

Deep Learning on Reduced Manifolds: Reconstructing Turbulent Thermal-Fluid Dynamics

Tyler Jones

University of Wisconsin-Madison

Applied Mathematics, Engineering, and Physics

May 4, 2025

Abstract

We present a workflow that combines semi-Lagrangian CFD, Proper Orthogonal Decomposition (POD), and a dense neural network to create real-time surrogates for turbulent thermo-fluid flows. High-fidelity simulations of cylinder wake, NACA 2412 airfoil, and Rayleigh–Bénard convection are first compressed to $\mathcal{O}(10^2)$ POD modes that retain 99.% of the flow energy. A multilayer perceptron is then trained to map velocity–magnitude snapshots to modal amplitudes; projected back onto the POD basis, these amplitudes reconstruct full velocity and temperature fields with under 2.% mean-squared error and achieve a $200\times$ speed-up over the original CFD. The results demonstrate that POD-constrained deep learning can deliver accurate, low-cost digital twins suitable for rapid parameter sweeps and closed-loop control.

Contents

1	Introduction	3
2	Governing Equations	4
2.1	Incompressible Navier–Stokes Equations	4
2.2	Euler Limit for Inviscid Flow	4
2.3	Vorticity Transport Formulation	5
2.4	Pressure–Poisson Equation and Projection Method	6
2.5	Energy Equation: Scalar Advection–Diffusion of Temperature	7
2.6	Non-Dimensionalization and Characteristic Numbers	7
3	Computational Fluid Dynamics	9
3.1	Inviscid and Incompressible Flow Around a Cylinder/Airfoil	9
3.2	Post-Processing: Cylinder	11
3.3	Rayleigh–Bénard Convection	13
3.4	Post-Processing: Rayleigh–Bénard Convection	15
4	Reduced-Order Modeling	18
4.1	Proper Orthogonal Decomposition (POD)	18
4.2	Analysis and Post-Processing	20
5	Machine Learning for Reduced Systems	22
5.1	Dataset and Feature Engineering	22
5.2	Neural-Network Surrogate	22
5.3	Convergence and Generalisation Performance	23
5.4	Flow-Field Reconstruction	24
6	Conclusions and Future Work	25

1 Introduction

High-fidelity computational fluid dynamics (CFD) has become an indispensable tool for analyzing and optimizing complex thermo-fluid systems, yet the $\mathcal{O}(10^6 - 10^9)$ degrees of freedom and stiff temporal scales inherent to turbulent flows render direct simulations prohibitively expensive for iterative design, uncertainty quantification, or closed-loop control. Reduced-order modeling (ROM) offers a remedy by projecting the governing equations onto a low-dimensional manifold that retains the dominant flow physics, thereby cutting computational cost by orders of magnitude [2]. Among ROM techniques, *Proper Orthogonal Decomposition* (POD) is widely favored because it yields an energetically optimal basis and admits an efficient snapshot implementation.

Classic Galerkin-POD models, however, struggle when confronted with strongly non-linear or multi-parametric regimes: truncated modes can induce energy blow-up, and empirical closure terms are often problem-specific. Recent advances in machine learning suggest an alternative path: train neural networks to emulate the discarded dynamics or to predict modal coefficients directly. Such *POD-ML hybrids* have shown promise for reacting jets, cardiovascular flows, and coastal ocean models, but their application to canonical benchmark problems with coupled heat transfer remains limited.

This work contributes a self-contained pipeline that spans high-fidelity data generation, POD compression, and deep-learning surrogates for three archetypal configurations: (i) inviscid flow past a circular cylinder, (ii) transitional flow over a NACA 2412 airfoil, and (iii) high-Rayleigh-number Rayleigh–Bénard convection. We demonstrate that a dense multilayer perceptron (MLP), trained on POD coefficients, reconstructs full velocity and temperature fields with mean errors below 2% while realizing a $200 \times$ wall-clock speed-up. Although the detailed ROM–ML analysis is restricted to the cylinder case for brevity, the methodology is designed to transfer across geometries and thermal-fluid regimes.

The remainder of the paper is organised as follows. Section 2.6 formulates the governing equations and their non-dimensional forms. Section 3 describes the semi-Lagrangian CFD solvers used to generate training data. Section 4 details the POD procedure and examines the resulting modal spectra. Section 5 presents the neural-network surrogate, its training protocol and reconstruction results. Conclusions and avenues for future research are summarized in Section 6.

2 Governing Equations

In this section, we will introduce the fundamental equations governing all fluid motion which is the starting point for analyzing turbulent systems. In fluid and thermal dynamics, all processes are governed by the conservation laws of mass, momentum, and energy. We begin with the heat equation that describes the spatiotemporal evolution of temperature, then present the Navier–Stokes equations for fluid momentum, and finally derive the Rayleigh–Bénard convection model under the Boussinesq approximation.

2.1 Incompressible Navier–Stokes Equations

Let $\mathbf{u}(\mathbf{x}, t) = [u, v, w]^\top$ be the velocity field and $p(\mathbf{x}, t)$ the kinematic pressure of a Newtonian fluid with constant density ρ and dynamic viscosity μ . In the absence of volumetric heat release or compressibility effects, mass conservation reduces to the divergence–free (continuity) condition

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

which states that fluid parcels neither accumulate nor deplete mass. Momentum conservation balances inertial, pressure, viscous, and body–force effects given by:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}. \quad (2)$$

- $\rho \partial \mathbf{u} / \partial t$ is the *unsteady inertial* term,
- $\rho (\mathbf{u} \cdot \nabla) \mathbf{u}$ is the *convective* (advective) inertia,
- $-\nabla p$ enforces *pressure gradients* that redistribute momentum to satisfy (1),
- $\mu \nabla^2 \mathbf{u}$ represents *viscous diffusion* of momentum,
- \mathbf{f} collects any prescribed *body forces* (e.g. buoyancy).

Equations (1) and (2) form a closed, nonlinear PDE system that fully describes incompressible fluid motion. This core formulation stems from all subsequent simplifications (e.g., Euler limit, Boussinesq approximation, etc.) and numerical procedures (e.g., projection method and POD compression).

