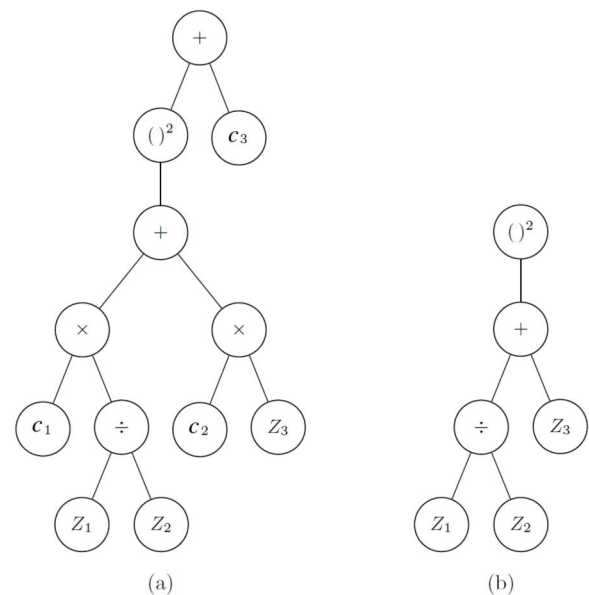


FIG. 9. Example of a GP model and its mathematical structure: The corresponding full model (a) and structure of GP model where the tree is in a “parameterless” state (b).

algorithm (Particle Swarm Optimization⁶⁰) to robustly explore the search space and then by applying a deterministic method (Sequential Quadratic Programming or SQP⁶¹) to fine-tune the values of the parameters. A schematic description of the logical steps featured in the suggested HyGP algorithm is shown in Fig. 10. Additional discussion and information about Genetic Programming, including background theory, can be found in Refs. 56 and 57.

The current HyGP architecture uses 23 hyperparameters, most of which were set after a short series of trial runs for the test cases considered in the present study. The optimal definition of the objective coefficients of the fitness function, together with the choice of these hyperparameters, is a feature of the present HyGP implementation that relies heavily on the user's input. Formal approaches are available to study the sensitivity of the HyGP results to changes in the hyperparameters and to optimize them, thereby reducing the dependence on user experience; they were, however, not applied in the present study.

C. Reservoir computing and echo state networks

In recent years, Reservoir Computing (RC) has garnered interest as an ML approach for building simple models and forecasting the time and space evolution of reconstructed dynamical systems. RC algorithms utilize a recurrent neural network (RNN) comprising interconnected neurons known as the reservoir. The reservoir is composed of an input layer that provides the observed data and an output layer that evaluates the network states.³⁶ Echo State Networks (ESNs) introduced in Refs. 37, 62, and 63 deploy a high-dimensional numerical oscillator called the reservoir, similar to RNN. Given a data point from the input time series (the driver signal), the reservoir matrix transforms the data into features which are then readout and weighted to synthesize an output signal. The feature space in a reservoir consists of the outputs for each of the recurring layers. The teacher signal is returned to the output neurons, which drive the reservoir using the output feedback mechanism. The reservoir then runs recurrently to generate feature outputs in each of its recurring layers, which are then readout by the same output layer. It has been shown that after an initial transient, the output layer tends to adapt the feedback to the reservoir in such a way that the internal recurrent layers echo the relevant features for the teacher signals.⁶² Unlike conventional neural networks, where neuron activation depends on weighted inputs, ESNs generate feature outputs recurrently, resembling an artificial dynamical system whose state spaces are sampled to approximate a given signal.

The ESNs implemented in this study follow the principles and methods described in Ref. 64 on deploying ESNs with leaky integrator neurons. The structure of the ESN consists of an input layer, a reservoir with N neurons, and an output layer. The appropriate size of the reservoir (N) should consider both the length of the training data (T) and the complexity of the task. Generally, N falls within the range of $T/10$ – $T/2$.⁶⁵ The input layer receives the input signals and the output layer generates the output signal. For testing, a network with 200 reservoir neurons and 20% sparsity has been used.

The selection of the ESN hyperparameters was based on established defaults in the literature for leaky-integrator ESNs,^{64,66} refined by a short series of trial runs to confirm their suitability for the present dataset. Several studies^{67–69} have proposed automated optimization methods, including Bayesian optimization, to refine ESN hyperparameters

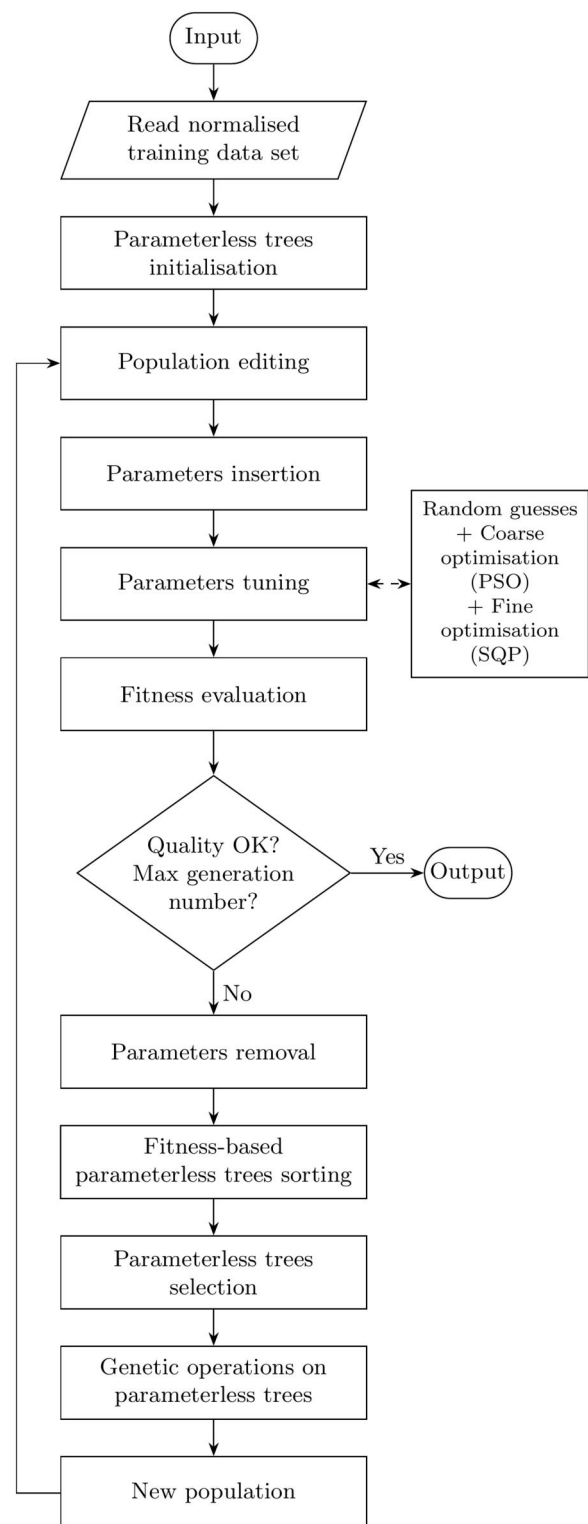


FIG. 10. Flowchart of HyGP algorithm.

further. These techniques, however, do not provide a comprehensive sensitivity analysis and proved unstable for the present prediction task.

A key control parameter is the spectral radius (the maximum eigenvalue of the reservoir matrix W), which must be smaller than unity to satisfy the echo-state property;⁶⁴ in the present configuration, it is set to 0.99. The leakage rate, which controls the speed of internal state evolution, is set to 0.1. The reservoir size N also affects the model capacity,⁶⁴ with larger N risking overfitting and higher computational cost. The value $N = 200$ neurons with 20 % sparsity adopted here was found to give the best trade-off between predictive accuracy and trainability on the available compute. A short sensitivity scan around the operating point indicates that the spectral radius is the most influential of these parameters: higher values significantly reduce stability, while lower values reduce accuracy and lead to premature damping of the predicted signal.

For the driver signal of the ESN method, a bandpass-filtered white noise is used. During the training step, the input layer receives the driver signal, which is sufficiently broadband to be adequately processed by the ESN algorithm using the SPOD coefficients of the jet data as teacher signals. Training, thus, becomes nearly equivalent to a least squares fit of the teacher signal in the white-noise-provided latent space. Once fully trained, this allows the same bandpass-filtered white noise driver signal of arbitrary length to drive the ESN over extended periods, with the output beyond the initial training dataset interpreted as the prediction.

D. Transformer-based neural network

For a physical field/variable, seeking a mapping function between two consecutive time snapshots is a key step toward “learning” the temporal evolution. With recent developments in the field of deep learning, such correlation can be sought using a transformer-based attention mechanism, which has been proven to be computationally efficient compared to other Artificial Neural Network methods.^{70,71}

In the present work, a kernel function K_t is employed to realize the mapping between the two time instances, and the kernel that evaluates the high-dimensional similarity (i.e., the mapping function) is approximated using a Transformer architecture.⁴² Figure 11 illustrates the workflow of the Transformer-based model for the prediction of the SPOD coefficients. For a given time series, x_t , it first undergoes a lifting to the required dimension, $z_t(x)$, which generates the input feature for the transformer.^{72,73} Subsequently, a trainable kernel integration is performed to obtain the next time instance, i.e., $K_t \cdot z_t(x) \rightarrow z_{t+1}(x)$, and hence completing the prediction. In order to implement the kernel operation in the transformer model, where the evaluation of high-dimensional similarity is realized through the attention matrix, the kernel integration can, thus, be decomposed into *Query* vector, $Q = W_Q z_t$, *Key* vector, $K = W_K z_t$, and *Value* vector, $V = W_V z_t$.⁴² Here, W_Q , W_K , W_V represent learnable weight matrices. These vectors are then fed into the self-attention mechanism in the transformer models, where the computation of the attention scores relies on the dot product between Q , K^T and V as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_z}}\right)V, \quad (20)$$

where d_z denotes the input feature dimension of z_t . The *softmax* function is used to normalize the attention scores in all time instances for the prediction. Finally, the predicted outcome from the transformer model, $z_{t+1}(x)$, is projected back to the physical space to yield, x_{t+1} . Since the objective of the Transformer training is to minimize the prediction error from the training inputs, the loss function of the transformer can be defined as

$$\mathcal{L}_{\text{transformer}} = \mathbb{E}[|z_{t+1} - \hat{z}_{t+1}|^2], \quad (21)$$

where $\hat{z}_{t+1}$ represents the prediction of the transformer and z_{t+1} denotes the ground truth for the next instance.