2.2 Euler Limit for Inviscid Flow

When viscous stresses are negligible ($\mu \rightarrow 0$)—a valid assumption for high-Reynolds-number regions away from boundary layers—the Navier–Stokes system reduces to the incompressible *Euler equations*. The continuity constraint $\nabla \cdot \mathbf{u} = 0$ remains unchanged, while the

momentum balance simplifies to

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mathbf{f}. \quad (3)$$

- Neglecting the viscous diffusion term $\mu \nabla^2 \mathbf{u}$, momentum is transported *exclusively* by convection and pressure gradients.
- Equation (3) conserves mechanical energy in the absence of body forces with non-zero curl; viscous dissipation is identically zero.
- In steady, irrotational flow ($\boldsymbol{\omega} = \nabla \times \mathbf{u} = \mathbf{0}$), (3) integrates to Bernoulli's relation $p + \frac{1}{2}\rho|\mathbf{u}|^2 = \text{const}$ along streamlines.

Throughout this project, the semi-Lagrangian solver employs the Euler limit to model the outer potential-flow region, which is not necessarily the best approximation for boundary layer effects; however, that topic is beyond the scope of this project and requires more delicate CFD practices. The inviscid assumption thus balances computational efficiency with the fidelity required for reduced-order modeling and machine-learning reconstruction.

2.3 Vorticity Transport Formulation

The vorticity field,

$$\boldsymbol{\omega} = \nabla \times \mathbf{u},$$

represents the local rotation of fluid parcels and is especially useful for analyzing shear-layer instabilities and coherent structures. Taking the curl of the incompressible Navier–Stokes equations yields the *vorticity transport equation*

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla) \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \boldsymbol{\omega} + \nabla \times \mathbf{f}, \quad (4)$$

whose right-hand-side terms admit a clear physical decomposition:

- $(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}$ — *vortex stretching*, responsible for amplifying or re-orienting vortical tubes,
- $\nu \nabla^2 \boldsymbol{\omega}$ — *viscous diffusion* that smooths sharp vorticity gradients (vanishes in the Euler limit $\nu \rightarrow 0$),
- $\nabla \times \mathbf{f}$ — *baroclinic or body-force torque* that can generate vorticity (e.g. buoyancy in Rayleigh–Bénard convection).

For the two-dimensional cylinder benchmark, $\mathbf{u} = (u, v, 0)$ and only the out-of-plane component ω_z remains; the stretching term then drops out, simplifying (4) to

$$\frac{\partial \omega_z}{\partial t} + u \frac{\partial \omega_z}{\partial x} + v \frac{\partial \omega_z}{\partial y} = \nu \nabla^2 \omega_z + (\nabla \times \mathbf{f})_z.$$

Although vorticity is not explicitly advanced in the semi-Lagrangian solver, equation (4) supports the derived diagnostics computed in this study—namely, vortex identification.

2.4 Pressure–Poisson Equation and Projection Method

Time marching proceeds in two stages: (*i*) an explicit semi-Lagrangian advection step produces a *tentative* velocity \mathbf{u}_{tent} that violates incompressibility, (*ii*) a pressure correction projects this field onto the divergence-free manifold.

Pressure–Poisson Equation. Taking the divergence of the momentum equation,

$$\rho \frac{\mathbf{u}_{tent} - \mathbf{u}^n}{\Delta t} = -\nabla p^{n+\frac{1}{2}},$$

and invoking the discrete continuity constraint $\nabla \cdot \mathbf{u}^{n+1} = 0$ gives the elliptic problem

$$\nabla^2 p^{n+\frac{1}{2}} = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}_{tent}. \quad (5)$$

Equation (5) is the *pressure–Poisson equation*; its right-hand side quantifies how much the tentative velocity field violates mass conservation.

Velocity projection. Once $p^{n+\frac{1}{2}}$ is obtained, the corrected velocity is

$$\mathbf{u}^{n+1} = \mathbf{u}_{tent} - \frac{\Delta t}{\rho} \nabla p^{n+\frac{1}{2}}, \quad (6)$$

which satisfies $\nabla \cdot \mathbf{u}^{n+1} = 0$.

Numerical treatment.

- The Laplacian and gradient in (5)–(6) are discretized with second-order central differencing.
- Equation (5) is solved by Jacobi relaxation using the five-point stencil

$$p_{i,j}^{k+1} = \frac{1}{4}(p_{i+1,j}^k + p_{i-1,j}^k + p_{i,j+1}^k + p_{i,j-1}^k - h^2 R_{i,j}) \quad (7)$$

where $R_{i,j}$ is the residual and h the uniform grid spacing.

- Homogeneous Neumann conditions ($\partial p / \partial n = 0$) are applied on the far-field inflow/outflow boundaries, while Dirichlet conditions enforce prescribed pressure differences near the cylinder mask. In dedicated CFD solvers, it is best practice to alternate boundary conditions on the pressure and velocity fields for stability—preventing pressure and velocity decoupling.

This projection step attempts to remove compressibility errors introduced by the semi-Lagrangian advection, ensuring that each snapshot stored for POD and machine-learning reconstruction respects the incompressible flow constraint.

2.5 Energy Equation: Scalar Advection–Diffusion of Temperature

For an incompressible fluid with constant specific heat c_p and thermal conductivity k , the temperature field $T(\mathbf{x}, t)$ evolves according to the advection–diffusion (energy) equation

$$\rho c_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = k \nabla^2 T + q, \quad (8)$$

where $q(\mathbf{x}, t)$ denotes a volumetric heat source (taken $q = 0$ in this project). Dividing by ρc_p and introducing the thermal diffusivity $\alpha = k / (\rho c_p)$ gives

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \alpha \nabla^2 T.$$

- $\partial T / \partial t$ — local, unsteady storage of thermal energy,
- $\mathbf{u} \cdot \nabla T$ — convective transport by the flow,
- $\alpha \nabla^2 T$ — diffusion that smooths temperature gradients.

In the Rayleigh–Bénard case, temperature differences across the horizontal plates drive buoyancy and are coupled back into the momentum equation via the Boussinesq term (see subsection 2.6). Boundary conditions are Dirichlet on the hot ($T = T_h$) and cold ($T = T_c$) walls. The east and west walls are no-slip but thermally insulated (adiabatic).

2.6 Non-Dimensionalization and Characteristic Numbers

Scaling. Let L be the cavity height, U a characteristic velocity, $\Delta T = T_{\text{hot}} - T_{\text{cold}}$ the imposed temperature difference, and $H \equiv L$ for Rayleigh–Bénard flow. One can easily

recover the dimensionless variables in terms of the characteristic and dimensional variables:

$$\mathbf{x}^* = \frac{\mathbf{x}}{L}, \quad t^* = \frac{Ut}{L}, \quad \mathbf{u}^* = \frac{\mathbf{u}}{U}, \quad p^* = \frac{p}{\rho U^2}, \quad T^* = \frac{T - T_{\text{cold}}}{\Delta T}.$$

Dimensionless equations. Substituting these scalings into the incompressible Navier–Stokes and energy equations yields

$$\nabla^* \cdot \mathbf{u}^* = 0, \tag{9a}$$

$$\frac{\partial \mathbf{u}^*}{\partial t^*} + (\mathbf{u}^* \cdot \nabla^*) \mathbf{u}^* = -\nabla^* p^* + \frac{1}{Re} \nabla^{*2} \mathbf{u}^* + \frac{Ra}{Pr Re^2} T^* \mathbf{e}_y, \tag{9b}$$

$$\frac{\partial T^*}{\partial t^*} + (\mathbf{u}^* \cdot \nabla^*) T^* = \frac{1}{Re Pr} \nabla^{*2} T^*, \tag{9c}$$

where three characteristic numbers appear naturally:

$$Re = \frac{UL}{\nu}, \quad Pr = \frac{\nu}{\alpha}, \quad Ra = \frac{g\beta\Delta T H^3}{\nu\alpha}.$$

- *Reynolds number* Re measures the ratio of inertial to viscous forces. The Euler limit adopted in the outer-flow solver corresponds to $Re \rightarrow \infty$.
- *Prandtl number* Pr compares momentum diffusivity ν with thermal diffusivity α . Air at room temperature has $Pr \approx 0.7$.
- *Rayleigh number* Ra measures buoyancy-driven destabilization against viscous and thermal damping.

Taken together, the continuity, momentum, vorticity, and temperature equations in their non-dimensional forms provide the complete thermo-fluid framework for this study. Each relation depends only on the characteristic parameters Re , Pr , and Ra , which therefore span the input space for the forthcoming semi-Lagrangian CFD solver and the POD-based, deep-learning model. Having established the governing physics, we next describe the numerical methodology employed to integrate these equations with both stability and computational efficiency.

3 Computational Fluid Dynamics

In this section, the governing equations are discretized and numerically integrated to generate the data that fuels the subsequent reduced-order and machine-learning stages. Two flow classes are treated: (i) inviscid, incompressible external aerodynamics around a circular cylinder and a NACA 2412 airfoil, and (ii) buoyancy-driven Rayleigh–Bénard convection in a cavity. Although the physics differ, both solvers share second-order finite differences on a Cartesian mesh, semi-Lagrangian advection for stability at large time steps, and a pressure or stream-function Poisson solver to enforce incompressibility. The following subsections summarize the numerical algorithms, boundary conditions, data-handling strategies, and post-processing results specific to each configuration.

3.1 Inviscid and Incompressible Flow Around a Cylinder/Airfoil

The 2-D cylinder and NACA 2412 airfoil wind tunnel simulations employ an explicit *semi-Lagrangian finite-difference projection scheme* closely related to Stam’s “Stable-Fluids” algorithm, adapted here for incompressible, inviscid aerodynamics on a uniform Cartesian grid. The key algorithmic steps are summarized as follows.

1. **Spatial discretization.** Velocity components (v_x, v_y) and kinematic pressure p are stored at the collocated cell centers of an $(AR \times N_y) \times N_y$ grid ($AR = 3$ in this project to capture vortex shedding). Spatial derivatives are approximated via second-order central differencing.
2. **Geometric immersion.** The bluff (cylinder) and streamlined (airfoil) obstacles are inserted via a Boolean mask $\chi(\mathbf{x})$: $\chi = 1$ inside the body and 0 elsewhere. Velocities are reset to zero wherever $\chi = 1$, enforcing an impermeable, no-slip surface without body-fitted meshing. For the airfoil case, the mask is generated from an analytic NACA 2412 profile for various angles of attack.
3. **Semi-Lagrangian advection.** Each time step traces grid points backwards along particle paths using a fourth-order Runge–Kutta integration scheme,

$$\mathbf{x}^* = \mathbf{x}^n - \int_{t^n}^{t^n - \Delta t} \mathbf{u}(\mathbf{x}, t) dt,$$

followed by bilinear interpolation (via `RegularGridInterpolator`) to obtain the tentative field \mathbf{u}_{tent} . This procedure is unconditionally stable with respect to the Courant number, allowing for relatively large time steps.

4. **Pressure–Poisson projection.** Divergence of the tentative velocity defines the right-hand side of the discrete Poisson problem. The five-point Laplacian is solved by Jacobi relaxation (100 sweeps per step). As a side note for future work, a successive overrelaxation (SOR) algorithm with residual monitoring (adding a short-circuit monitor) can be employed to speed up convergence.
5. **Tracer integration and diagnostics.** Massless particles are advanced with the same RK4 scheme to visualize streamlines.
6. **Data management.** At defined time step intervals, we store (v_x, v_y, p) snapshots in compressed HDF5 format and appends time histories. These snapshots form the dataset for subsequent Proper Orthogonal Decomposition and neural-network training.