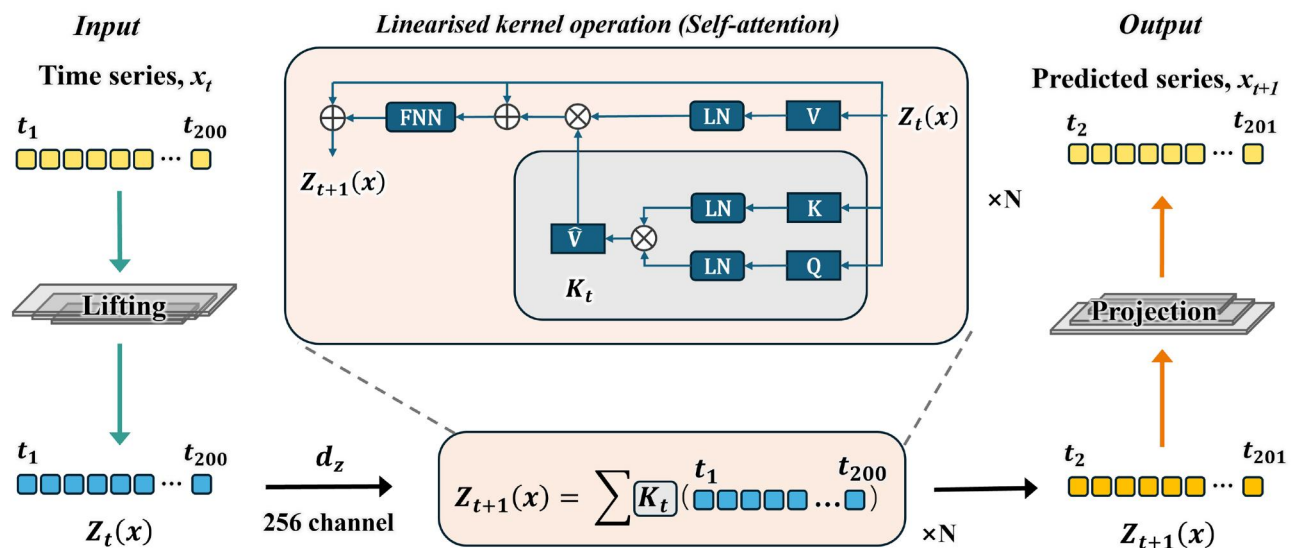


FIG. 11. Schematic of the transformer-based neural network.

In the present implementation, both the lifting and projection, which map the physical space to the hidden high-dimensional one and vice versa, are carried out using a pre-trained convolutional neural network.^{72,73} It is useful to note that there is no particular preference to a neural network structure for this mapping procedure in the current study, however, further investigation can be conducted to optimize this step in terms of prediction accuracy and computational efficiency. For the transformer model, the multi-head attention mechanism is adopted,⁴² which subdivides the Query, Key, and Value into 16 attention heads, each possessing a feature dimension of 16, thus giving rise to $d_z = 256$. This configuration allows the model's capacity to process complex correlation between the time instances of the input. To further verify the choice of the Transformer architecture, an ablation study has been carried out to examine the effect of varying the number of attention heads and the feature dimension of each head. Note that d_z is kept at 256 by convention. Although not shown here for the sake of brevity, the ablation results indicate that the model performance remains essentially comparable when increasing either the number of attention heads or the feature dimension by twofold. As a result, the present transformer model adopts an equal split of 16×16 for d_z . There are a total of 2×10^6 trainable parameters in this model and the selection of hyperparameters is based on previous studies which have demonstrated promising capability on complex tasks.^{42,74,75} The trainings are run for 1000 epochs with an input size of 200 (i.e., $x_t = [x(t_1), x(t_2) \dots x(t_{200})]$).

V. RESULTS

All four machine-learning methods considered have been applied to reconstruct the time coefficient of the first SPOD mode corresponding to the dynamically important mode at a frequency corresponding to $St = 0.1$. The LES dataset consists of $N = 5348$ snapshots of the leading SPOD time coefficient $a_1(t)$. For each method, two training windows are considered, $T = 2000$ and $T = 3000$ samples (Fig. 7). The remaining $N - T$ snapshots (3348 for $T = 2000$ and 2348 for $T = 3000$) form a strictly unseen test segment on which every metric reported in this section is evaluated. The same train/test partition is applied to every model, so differences in the reported performance reflect the methods themselves rather than the data they were trained on.

In contrast to the GP, ESN, and Transformer methods, which have proved themselves sufficiently robust to result in stable solutions, the FNN model failed this test, leading to a divergent solution for the SPOD coefficients (Fig. 21 in Appendix A). This is likely due to a very broadband nature of the SPOD signal corresponding to the high-Reynolds number jet considered, in-line with the previously reported strong sensitivity of the FNN model's performance to the bandwidth of the considered turbulent signal.⁵⁵

Performance of the three other ML models, which have remained reasonably valid, is further compared according to the four criteria: (i) convergence rate, (ii) qualitative and quantitative comparison with the ground truth LES signal using the RMSE, (iii) spatial correlation of the reconstructed reduced-order spatiotemporal pressure distribution with the corresponding ground truth LES solution, and (iv) spectral consistency of the reconstructed PSD coefficient with the reference LES signal in terms of the power spectral density and the average phase. In addition, the training and computational costs of the three methods are reported in Appendix B.

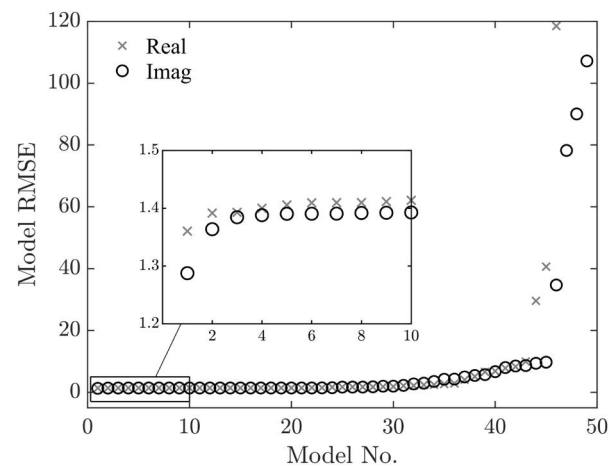


FIG. 12. (a) Root-mean square error of the initial GP populations (generation 0) evaluated on the training dataset and (b) training data RMSE of the best ten GP models at generation 0.

A. Convergence of the model training

For all three ML methods which produce stable solutions for the SPOD time coefficient, GP, Transformer, and ESN, the decay of the loss function in terms of Root-Mean-Square-Error (RMSE) has been evaluated for the training dataset corresponding to 2000 time samples. The convergence for the training over 3000 samples shows essentially the same trends. In most cases, the model results are shown only for the real part of the signal, since the convergence results for the imaginary part (in those simulations without Hilbert transform) are very similar.

In the case of GP, the algorithm involves two stages: the initial generation step and the subsequent training involving the model evolution. Figure 12(a) shows the RMSE of two specific initial populations of GP models before evolution begins (initial generation). At this stage, the evolutionary engine has randomly generated the models assembling the building blocks provided by the user only and keeping optimized their numerical coefficients minimizing the models' RMSE on the training dataset. The training RMSE is plotted for each of the 50 randomly seeded models after sorting, so that the RMSE curve increases monotonically, and the best model is always the first. Figure 12(b) provides more detail on the first ten best-performing GP models of the initial population, being a magnified excerpt of the first entries in Fig. 12(a). The two orders of magnitude RMSE variation between the best and the worst performing of the 50 models, as shown in Fig. 12, provides an initial evaluation of the optimization capacity of the population-based approach of GP, prior to any recombination of the model structures via genetic operators. The convergence of the evolutionary stage of GP is shown in Fig. 13. The drop in RMSE of different models throughout the generations is plotted here from the initial model (generation 0) to the last one. In the figure, each gray curve represents the best performing model retained in the archive of each independent GP run, while the black curve denotes the average RMSE of all the best performing models across all runs. Note that each GP simulation consists of ten independent runs. In general, the reduction in RMSE of the GP model during the two stages of the training phase corresponds to 2–3 orders of magnitude.

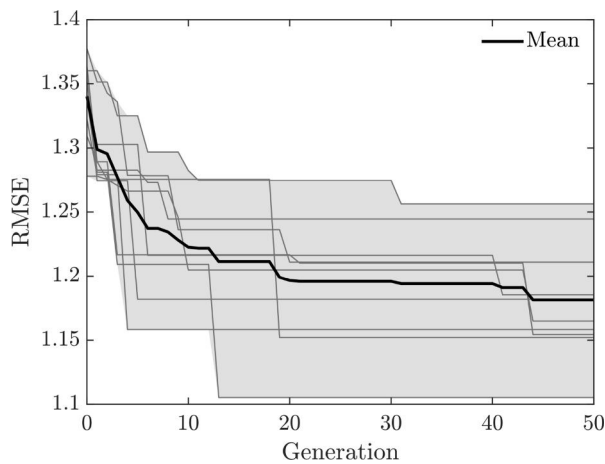


FIG. 13. Convergence of the minimum RMSE through GP archives using Strategy 1 vs generation; Gray lines correspond to individual runs and the black line shows the mean across all runs.

The convergence results of Echo State Networks (ESNs) is shown in Fig. 14. Similarly to GP, ESN generates a stochastic signal that is not designed for exact fitting of the target signal in the time domain. Unlike the current GP implementation based on a series of combinations of harmonic functions, ESN is not tailored for periodic signals but rather for a chaotic signal reconstruction. However, in the context of the current turbulent signals, due to their stochastic nature, the GP and the ESN approaches have similarities in the model setup stages. For example, like GP, the ESN model involves an initial configuration stage, where its main hyperparameters are specified.⁶⁶ Having completed the initial configuration, the ESN training procedure involves a separate run to further optimize the weight coefficients, which takes between 3000 and 4000 epochs to further reduce the RMSE loss function by a factor similar to that of the GP case (see Fig. 13). Compared to the GP and ESN models, the transformer model is focused on

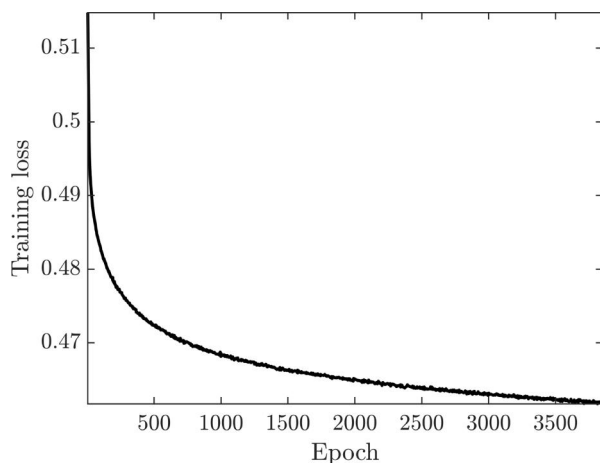


FIG. 14. Convergence of Echo State Networks (ESNs) during training.

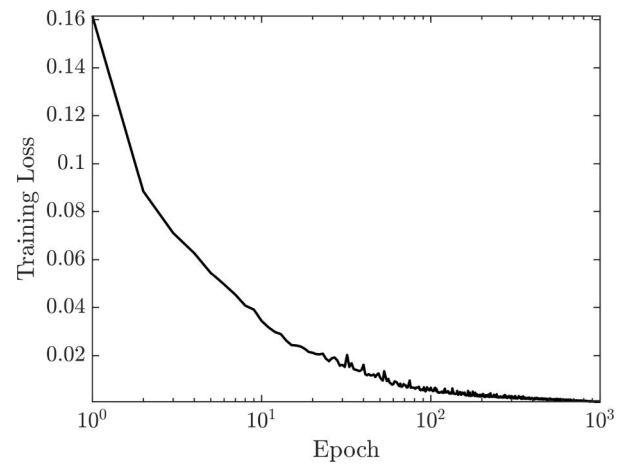


FIG. 15. Convergence of the transformer model during training.

exact replication of the training signal in the time domain within a single calibration run procedure. Figure 15 illustrates the training convergence of the transformer model, where the training loss defined in Eq. (21) is plotted against the number of epochs. The training convergence obtained from a single run shows an initial rapid decrease in the training loss up to approximately 100 epochs, followed by a slower convergence. The total loss reduction over 1000 epochs corresponds to 2–3 orders of magnitude in comparison with the initial residual. The convergence trend observed in the present study using transformer agrees well with those reported in the literature.^{42,44}

B. Comparison with the ground-truth LES signal

Detailed comparison of time-series reconstructed using different ML models with the ground truth SPOD time signal is provided in Appendix A. It can be seen that after the training period all models provide stable solutions capturing the signal variability quite well for both 2000 and 3000 initial snapshots used for training. Interestingly, compared to the ground truth, the temporal coefficient predicted from the transformer model shows notably smaller variations in amplitude, especially for the longer training period. This is in agreement with the expected behavior of transformer, since their predictions are based on the most probable outcome given the training input. This helps to keep the amplitude within the expected range but may lead to amplitude modulations over a longer period and some underprediction of the peaks⁷⁰ (see Appendix A).

For quantitative comparison of the accuracy of the ML models, the Root Mean Square Errors (RMSE) between the ground truth signal (LES) and the data-driven reconstructions are computed. Results of this comparison are shown in Fig. 16, which includes the distribution of the RMSE over time for all models and training durations (2000 and 3000 snapshots) using box plots (25th–75th percentiles with 10th/90th percentile whiskers).

For GP, results for two variants are shown, which differ by definition of the loss function: Strategy 1 is purely based on RMSE and Strategy 15 based on a multivariate loss functions. The RMSE distributions show that GP Strategy 1 consistently achieves low median errors

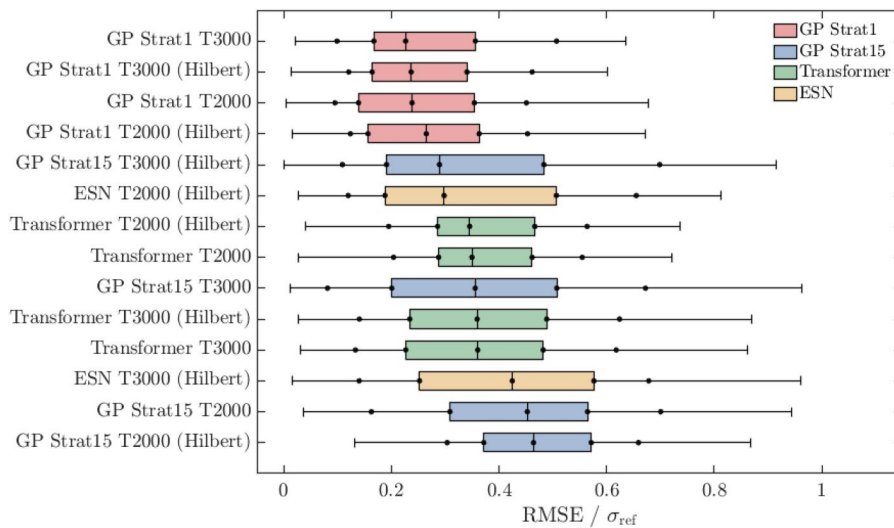


FIG. 16. Distribution of the normalized RMSE, $\text{RMSE}/\sigma_{\text{ref}}$, for the different ML methods and training durations, where σ_{ref} is the temporal standard deviation of the reference leading-SPOD time coefficient. Each box plot summarizes the RMSE using the 10th, 25th, 50th (median), 75th, and 90th percentiles.

and relatively tight inter-quartile ranges for both training lengths, particularly when combined with the Hilbert formulation. In contrast, GP Strategy 15 exhibits a markedly broader spread and, in some configurations, higher median errors, especially for the longer training window and when the Hilbert transform is applied. In these cases, the performance of Strategy 15 becomes comparable to, or worse than, that of the Transformer and ESN models. Transformer rank second overall with longer training (3000). The use of the Hilbert transform to skip the reconstruction of the imaginary part of the signal leads to some reduction in both the median error and variability. For ESN, only the version of the model with the Hilbert transform is shown, since the results of the model without it are very similar. The ESN results show the highest errors and the widest spreads, suggesting the presence of a higher noise compared to those obtained by GP and transformer.

C. Correlation analysis of the reconstructed pressure solutions

To analyze accuracy of the reduced-order modeling for pressure fluctuations of the turbulent jet, the SPOD coefficients reconstructed using the three ML methods are combined with the corresponding modes. The resulting spatiotemporal distributions representing pressure fluctuations of the dominant part of the axisymmetric pressure mode at frequency $\text{St} = 0.1$ are compared with the reference solution extracted from LES.

Due to the stochastic nature of the compared distributions, to separate structural agreement from amplitude bias, the comparison between the reconstructed and the reference solution is performed in terms of the spatial correlation coefficient. It can be noted that, while the RMSE emphasizes point-wise deviations in time coefficient amplitude, the correlation coefficient measures the linearity in the dependence between spatial distributions independent of magnitude scaling. The spatial correlation that is insensitive to uniform scaling and highlights phase-structure alignment is defined as

$$C(t) = \frac{\sum_{i=1}^{n_z} \sum_{k=1}^{n_r} p_0^{\text{ref}}(z_i, r_k, t) p_0^{\text{rec}}(z_i, r_k, t)}{\sqrt{\sum_{i=1}^{n_z} \sum_{k=1}^{n_r} (p_0^{\text{ref}}(z_i, r_k, t))^2} \sqrt{\sum_{i=1}^{n_z} \sum_{k=1}^{n_r} (p_0^{\text{rec}}(z_i, r_k, t))^2}}, \quad (22)$$

where $p_0^{\text{ref}}(z_i, r_k, t)$ and $p_0^{\text{rec}}(z_i, r_k, t)$ correspond to the reference and ML-reconstructed pressure fluctuation fields of the axisymmetric mode in the first SPOD mode and the frequency $\text{St} = 0.1$.

The pressure fields considered have a wavepacket structure as illustrated in Fig. 17(a), which shows snapshots of the reference solution plotted in the upper half of the plane ($r/D_j > 0$) and the reconstructed solution in the lower half of the plane ($r/D_j < 0$) using the GP model (Strategy 1 with 2000 time samples as training dataset) at several time moments. Figure 17(b) shows the evolution of the correlation coefficient as a function of time between the GP reconstructed solution and the reference LES solution. Here, positions of the marker in each row correspond to the same time moment as the snapshot shown in the left panel. For example, the top row ($C \approx -1$) shows that the GP-reconstructed solution has the same structure as the reference fields but is aligned in anti-phase. The middle row ($C \approx 0$) corresponds to the case of spatial misalignment with a near-zero correlation between the GP solution and the reference. The bottom row ($C \approx 1$) shows a strong positive correlation, where the location and phase of the reference and the reconstructed wavepacket are closely matched.

The same correlation analysis has been performed for all three methods, GP, ESN, and Transformer for two training windows (2000 and 3000 snapshots) with and without using the Hilbert transform to analytically reconstruct the imaginary part of the SPOD signals. The results are compared in Fig. 18, which presents an integral view of how long the reconstructed solutions remain strongly correlated ($C > 0.5$) with the reference LES solution.

For ESN, only the Hilbert-based formulation is shown (the non-Hilbert results are similar). In contrast to the earlier trend suggested by the RMSE statistics (Fig. 16), Fig. 18 shows that ESN

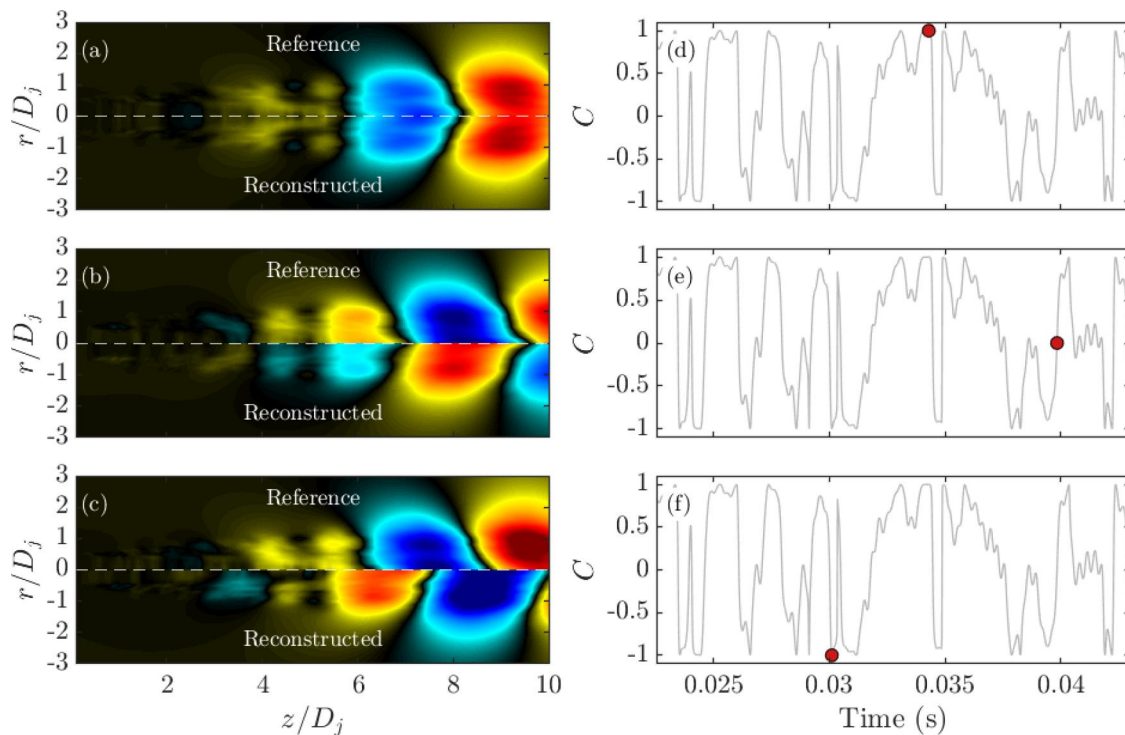


FIG. 17. Instantaneous snapshots of the predicted and reference solutions at three representative correlation levels: (a) and (d) strongly positive correlation ($C \approx 1$), (b) and (e) near-zero correlation ($C \approx 0$), and (c) and (f) strongly negative correlation ($C \approx -1$). The reference field is shown in the upper half-plane ($r/D_j > 0$) and the reconstructed field in the mirrored lower half-plane ($r/D_j < 0$). The right panels indicate the evolution of spatial correlation with time, with markers highlighting the selected correlation levels at the corresponding time instants.