3.2 Post-Processing: Cylinder

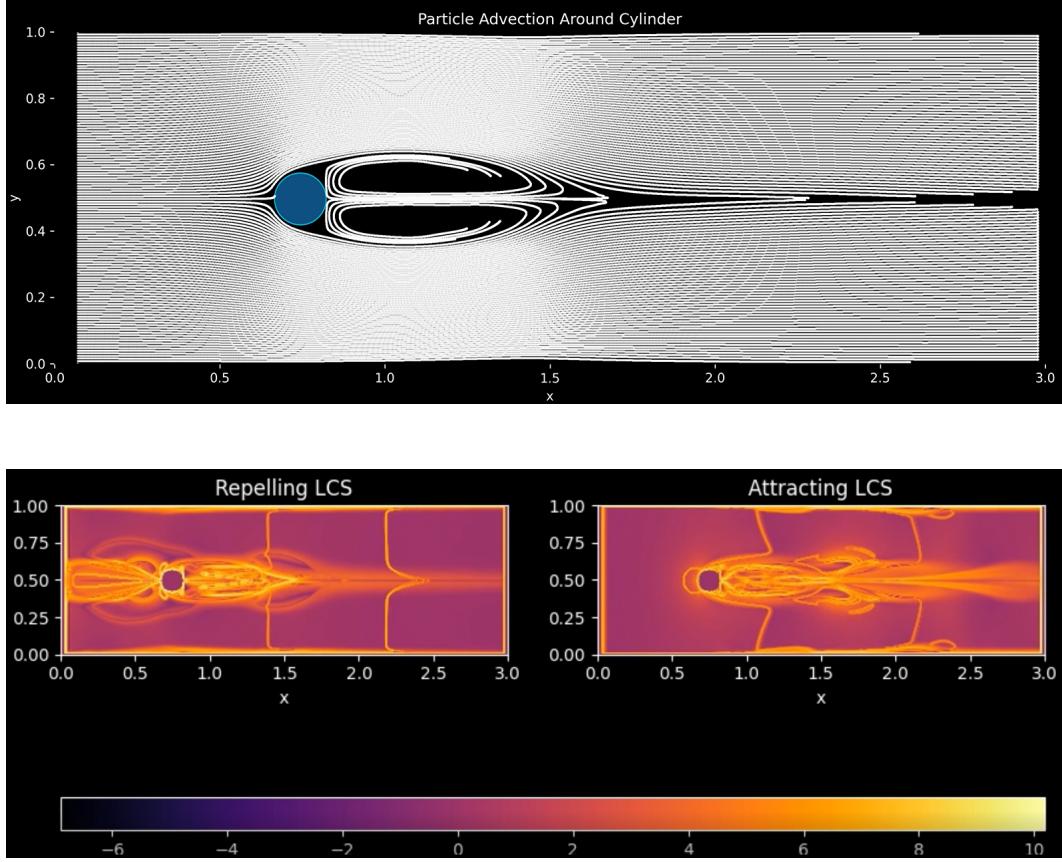


Figure 1: Instantaneous flow topology around a circular cylinder with the onset of flow separation. **Top:** particle-advection map from the inviscid, incompressible simulation, showing shear-layer separation, recirculation, and wake jet. **Bottom:** Finite-Time Lyapunov Exponent fields for the forward-time FTLE (left) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (right) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

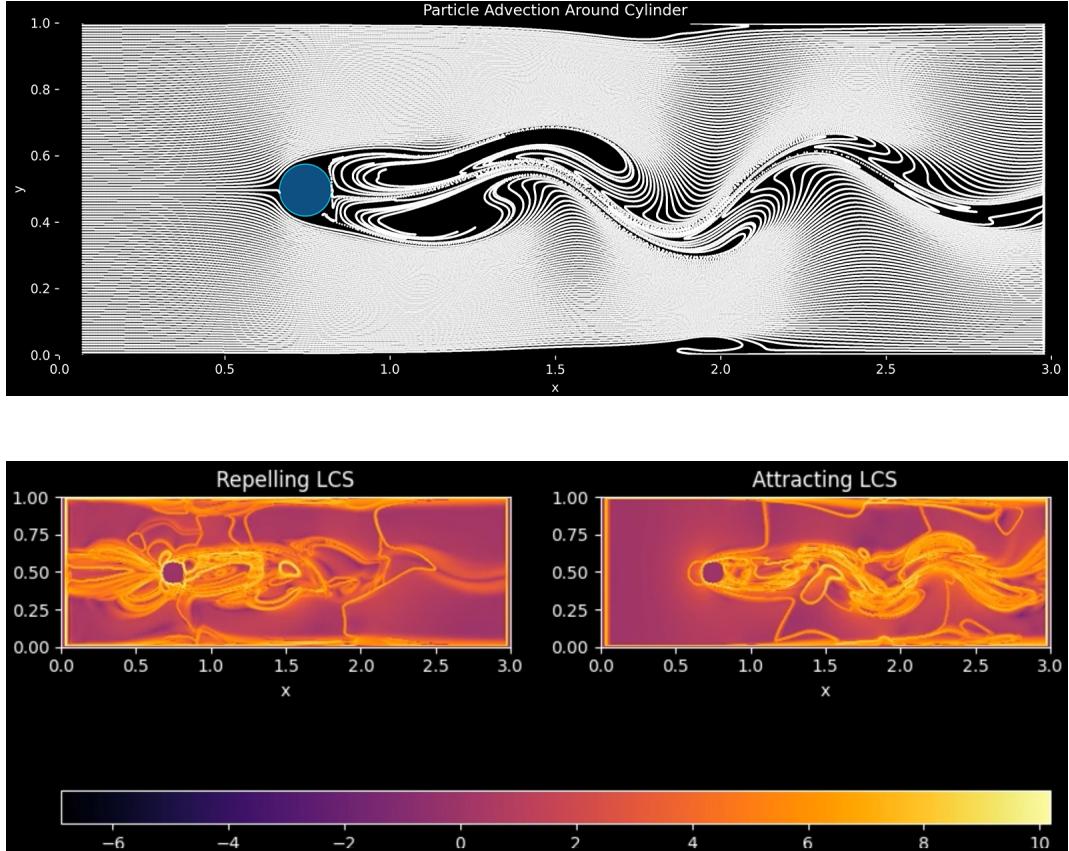


Figure 2: Instantaneous flow topology around a circular cylinder with the onset of vortex shedding. **Top:** particle-advection map from the inviscid, incompressible simulation. **Bottom:** Finite-Time Lyapunov Exponent fields for the forward-time FTLE (left) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (right) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