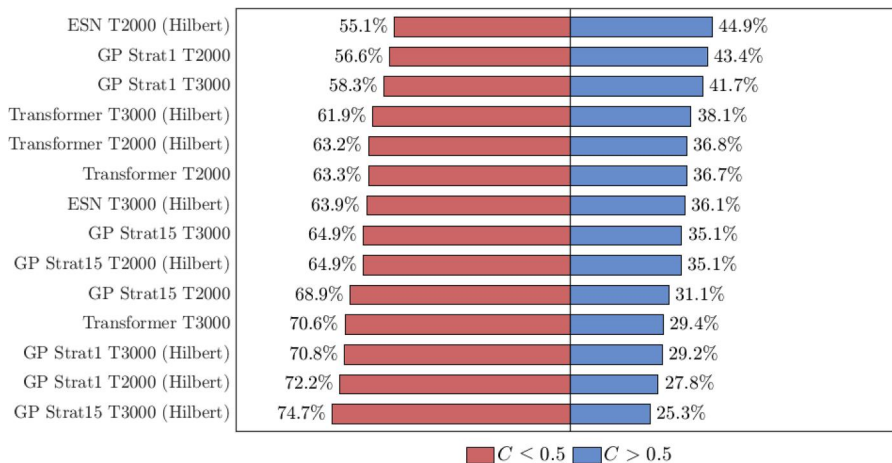


FIG. 18. Fraction of spatial positions falling into strong positive correlation $C > 0.5$ (blue) and strong negative correlation $C < 0.5$ (red), for each reconstruction model. Percentages within each bar indicate the coverage of the spatial field by the corresponding correlation regime, highlighting differences in model performance and the presence of highly correlated or anti-correlated regions.

achieves the largest fraction of positively correlated points for the shorter training window ($T2000$), with 44.9% of the spatial field satisfying $C > 0.5$. This coverage decreases for longer training window ($T3000$), indicating a non-monotonic sensitivity to training length.

For the transformer, the fraction of strongly correlated points is moderate ($\approx 36 - 37\%$) for the $T2000$ cases, whereas for the longer

training window the Hilbert option is beneficial, increasing the ($C > 0.5$) coverage from 29.4% to 38.1%.

For GP, Strategy 1 (RMSE-based loss function) provides among the best correlation coverage when trained directly on the complex coefficient, reaching 43.4% and 41.7%. However, introducing the Hilbert formulation in Strategy 1 leads to a marked reduction in the spatial correlation fraction ($\approx 28\% - 29\%$), suggesting more frequent

local phase inconsistencies and spatial misalignment relative to the LES reference. Strategy 15 generally yields lower correlation coverage than Strategy 1, with the weakest performance observed for GP Strat15 T3000 (Hilbert). Overall, these trends indicate that the choice of loss function, training window, and the Hilbert transform option can considerably affect the spatial extent of strongly correlated regions.

D. Spectral and phase analyses

Figure 19 present a comparison between the Power Spectral Density of the SPOD signal corresponding to the dominant frequency mode from the LES solution and the spectra reconstructed by the GP, ESN, and transformer models. Results are shown for different training-window lengths, with and without the use of the Hilbert transform. For GP, results for both the RMSE-based Strategy 1 and the multivariate Strategy 15 are included for completeness. Note that the ground truth LES spectrum shows a broad peak at $St = 0.1$ in accordance with the selected central SPOD frequency. GP-reconstructed spectra follow the low-frequency plateau and the initial roll-off closely compared to the reference spectrum. Extending the training window from 2000 to 3000 time samples yields slightly improved model prediction in the mid-frequency band. The influence of the Hilbert transform on the reconstructed spectra is frequency-dependent. While it improves amplitude agreement at low frequencies around the dominant peak and, in most cases, in the mid-frequency range, it also leads to a noticeable deterioration at higher frequencies, where increased spectral leakage is observed. This effect is particularly pronounced for the GP reconstructions. Transformer produce the best agreement with the reference over the widest frequency band out of all methods. Although the ESN solutions agree well with the reference spectrum around the

low-peak frequencies, they fail to reproduce the fast roll-off behavior of the LES solution at the mid- and high-frequencies.

To complement the PSD comparison in Fig. 19 and provide a single quantitative measure of how well each reconstruction matches the reference spectrum, the integrated PSD error is as follows:

$$\varepsilon = \frac{\int |S_{\text{rec}}(St) - S_{\text{ref}}(St)| dSt}{\int S_{\text{ref}}(St) dSt}, \quad (23)$$

computed on the linear power spectral density and integrated over the full resolved Strouhal range. Table II shows ε for all reconstructions.

It can be noted that the three diagnostics, the normalized RMSE in Fig. 16, the integrated PSD error ε Table II, and the correlation fraction in Fig. 18, rank the reconstructions differently because each is sensitive to a different aspect of the signal. The RMSE penalizes both amplitude and phase deviations in the time domain, integrated PSD error ε depends only on the spectral envelope and is, therefore, phase-blind, while spatial correlation C is amplitude-normalized but phase-sensitive. Taken together, these three views expose three distinct types of reconstruction behavior. GP Strategy 1 trained directly on the complex coefficient achieves the lowest RMSE and the highest correlation coverage but the largest ε . The GP Strategy 15 reconstructions with the Hilbert transform show the opposite signature, with the smallest ε at the cost of the largest RMSE and only moderate correlation coverage, indicating that the Hilbert transform recovers the spectral peak amplitude accurately while degrading phase coherence. The Transformer family and ESN T2000 (Hilbert), in contrast, occupy a balanced regime with moderate values across all three metrics.

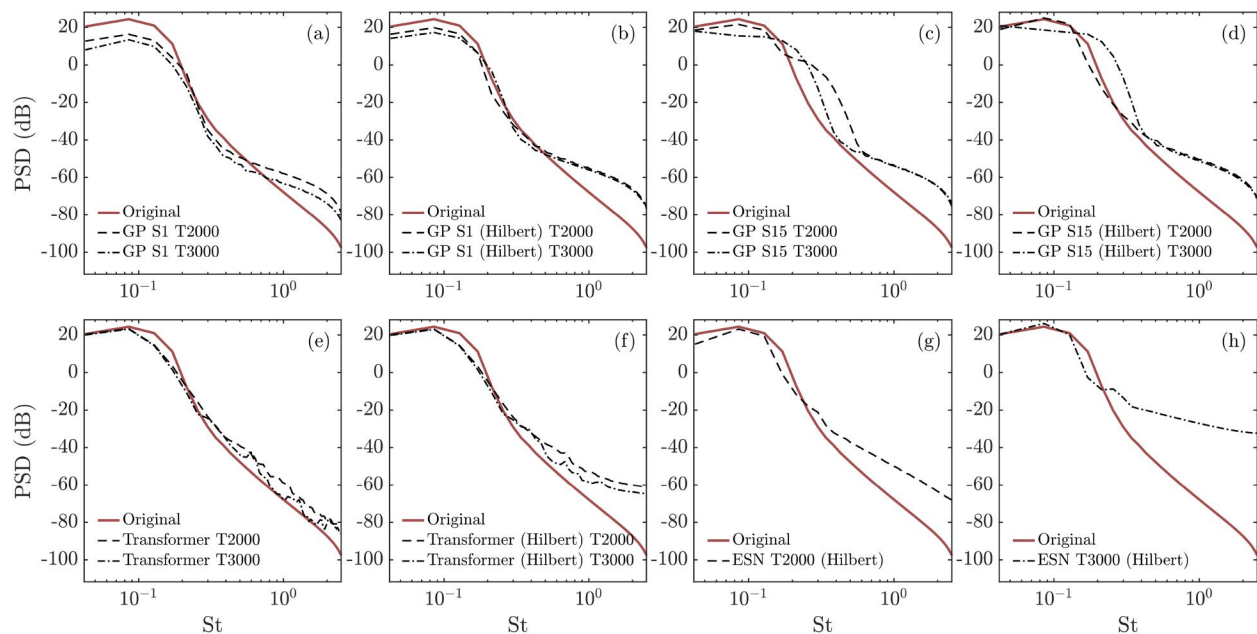


FIG. 19. Power spectral density (PSD) of the reference signal and the reconstructed signals for: (a) GP Strategy 1, (b) GP Strategy 1 with the Hilbert transform, (c) GP Strategy 15, (d) GP Strategy 15 with the Hilbert transform, (e) the transformer, (f) the transformer with the Hilbert transform, (g) the ESN model with the Hilbert transform for $T = 2000$ samples, and (h) the ESN model with the Hilbert transform for $T = 3000$ samples.

TABLE II. Integrated PSD error ε of Eq. (23) for each model reconstruction, computed for the leading SPOD coefficient and integrated over the full resolved Strouhal number range. Rows are sorted in the ascending order of ε .

Model	ε
GP S15 T2000 (Hilbert)	0.224
ESN T2000	0.340
Transformer T2000	0.355
ESN T3000	0.360
Transformer T2000 (Hilbert)	0.389
Transformer T3000	0.392
Transformer T3000 (Hilbert)	0.416
GP S15 T2000	0.459
GP S1 T2000 (Hilbert)	0.639
GP S15 T3000 (Hilbert)	0.704
GP S15 T3000	0.789
GP S1 T3000 (Hilbert)	0.791
GP S1 T2000	0.840
GP S1 T3000	0.923

The final challenge for the data-driven models considered is the reconstruction of the average phase and its comparison with the unwrapped phase extracted from the ground truth SPOD signal. As discussed in Sec. III D, an approximately linear dependency of the unwrapped phase from the LES signal is consistent with the travelling-wave behavior of coherent structures in the turbulent jet, which propagate at a phase speed approximately equal to the local jet

flow velocity. It is, thus, important that reduced-order models capture this pertinent feature of the high-speed jet flow if they are to be used as valid surrogates to ultimately replace LES.

Figure 20 compares the unwrapped phases of the ML models considered with different training windows with the reference LES solution. Quantitative convection-speed estimates extracted from the linear slope of the unwrapped phase for each model reconstruction are summarized in Table III, with the reference LES value $U_{\text{phase}}/U_j = 0.63$. Without applying the Hilbert transform, which automatically makes the imaginary and real parts of the reconstructed signal consistent in terms of the average phase, all GP models show large errors [Figs. 20(a) and 20(c)] in comparison with their Hilbert-based counterparts [Figs. 20(b) and 20(d)]. Transformer predictions [Figs. 20(e) and 20(f)] are found to be quite insensitive either to the length of the training dataset or to the use of the Hilbert transform and are in reasonable agreement with the reference phase solution. The ESN solutions without the Hilbert transform show a behavior very similar to GP and are not included here. Applying the Hilbert transform significantly improves the phase predictions using GP and ESN. The GP solution based on the multivariate Strategy 15 with Hilbert [Fig. 20(d)] shows greater deviation from the ground truth in comparison with the RMSE-based Strategy 1 with Hilbert [Fig. 20(b)]. This is consistent with the less coherent behavior of the GP based on Strategy 15 revealed during the correlation analysis. The ESN model based on a shorter training dataset of 2000 snapshots [Fig. 20(g)] shows some moderate deviation from the reference. At the same time, the phase reconstructed using the ESN model with 3000 snapshots [Fig. 20(h)] shows a remarkable agreement with the ground truth. However, it can be recalled that the predictions of the ESN model for

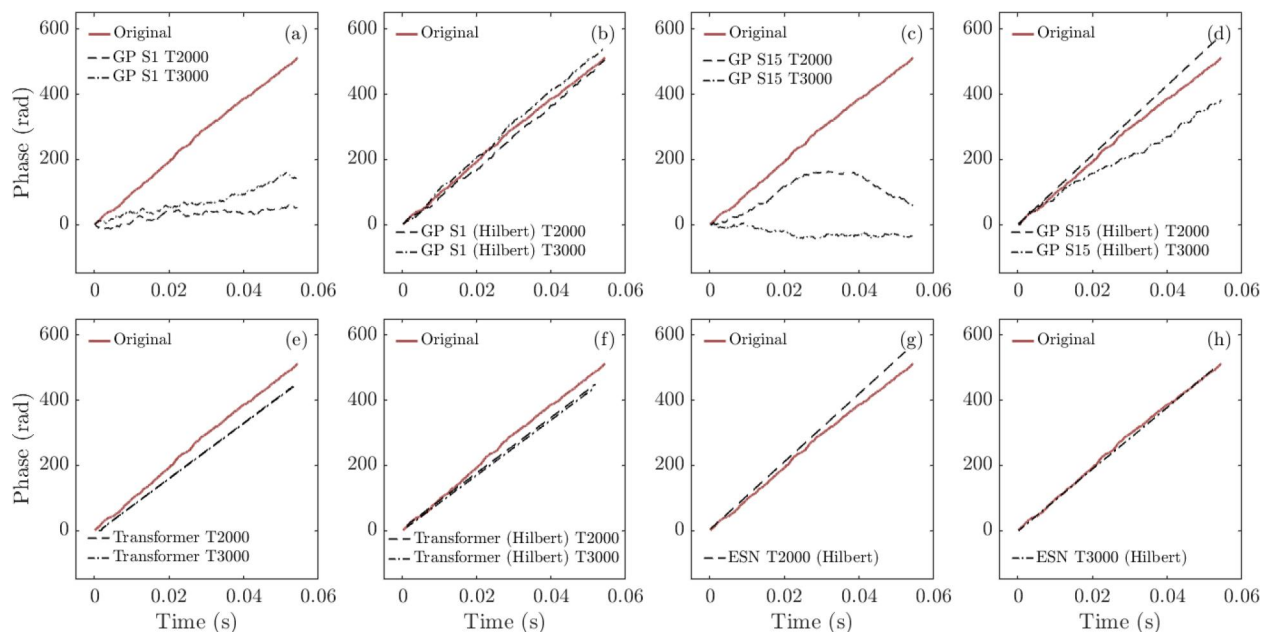


FIG. 20. Unwrapped phase of the reference signal (solid red line) and the reconstructed signals (black dashed line for $T = 2000$ samples, black dashed-dotted line for $T = 3000$ samples) for (a) GP Strategy 1, (b) GP Strategy 1 with the Hilbert transform, (c) GP Strategy 15, (d) GP Strategy 15 with the Hilbert transform, (e) the transformer model, (f) the transformer model with the Hilbert transform, (g) the ESN model with the Hilbert transform for $T = 2000$ samples, and (h) the ESN model with the Hilbert transform for $T = 3000$ samples.

the longer training dataset correlated with ground truth less well compared to the shorter training dataset in terms of the overall spatial solution structure. Together, this may suggest that the accuracy of the ESN model is sensitive to the training data and potential overfitting issues.

VI. CONCLUSIONS

Reconstruction of the most energetic component of the fluctuating pressure field in a high-Reynolds number jet flow, which is critical for peak noise radiation at small angles to the jet flow axis, is proposed as a benchmark problem for Machine Learning (ML) methods. The training data are derived from a Large Eddy Simulation (LES) validated against the NASA SHJAR jet experiment at a jet Mach number of 0.9. A key challenge for ML-based reconstruction lies in the noisy, broadband nature of scales typical of LES solutions for high-speed jets. The dominant pressure fluctuations are extracted using the frequency-domain Spectral Proper Orthogonal Decomposition (SPOD) method, with careful verification of the time-resolved SPOD signal against block-wise reconstructions.

While, in this study, the ML models are applied to reconstruct a dominant part of the axisymmetric pressure mode known to be crucial for peak jet noise at low frequencies, our ultimate goal is to develop a ML framework that would replace a significant part of the LES flow solution process for all frequencies of interest with much faster data-driven reduced-order models.

This goal is inspired by the fact that the training cost of the reduced-order ML models developed in this work to reconstruct the dominant low-frequency pressure mode corresponds to 30–50 min of compute time on a personal central processing unit (CPU) computer or a modest GPU card, depending on the ML model architecture. At the same time, the cost of LES that is largely driven by convergence of low frequencies is equivalent to continuously running of one GPU card for six days (Appendix B).

For the reconstruction of the leading SPOD pressure time coefficient of the axisymmetric pressure mode at $St = 0.1$ in this work, four state-of-the-art machine-learning strategies were assessed: Feedforward Neural Networks (FNN), Genetic Programming (GP), Echo State Networks (ESN), and Transformer.

Notably, the FNN approach, which had been previously found efficient for cylinder and plunging airfoil flow simulations, was found unsuitable for the present broadband, noisy SPOD signal,

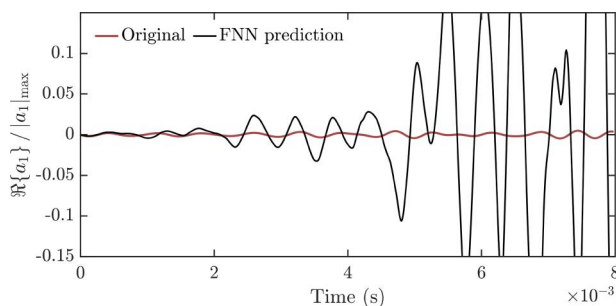


FIG. 21. FNN prediction of the leading SPOD time coefficient for $St = 0.1$ and azimuthal mode $m = 0$. Comparison of the reference signal (—) and the FNN prediction (---).

TABLE III. Convection speed U_{phase}/U_j for each model reconstruction, estimated from the linear slope of the unwrapped phase of the leading SPOD coefficient, similar to Fig. 8.

Model	U_{phase}/U_j
Reference (LES)	0.63
ESN T3000 (Hilbert)	0.63
GP S1 (Hilbert) T2000	0.62
GP S1 (Hilbert) T3000	0.67
ESN T2000 (Hilbert)	0.69
Transformer T2000	0.56
Transformer (Hilbert) T2000	0.56
Transformer T3000	0.56
Transformer (Hilbert) T3000	0.56
GP S15 (Hilbert) T2000	0.71
GP S15 (Hilbert) T3000	0.43
GP S1 T3000	0.17
GP S15 T2000	0.12
GP S1 T2000	0.07
GP S15 T3000	−0.04

exhibiting rapid divergence even within the training interval. In contrast, GP, ESN, and Transformer converged robustly during training and produced stable extrapolations, delivering accurate time-series predictions extending to 100% – 200% beyond the training window.

Performance of the other three stable ML models was further evaluated using complementary metrics designed to separate amplitude fidelity from structural/phase fidelity: (i) training convergence, (ii) RMSE of the reconstructed SPOD coefficient, (iii) spatial correlation of the reconstructed pressure wavepacket field with the LES reference, and (iv) spectral/phase consistency via PSD and unwrapped phase.

In terms of RMSE, GP consistently delivered the lowest median errors with the tightest spread across time, while transformer ranked second (improving with longer training and modestly benefiting from the Hilbert option), and ESN exhibited the largest errors and variability. Correlation analysis revealed additional physics-relevant differences not captured by RMSE alone: ESN (Hilbert) achieved the largest coverage of strongly positively correlated field points for the shorter training window. GP Strategy 1 trained directly on the complex coefficient provided comparably high correlation coverage (43.4% and 41.7%), and Transformer yielded more moderate but consistent coverage ($\approx 36\%$ – 37% for $T = 2000$). For longer training, the Hilbert option improved Transformer correlation coverage (by $\approx 10\%$), whereas introducing the Hilbert formulation in GP Strategy 1 reduced correlation coverage.