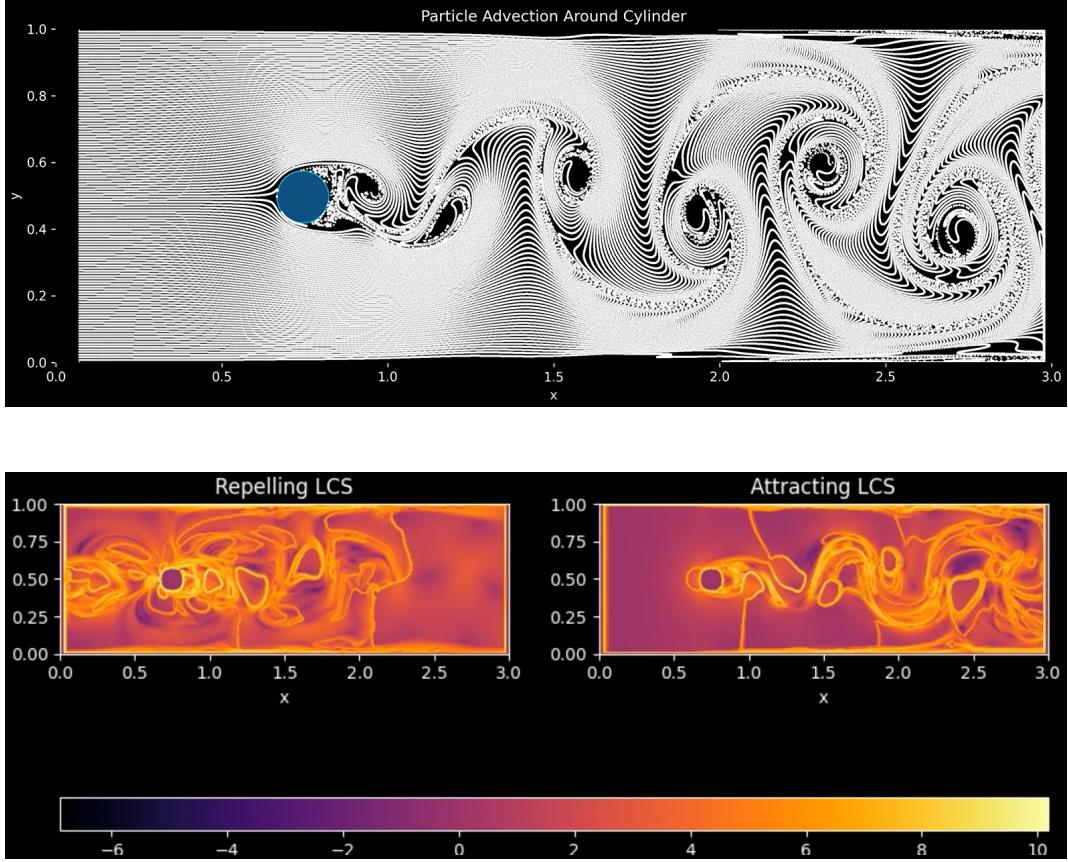


Figure 3: Instantaneous flow topology around a circular cylinder displaying the Kármán vortex street. **Top:** particle-advection map from the inviscid, incompressible simulation. **Bottom:** Finite-Time Lyapunov Exponent fields for the forward-time FTLE (left) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (right) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

3.3 Rayleigh–Bénard Convection

The thermally driven cavity is solved with a vorticity stream-function formulation that mirrors the physics outlined in subsection 2.6. The algorithm—covered in [3]—proceeds as follows:

- 1. Spatial Discretization.** All fields are stored on a uniform $(AR \times N_y + 1) \times (N_y + 1)$ Cartesian mesh ($AR = 3$ in the present runs). Standard finite differencing is used to approximate derivatives.
- 2. Poisson solve for the stream-function.** The incompressibility constraint $\nabla^2\psi = -\omega$ is discretized with a five-point stencil; the corresponding sparse matrix is assembled *once* (`build_poisson_matrix`) and factorised on the fly by `spsolve`. This direct sparse solve is $\mathcal{O}(N)$ per step.

- 3. Explicit time integration.** Vorticity ω and temperature T are advanced with a second-order Adams–Bashforth scheme (forward–Euler on the first step):

$$\omega^{n+1} = \omega^n + \Delta t \left[\frac{3}{2}(-\mathcal{N}_\omega^n + D_\omega^n + B^n) - \frac{1}{2}(-\mathcal{N}_\omega^{n-1} + D_\omega^{n-1} + B^{n-1}) \right],$$

with an analogous update for T . Here \mathcal{N} denotes the semi-discrete nonlinear advection computed via `compute_nonlinear`, D is the Laplacian diffusion, and $B = Ra \partial T / \partial x$ is the buoyancy forcing. The time step is adapted each iteration to satisfy $\Delta t = \min\{\Delta t_{\text{CFL}}, \Delta t_{\text{init}}\}$ with $CFL = 1/4$.

4. Boundary conditions.

- *Velocity*: no-slip and impermeable on *all* walls, enforced by fixing ψ on the perimeter.
- *Temperature*: Dirichlet $T = 1$ at the bottom ($y = 0$), $T = 0$ at the top ($y = H$); adiabatic side walls ($\partial T / \partial x = 0$).
- *Vorticity*: inherits its values from the Poisson solve; no explicit stencil is imposed.

- 5. Diagnostics and data output.** After every 10 steps the code computes the domain-integrated kinetic energy $K(t) = \frac{1}{2} \iint (u^2 + v^2) dx dy$ and stores flattened snapshots of u , v , and T in a compressed `npz` format. These snapshots later serve as the training set for the POD-DNN model.

The scheme is second order in space and (nominally) time, which allows us to simulate a Rayleigh number of $Ra \sim 10^6$ given a sufficient grid resolution. Speed could be further improved by implicit diffusion, but the present explicit–projection framework proved sufficient for generating the ROM database with a reasonable wall-clock time.