Spectral and phase diagnostics of the data-driven solutions further highlighted the tradeoffs between different methods. The transformer model produced the best agreement with the reference PSD over the widest frequency band, indicating strong preservation of broadband spectral content. GP reconstructions captured the low-frequency plateau and initial roll-off well; extending the training window improved mid-band agreement, and the Hilbert option reduced high-frequency leakage. ESN reproduced the

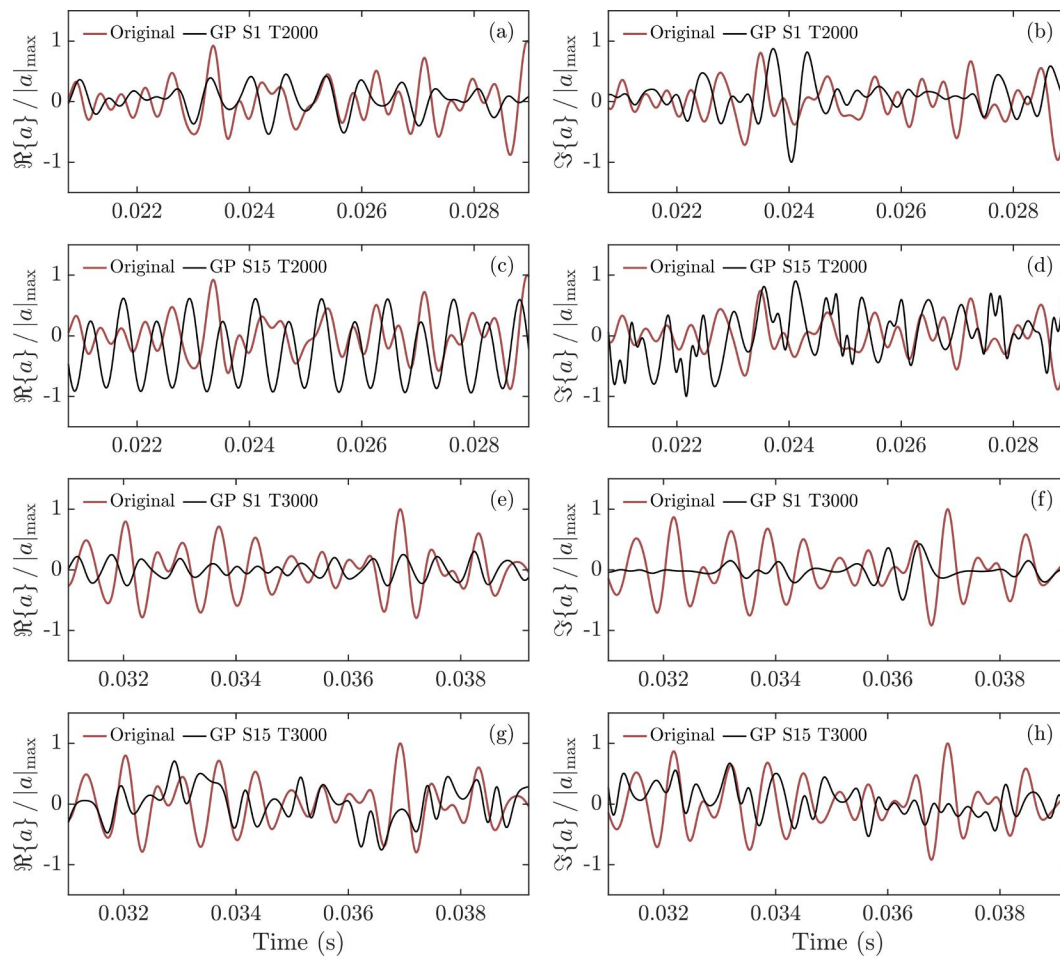


FIG. 22. GP prediction of the leading SPOD time coefficient at $St = 0.1$ for the axisymmetric azimuthal mode $m = 0$. In all panels, the GP prediction (—) is compared with the reference signal (—). The real parts are shown in panels (a), (c), (e), and (g) and the imaginary parts in panels (b), (d), (f), and (h). Results are presented for training window lengths $T = 2000$ (a)–(d) and $T = 3000$ (e)–(h). Two GP strategies are compared: Strategy 1 (a), (b), (e), and (f) and Strategy 15 (c), (d), (g), and (h). The signals show a zoomed view of the prediction region after the training window (first 25% of the prediction).

low-frequency peak region reasonably but struggled to match the faster mid- and high-frequency roll-off of the LES spectrum. For phase, the reference unwrapped phase exhibited an approximately linear trend consistent with travelling-wave dynamics of coherent jet structures. Transformer predictions were largely insensitive to training length and to the Hilbert option and remained in reasonable agreement with the reference phase. By contrast, GP and ESN required the Hilbert formulation for accurate phase recovery: without Hilbert transform, GP exhibited large phase errors; applying Hilbert transform substantially improved both GP and ESN phase predictions, with ESN ($T = 3000$) showing particularly close agreement—albeit with reduced spatial-correlation coverage relative to the $T = 2000$ case, suggesting sensitivity to training length and possible overfitting.

Overall, the use of transformer provides the most consistent performance across all metrics (correlation, PSD, and phase) with comparatively weak sensitivity to training-window length and

without requiring additional analytical post-processing to stabilize the complex signal reconstruction. GP remains highly competitive, offering the best amplitude fidelity (lowest RMSE) and strong correlation when trained directly on the complex coefficient. ESN can achieve excellent phase recovery under selected settings (notably with longer training and Hilbert transform), but its spectral roll-off and correlation behavior indicate higher sensitivity to training and hyperparameters.

Future work will be devoted to multi-frequency/multi modal extensions of the current reduced-order models of spatiotemporal fluctuating pressure, as well as the fluctuating Reynolds stresses that correspond to the source terms of linear jet noise models, such as the Goldstein Generalized Acoustic Analogy,⁴⁹ and which are available from LES in the same way as the pressure signals.⁵⁰ The current analysis will also be extended to higher-order SPOD modes and azimuthal modes, which are important for jet noise at high frequencies.

ACKNOWLEDGMENTS

The authors thank Dr. Oliver T. Schmidt for making his Spectral Proper Orthogonal Decomposition (SPOD) code publicly available.⁴⁷

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Vasily Gryazev: Conceptualization (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Project administration (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Umberto Armani:** Investigation (equal); Methodology (equal); Resources (equal); Software (equal); Writing – review & editing (equal). **Bin Zang:** Formal analysis (equal); Investigation (equal); Methodology (equal); Resources (equal); Software (equal); Writing – review & editing (equal). **Amal Roy Murali:** Formal analysis (equal); Methodology (equal); Resources (equal); Software (equal); Writing – review & editing (equal). **Lidiya Alathiparambil Kennedy:** Investigation (equal); Methodology (equal); Resources (equal). **Zhuofei Wang:** Investigation (equal); Methodology (equal); Software (equal). **Annabel P. Markesteijn:** Data curation (equal); Resources (equal); Software (equal). **Vassili Toropov:** Conceptualization (equal); Project administration (equal); Supervision (equal); Writing – review & editing (equal). **Marc C. Jacob:** Project administration (equal); Resources (equal); Supervision (equal); Writing – review & editing (equal). **Elnaz Naghibi:** Investigation (equal); Methodology (equal); Writing – review &

editing (equal). **Vladimir Ryabov:** Conceptualization (equal); Methodology (equal); Project administration (equal); Writing – review & editing (equal). **Sergey A. Karabasov:** Conceptualization (equal); Formal analysis (equal); Project administration (equal); Resources (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: RECONSTRUCTION RESULTS

Figures 22–24 show FNN, GP, ESN, and transformer predictions of the leading SPOD time coefficient and Fig. 25 shows the distribution of Genetic Programming wall-clock training time.

APPENDIX B: TRAINING AND INFERENCE COST

The training and inference cost of the three machine-learning methods that produced stable solutions in Sec. V (GP, ESN, and Transformer) is summarized below. For the GP, the wall-clock training time is reported for each configuration and is averaged over ten independent random-seed runs to account for the stochastic nature of the evolutionary search. For the ESN, the wall-clock training time is reported for both training-window lengths considered. For the Transformer, hardware-independent quantities (number of trainable parameters and floating-point operations) are reported together with wall-clock measurements.

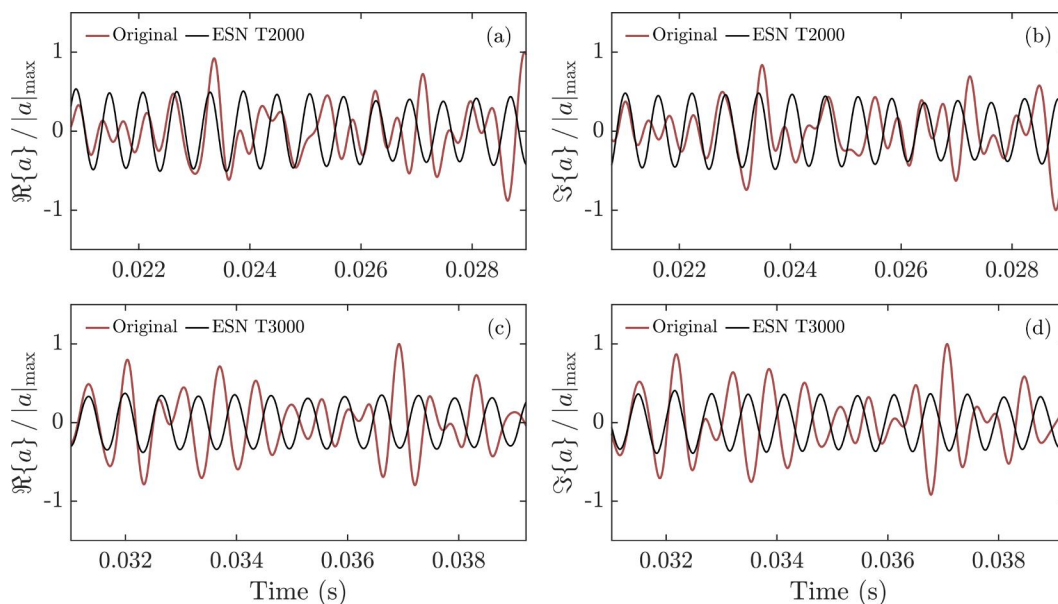


FIG. 23. ESN prediction of the leading SPOD time coefficient at $St = 0.1$ for the axisymmetric azimuthal mode $m = 0$. In all panels, the ESN prediction (—) is compared with the reference signal (—). The real parts are shown in panels (a) and (c) and the imaginary parts in panels (b) and (d). Results are presented for training window lengths $T = 2000$ (a) and (b) and $T = 3000$ (c) and (d). The signals show a zoomed view of the prediction region after the training window (first 25% of the prediction).