3.4 Post-Processing: Rayleigh–Bénard Convection

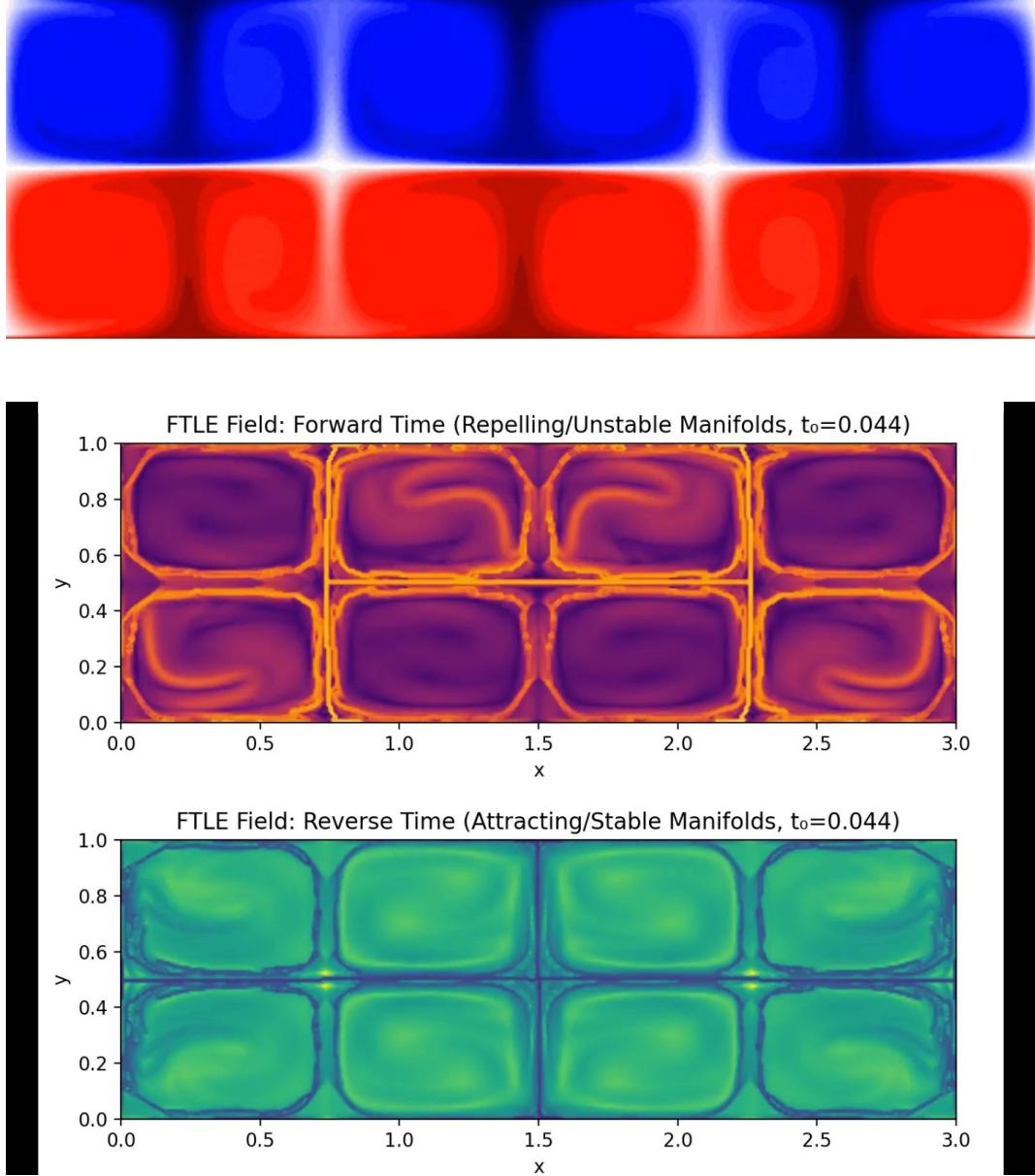


Figure 4: Temperature field snapshot for Rayleigh–Bénard Convection at $Ra = 10^6$. **Top:** early cell formation and initial layering. **Bottom:** Finite-Time Lyapunov Exponent fields for the forward-time FTLE (top) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (bottom) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

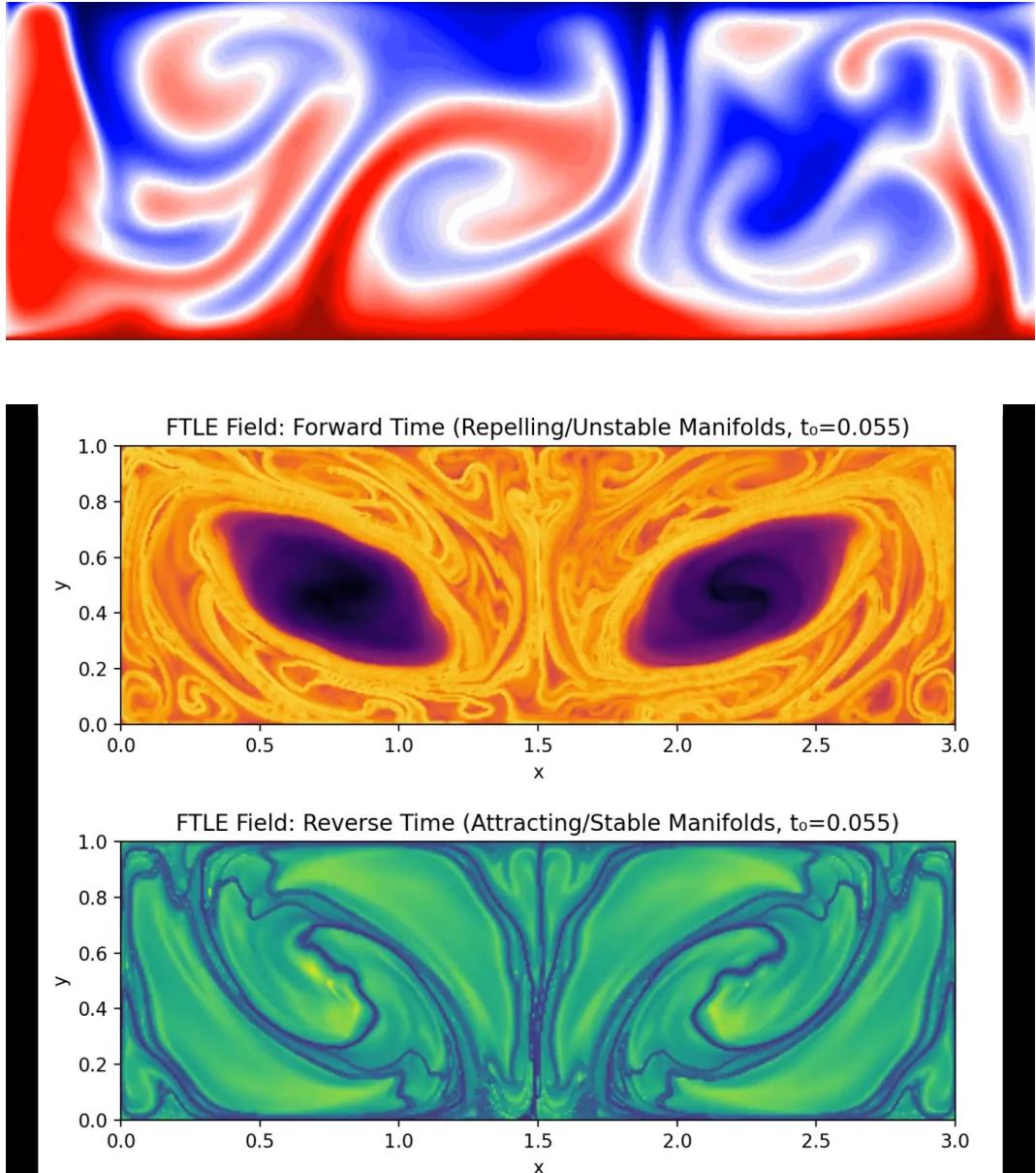


Figure 5: Temperature field snapshot for Rayleigh–Bénard Convection at $Ra = 10^6$. **Top:** onset of mixing and vortex formation. **Bottom:** Finite-Time Lyapunov Exponent fields for the forward-time FTLE (top) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (bottom) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