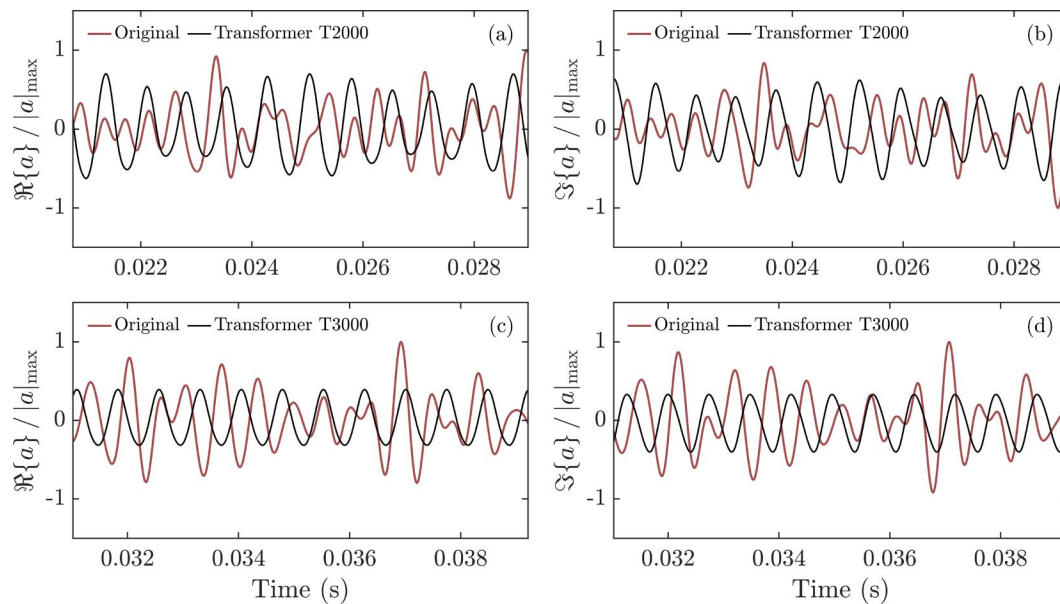


FIG. 24. Transformer prediction of the leading SPOD time coefficient at $St = 0.1$ for the axisymmetric azimuthal mode $m = 0$. In all panels, the transformer prediction (—) is compared with the reference signal (—). The real parts are shown in panels (a) and (c) and the imaginary parts in panels (b) and (d). Results are presented for training window lengths $T = 2000$ (a) and (b) and $T = 3000$ (c) and (d). The signals show a zoomed view of the prediction region after the training window (first 25% of the prediction).

For the Genetic Programming model, all training runs were performed on a 2.5 GHz i7-4710HQ Intel Core laptop CPU used in sequential mode, with a maximum training RAM usage of about 2 GB. The wall-clock training time depends on the strategy (Strategy 1 vs Strategy 15), the training window length ($T = 2000$ or $T = 3000$ samples), and on whether the real or imaginary part of the complex SPOD coefficient is being reconstructed. Each configuration was trained for ten independent random-seed runs. Table IV

reports the mean training time across the ten runs, and the full distribution of runtimes across the ten runs per configuration is shown in Fig. 25. The real and imaginary parts have comparable runtimes for every configuration, but for a given training-window length Strategy 15 is consistently more expensive than Strategy 1. Two factors contribute to this difference: Strategy 15 uses a multi-objective fitness function, whose per-individual evaluation is heavier than the single-objective RMSE used in Strategy 1, and the evolved Strategy 1 expressions tend to be smaller, which further reduces the per-generation cost. Overall, the mean training time

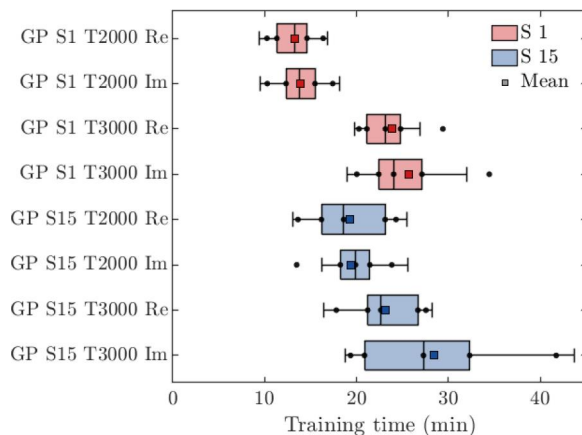


FIG. 25. Distribution of Genetic Programming wall-clock training time over the ten independent random-seed runs for each of the eight configurations considered. Boxes show the 25th–75th percentiles with 10th/90th percentile whiskers, dots mark the 10/25/50/75/90 percentile levels, and colored squares mark the per-configuration mean.

TABLE IV. Mean Genetic Programming wall-clock training time per configuration, averaged over ten independent random-seed runs.

Configuration	Real part (min)	Imag. part (min)
Strategy 1, $T = 2000$	13.2	13.9
Strategy 1, $T = 3000$	23.9	25.6
Strategy 15, $T = 2000$	19.3	19.4
Strategy 15, $T = 3000$	23.2	28.5

TABLE V. Hardware-independent cost figures for the transformer model.

Quantity	Value
Number of trainable parameters	2×10^6
Compute per predictive step	$\approx 4.73 \times 10^9$ FLOPs
Compute for full time series	$\approx 2.4 \times 10^{13}$ FLOPs

TABLE VI. Wall-clock cost figures for the transformer model on a single NVIDIA RTX 4090 GPU.

Quantity	Value
Training (100 epochs)	≈ 28 min
Inference latency per step	2.0 ms
Peak training memory	≈ 8 GB

ranges from approximately 13 min for the fastest configuration (Strategy 1, $T = 2000$, real part) to approximately 28 min for the slowest (Strategy 15, $T = 3000$, imaginary part); reconstructing the full complex coefficient, therefore, requires roughly 27 min for Strategy 1, $T = 2000$, and up to approximately 52 min for Strategy 15, $T = 3000$. As for the deployment costs, as HyGP generates symbolic expressions, the inference costs are only linked to the evaluation of a text expression from the given values of the independent variables.

For the ESN, the trainable weights of the output layer are optimized on a single CPU by iterative mini-batch gradient descent. The maximum number of training epochs is set to 5000, with an early-stopping criterion that halts training when the validation loss has not improved for 1000 consecutive epochs. As shown in Fig. 14, the training loss plateaus broadly between 3000 and 4000 epochs, at which point the early-stopping criterion typically triggers and the run is terminated. The corresponding wall-clock training time is approximately 10 h for $T = 2000$ and 15 h for $T = 3000$.

For the transformer model, the hardware-independent cost quantities (the number of trainable parameters and the floating-point operation budgets) are reported in Table V, and the corresponding wall-clock figures on a single NVIDIA RTX 4090 GPU are reported in Table VI. Training the model for 100 epochs takes approximately 28 min and is a one-off cost. The inference latency of 2.0 ms per predictive step makes the trained model inexpensive to deploy.

For reference, the LES dataset on which the present machine-learning models are trained is produced by the GPU-CABARET solver⁵¹ running on two NVIDIA RTX A6000 (48 GB) GPU cards at a throughput of approximately 260 TUs per 24 h of wall-clock time (equivalently, 130 TUs per day per GPU card), for the SMC000 SP7 jet on the present 110×10^6 -cell mesh. The $N = 5348$ snapshots of the leading SPOD coefficient analyzed in Sec. V correspond to approximately 740 TUs (Table I), so the generation of the full LES record requires of the order of 3 days of wall-clock time on the two GPU cards (equivalently, ~ 6 GPU-days).

REFERENCES

- H. F. S. Lui and W. R. Wolf, "Construction of reduced-order models for fluid flows using deep feedforward neural networks," *J. Fluid Mech.* **872**, 963–994 (2019).
- S. L. Brunton and B. R. Noack, "Closed-loop turbulence control: Progress and challenges," *Appl. Mech. Rev.* **67**, 050801 (2015).
- S. E. Naghibi, S. A. Karabasov, M. A. Jalali, and S. H. Sadati, "Fast spectral solutions of the double-gyre problem in a turbulent flow regime," *Appl. Math. Modell.* **66**, 745–767 (2019).
- S. E. Naghibi, S. A. Karabasov, and I. Kamenkovich, "Zonal jets in the Southern Ocean: A semi-analytical model based on scale separation," *Ocean Modell.* **183**, 102198 (2023).
- J. L. Lumley, "The structure of inhomogeneous turbulent flows," in *Atmospheric Turbulence and Radio Wave Propagation*, edited by A. M. Yaglom and V. I. Tatarski (Nauka, 1967), pp. 166–178.
- M. Sieber, C. O. Paschereit, and K. Oberleithner, "Spectral proper orthogonal decomposition," *J. Fluid Mech.* **792**, 798–828 (2016).
- J. H. M. Ribeiro and W. R. Wolf, "Identification of coherent structures in the flow past a NACA0012 airfoil via proper orthogonal decomposition," *Phys. Fluids* **29**, 085104 (2017).
- K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. M. Dawson, and C.-A. Yeh, "Modal analysis of fluid flows: Applications and outlook," *AIAA J.* **58**, 998–1022 (2020).
- B. Begiashvili, N. Groun, J. Garicano-Mena, S. Le Clainche, and E. Valero, "Data-driven modal decomposition methods as feature detection techniques for flow problems: A critical assessment," *Phys. Fluids* **35**, 041301 (2023).
- A. Towne, O. T. Schmidt, and T. Colonius, "Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis," *J. Fluid Mech.* **847**, 821–867 (2018).
- S. Karami and J. Soria, "Analysis of coherent structures in an under-expanded supersonic impinging jet using spectral proper orthogonal decomposition (SPOD)," *Aerospace* **5**, 73 (2018).
- C. W. Rowley, T. Colonius, and R. M. Murray, "Model reduction for compressible flows using POD and Galerkin projection," *Physica D* **189**, 115–129 (2004).
- S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Natl. Acad. Sci. U. S. A.* **113**, 3932–3937 (2016).
- K. Carlberg, C. Bou-Mosleh, and C. Farhat, "Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations," *Int. J. Numer. Methods Eng.* **86**, 155–181 (2011).
- J. Öst, B. R. Noack, S. Krajnović, D. Barros, and J. Borée, "On the need for a non-linear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body," *J. Fluid Mech.* **747**, 518–544 (2014).
- B. Protas, B. R. Noack, and J. Öst, "Optimal nonlinear eddy viscosity in Galerkin models of turbulent flows," *J. Fluid Mech.* **766**, 337–367 (2015).
- O. San and R. Maulik, "Extreme learning machine for reduced order modeling of turbulent geophysical flows," *Phys. Rev. E* **97**, 042322 (2018).
- Z. Y. Wan, P. Vlachas, P. Koumoutsakos, and T. Sapsis, "Data-assisted reduced-order modeling of extreme events in complex dynamical systems," *PLoS One* **13**, e0197704 (2018).
- S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," *J. Fluid Mech.* **879**, 217–254 (2019).
- S. Laima, X. Zhou, X. Jin, D. Gao, and H. Li, "DeepTRNet: Time-resolved reconstruction of flow around a circular cylinder via spatiotemporal deep neural networks," *Phys. Fluids* **35**, 015118 (2023).
- J. Xu and K. Duraisamy, "Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics," *Comput. Methods Appl. Mech. Eng.* **372**, 113379 (2020).
- M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science* **367**, 1026–1030 (2020).
- J. Lee, S. Shin, T. Kim, B. Park, H. Choi, A. Lee, M. Choi, and S. Lee, "Physics informed neural networks for fluid flow analysis with repetitive parameter initialization," *Sci. Rep.* **15**, 16740 (2025).
- J. Bridges and M. Wernet, "Establishing consensus turbulence statistics for hot subsonic jets," in *16th AIAA/CEAS Aeroacoustics Conference* (American Institute of Aeronautics and Astronautics, 2010).
- V. M. Golovizin and A. A. Samarskii, "Finite difference approximation of convective transport equation with space splitting time derivative," *Mat. Model.* **10**, 86–100 (1998).
- S. Karabasov and V. Golovizin, "Compact accurately boundary-adjusting high-resolution technique for fluid dynamics," *J. Comput. Phys.* **228**, 7426–7451 (2009).
- A. Chintagunta, S. Naghibi, and S. Karabasov, "Flux-corrected dispersion-improved cabaret schemes for linear and nonlinear wave propagation problems," *Comput. Fluids* **169**, 111–128 (2018).
- G. A. Faranosov, V. M. Golovizin, S. A. Karabasov, V. G. Kondakov, V. F. Kopiev, and M. A. Zaitsev, "Cabaret method on unstructured hexahedral grids for jet noise computation," *Comput. Fluids* **88**, 165–179 (2013).