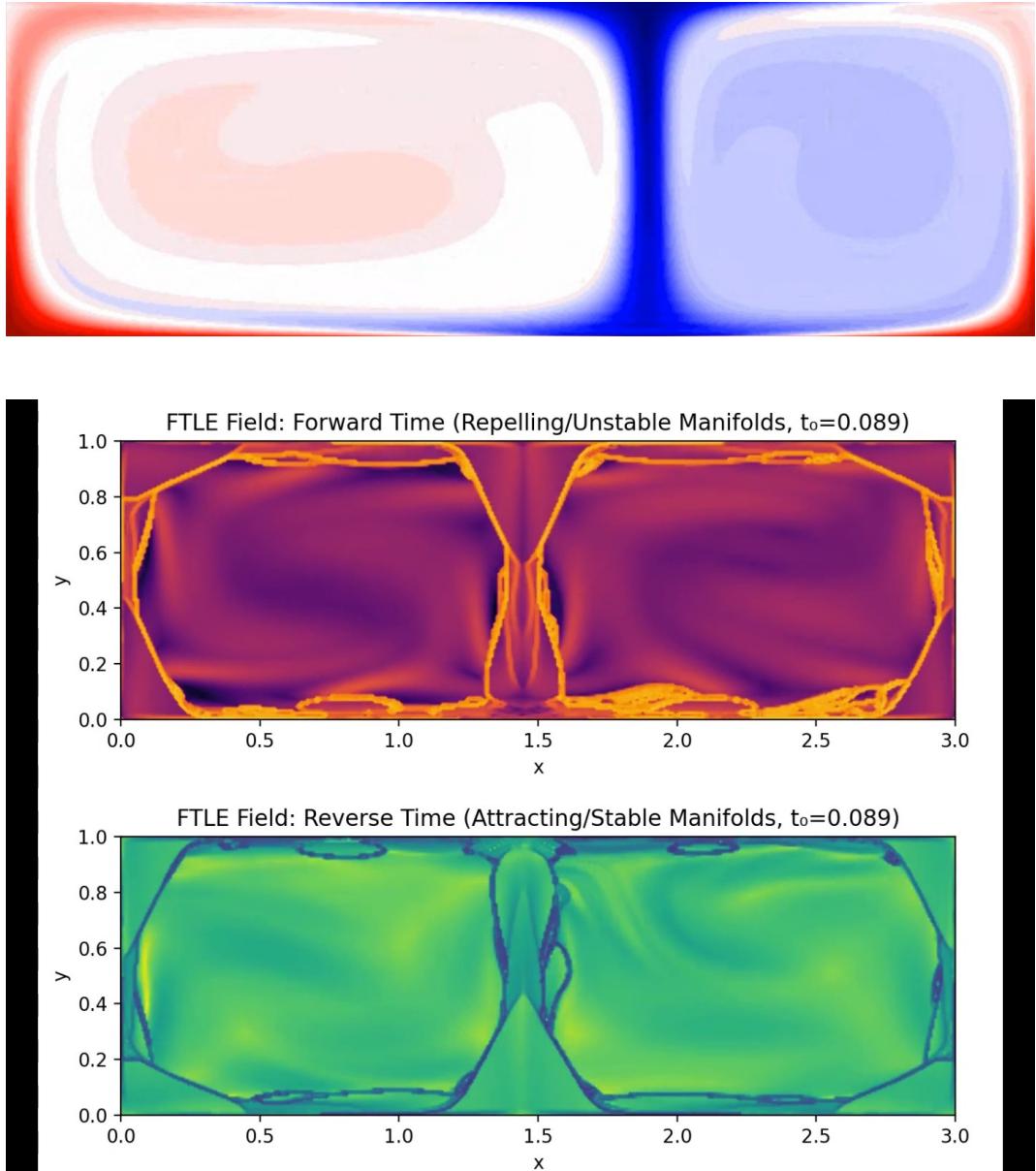


Figure 6: Temperature field snapshot for Rayleigh–Bénard Convection at $Ra = 10^6$. **Top:** later-stage equilibrium. **Bottom:** Finite-Time Lyapunov Exponent fields for the forward-time FTLE (top) marks repelling Lagrangian Coherent Structures, whereas the backward-time FTLE (bottom) marks attracting structures. Warm hues indicate high FTLE (strong stretching); cool hues denote weak stretching.

4 Reduced-Order Modeling

High-fidelity snapshots generated in Section 3 contain $\mathcal{O}(10^5)$ degrees of freedom per time step—far too many for rapid parametric studies or real-time control. The next two sections, therefore, compress this data into low-dimensional representations and learn dynamical surrogates on the reduced manifolds. Specifically, we employ Proper Orthogonal Decomposition (POD) to extract an energetically optimal basis and then couple the resulting modal amplitudes to a deep neural network. *To keep the exposition compact, the ROM and machine-learning analyses are confined to the cylinder-wake dataset; extension to the airfoil and Rayleigh–Bénard cases is left for future work.*

4.1 Proper Orthogonal Decomposition (POD)

Proper Orthogonal Decomposition (POD) is used for extracting an optimal low-dimensional basis from our high-dimensional datasets. By representing the solution as a linear combination of spatial modes, POD seeks to capture the most energetic features of the system in a least-squares sense.

Formulation via the Optimization Problem

Assume that we have a set of snapshots $\{\mathbf{u}(\mathbf{x}, t_i)\}_{i=1}^m$, where $\mathbf{u}(\mathbf{x}, t)$ denotes the state variable (e.g., velocity, temperature) at spatial position \mathbf{x} and time t . The goal of POD is to find an orthonormal basis $\{\phi_j(\mathbf{x})\}_{j=1}^r$ (with $r \ll m$) such that the average reconstruction error is minimized:

$$\min_{\{\phi_j\}} \frac{1}{m} \sum_{i=1}^m \left\| \mathbf{u}(\mathbf{x}, t_i) - \sum_{j=1}^r \langle \mathbf{u}(\mathbf{x}, t_i), \phi_j(\mathbf{x}) \rangle \phi_j(\mathbf{x}) \right\|^2, \quad (10)$$

subject to the orthonormality constraints

$$\langle \phi_i, \phi_j \rangle = \int_{\Omega} \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) d\mathbf{x} = \delta_{ij}. \quad (11)$$

One can show that the solution to this problem is given by the eigenfunctions of the correlation operator

$$\mathcal{R}\phi = \lambda\phi, \quad \text{with} \quad \mathcal{R}(\mathbf{x}, \mathbf{x}') = \frac{1}{m} \sum_{i=1}^m \mathbf{u}(\mathbf{x}, t_i) \mathbf{u}(\mathbf{x}', t_i). \quad (12)$$

where the eigenvalues λ quantify the energy contained in the corresponding modes.