- ²⁹V. Semiletov and S. Karabasov, "Cabaret scheme with conservation-flux asynchronous time-stepping for nonlinear aeroacoustics problems," *J. Comput. Phys.* **253**, 157–165 (2013).
- ³⁰V. Semiletov and S. Karabasov, "Cabaret scheme for computational aero acoustics: Extension to asynchronous time stepping and 3D flow modelling," *Int. J. Aeroacoustics* **13**, 321–336 (2014).
- ³¹V. A. Semiletov, P. G. Yakovlev, S. A. Karabasov, G. A. Faranosov, and V. F. Kopiev, "Jet and jet-wing noise modelling based on the CABARET MILES flow solver and the Ffowcs Williams–Hawkins method," *Int. J. Aeroacoustics* **15**, 631–645 (2016).
- ³²A. P. Markesteijn and S. A. Karabasov, "Simulations of co-axial jet flows on graphics processing units: The flow and noise analysis," *Philos. Trans. R. Soc. A* **377**, 20190083 (2019).
- ³³A. P. Markesteijn, V. Gryazev, S. A. Karabasov, R. S. Ayupov, L. A. Benderskiy, and D. A. Lyubimov, "Flow and noise predictions of coaxial jets," *AIAA J.* **58**, 5280–5293 (2020).
- ³⁴V. Gryazev, A. Kalyan, A. P. Markesteijn, and S. A. Karabasov, "Broadband shock-associated noise modelling for high-area-ratio under-expanded jets," *J. Acoust. Soc. Am.* **150**, 1534–1547 (2021).
- ³⁵S. H. Rudy, J. Nathan Kutz, and S. L. Brunton, "Deep learning of dynamics and signal-noise decomposition with time-stepping constraints," *J. Comput. Phys.* **396**, 483–506 (2019).
- ³⁶D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, "Next generation reservoir computing," *Nat. Commun.* **12**, 5564 (2021).
- ³⁷H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks—with an erratum note," GMD Technical Report (German National Research Center for Information Technology, 2001).
- ³⁸F. Huhn and L. Magri, "Gradient-free optimization of chaotic acoustics with reservoir computing," *Phys. Rev. Fluids* **7**, 014402 (2022).
- ³⁹N. A. K. Doan, W. Polifke, and L. Magri, "Short- and long-term predictions of chaotic flows and extreme events: A physics-constrained reservoir computing approach," *Proc. R. Soc. A* **477**, 20210135 (2021).
- ⁴⁰G. Margazoglou and L. Magri, "Stability analysis of chaotic systems from data," *Nonlinear Dyn.* **111**, 8799–8819 (2023).
- ⁴¹A. Racca and L. Magri, "Data-driven prediction and control of extreme events in a chaotic flow," *Phys. Rev. Fluids* **7**, 104402 (2022).
- ⁴²A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017), Vol. 30.
- ⁴³A. Patil, J. Viquerat, and E. Hachem, "Autoregressive transformers for data-driven spatiotemporal learning of turbulent flows," *APL Mach. Learn.* **1**, 046101 (2023).
- ⁴⁴B. Xu, Y. Zhou, and X. Bian, "Self-supervised learning based on transformer for flow reconstruction and prediction," *Phys. Fluids* **36**, 023607 (2024).
- ⁴⁵J. R. Koza, *Genetic Programming: On the programming of Computers by Means of Natural Selection*, 6th ed. (MIT Press, Cambridge, MA, 1992).
- ⁴⁶U. Armani, Z. Khatir, A. Khan, V. Toropov, A. Polynkin, H. Thompson, N. Kapur, and C. Noakes, "Control of physical consistency in metamodel building by genetic programming," in *Proceedings of the Second International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*, CSC2011 (Civil-Comp Press, 2011), Vol. 97.
- ⁴⁷A. Nekkanti and O. T. Schmidt, "Frequency-time analysis, low-rank reconstruction and denoising of turbulent flows using SPOD," *J. Fluid Mech.* **926**, A26 (2021).
- ⁴⁸C. K. W. Tam, K. Viswanathan, K. K. Ahuja, and J. Panda, "The sources of jet noise: Experimental evidence," *J. Fluid Mech.* **615**, 253–292 (2008).
- ⁴⁹M. E. Goldstein, "A generalized acoustic analogy," *J. Fluid Mech.* **488**, 315–333 (2003).
- ⁵⁰V. Semiletov and S. A. Karabasov, "A volume integral implementation of the Goldstein generalised acoustic analogy for unsteady flow simulations," *J. Fluid Mech.* **853**, 461–487 (2018).
- ⁵¹A. P. Markesteijn and S. A. Karabasov, "Cabaret solutions on graphics processing units for NASA jets: Grid sensitivity and unsteady inflow condition effect," *C. R. Méc.* **346**, 948–963 (2018).
- ⁵²M. Harper-Bourne, "Jet noise turbulence measurements," in *9th AIAA/CEAS Aeroacoustics Conference and Exhibit* (American Institute of Aeronautics and Astronautics, 2003).
- ⁵³J. Bridges and M. P. Wernet, "Measurements of turbulent convection speeds in multistream jets using time-resolved PIV," NASA Technical Memorandum NASA/TM-20205002260 (NASA Glenn Research Center, 2020).
- ⁵⁴S. J. Leib, B. R. Heberling, and J. E. Bridges, "Effect of flight on turbulence statistics and noise sources in a round turbulent jet," AIAA Paper No. 2024-3412, 2024.
- ⁵⁵A. R. Murali, M. L. K. Kennedy, V. Gryazev, U. Armani, A. Markesteijn, E. Naghibi, V. Toropov, M. C. Jacob, R. Hinkelmann, and S. A. Karabasov, "Reduced order modelling of a reynolds number 10^6 jet flow using machine learning approaches," in *30th AIAA/CEAS Aeroacoustics Conference* (American Institute of Aeronautics and Astronautics, 2024).
- ⁵⁶R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming* (Lulu.com, 2008). (With contributions by J. R. Koza).
- ⁵⁷U. Armani, "Development of a hybrid genetic programming technique for computationally expensive optimisation problems," Ph.D. thesis (University of Leeds, 2014).
- ⁵⁸M. Kommenda, B. Burlacu, G. Kronberger, and M. Affenzeller, "Parameter identification for symbolic regression using nonlinear least squares," *Genet. Program. Evolvable Mach.* **21**, 471–501 (2020).
- ⁵⁹M. Märtens and D. Izzo, "Symbolic regression for space applications: Differentiable Cartesian genetic programming powered by multi-objective memetic algorithms," *arXiv:2206.06213* (2022).
- ⁶⁰J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95—International Conference on Neural Networks*, ICNN-95 (IEEE, 1995), Vol. 4, pp. 1942–1948.
- ⁶¹P. Brock, K. Madsen, and H. B. Nielsen, "Robust c subroutines for non-linear optimization," Technical Report NI-91-03 (Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004).
- ⁶²H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science* **304**, 78–80 (2004).
- ⁶³W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.* **14**, 2531–2560 (2002).
- ⁶⁴H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Networks* **20**, 335–352 (2007).
- ⁶⁵H. Jaeger, "A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the 'echo state network' approach," GMD Report 159 (German National Research Center for Information Technology, 2002).
- ⁶⁶M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade*, 2nd ed. (Springer, 2012), pp. 659–686.
- ⁶⁷J. R. Maat, N. Gianniotis, and P. Protopapas, "Efficient optimization of echo state networks for time series datasets," in *International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2018), pp. 1–7.
- ⁶⁸A. Racca and L. Magri, "Robust optimization and validation of echo state networks for learning chaotic dynamics," *Neural Networks* **142**, 252–268 (2021).
- ⁶⁹G. Trierweiler Ribeiro, J. Guilherme Sauer, N. Fraccanabbia, V. Cocco Mariani, and L. dos Santos Coelho, "Bayesian optimized echo state network applied to short-term load forecasting," *Energies* **13**, 2390 (2020).
- ⁷⁰N. Geneva and N. Zabarav, "Transformers for modeling physical systems," *Neural Networks* **146**, 272–289 (2022).
- ⁷¹M. Z. Yousif, M. Zhang, L. Yu, R. Vinuesa, and H. Lim, "A transformer-based synthetic-inflow generator for spatially developing turbulent boundary layers," *J. Fluid Mech.* **957**, A6 (2023).
- ⁷²H. Eivazi, H. Veisi, M. H. Naderi, and V. Esfahanian, "Deep neural networks for nonlinear model order reduction of unsteady flows," *Phys. Fluids* **32**, 105104 (2020).
- ⁷³T. Murata, K. Fukami, and K. Fukagata, "Nonlinear mode decomposition with convolutional neural networks for fluid dynamics," *J. Fluid Mech.* **882**, A13 (2020).
- ⁷⁴I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017.
- ⁷⁵P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," *arXiv:1706.02677* (2017).