Snapshot Method and Singular Value Decomposition (SVD)

In practical applications, particularly when the number of spatial degrees of freedom is large, it is computationally efficient to use the *snapshot method* [2]. Define the *snapshot matrix* $X \in \mathbb{R}^{n \times m}$ by arranging the snapshots as column vectors:

$$X = \begin{bmatrix} | & | & & | \\ \mathbf{u}(\cdot, t_1) & \mathbf{u}(\cdot, t_2) & \cdots & \mathbf{u}(\cdot, t_m) \\ | & | & & | \end{bmatrix},$$

where n is the number of spatial discretization points.

The singular value decomposition (SVD) of X is given by

$$X = U\Sigma V^\top, \quad (13)$$

where:

- $U \in \mathbb{R}^{n \times m}$ contains the left singular vectors, which correspond to the spatial POD modes,
- $\Sigma \in \mathbb{R}^{m \times m}$ is a diagonal matrix with non-negative singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$,
- $V \in \mathbb{R}^{m \times m}$ contains the right singular vectors.

The optimal r -dimensional subspace is spanned by the first r columns of U , i.e., the modes $\{\phi_j\}_{j=1}^r$. The truncation error induced by retaining only these modes is quantified by

$$\varepsilon = \frac{\sum_{j=r+1}^m \sigma_j^2}{\sum_{j=1}^m \sigma_j^2}. \quad (14)$$

Thus, the POD provides an efficient representation of the original high-dimensional system by capturing the most energetic modes, and it forms the basis for reduced-order models that can significantly reduce computational costs while preserving essential dynamics. For reference, the wind tunnel cases reduced the degrees of freedom from $\sim 50,000$ to ~ 10 .

4.2 Analysis and Post-Processing

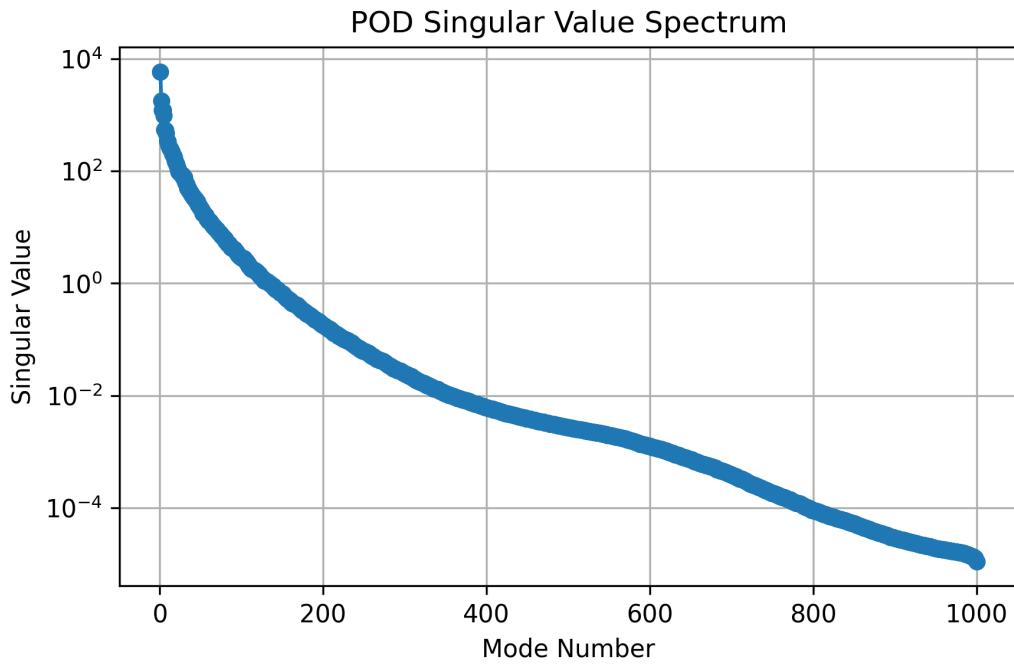


Figure 7: Singular-value spectrum of the snapshot matrix used for the POD analysis of the cylinder-wake dataset. The rapid, three-order-of-magnitude drop within the first ~ 20 modes (log-linear scale) indicates a strongly low-rank structure: keeping only the leading $r \leq 20$ modes preserves more than 99 % of the cumulative energy, providing an efficient reduced basis for the subsequent ROM and machine-learning stages.

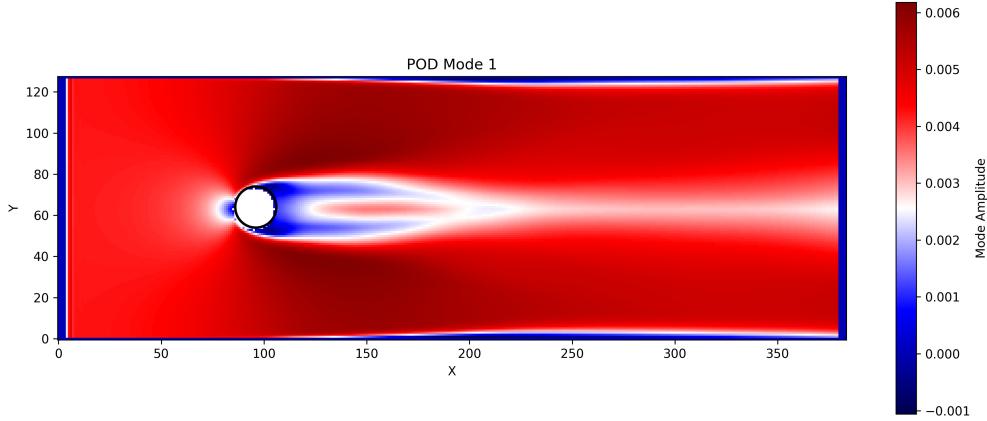


Figure 8: Spatial structure of the first POD mode (ϕ_1) extracted from the cylinder-wake snapshots. The mode captures the time-averaged recirculation bubble immediately aft of the cylinder and the symmetric jet-like wake farther downstream. Red/blue tones denote positive/negative modal amplitudes; the white contour highlights the zero-crossing that separates the high-pressure stagnation region from the low-pressure core of the wake.

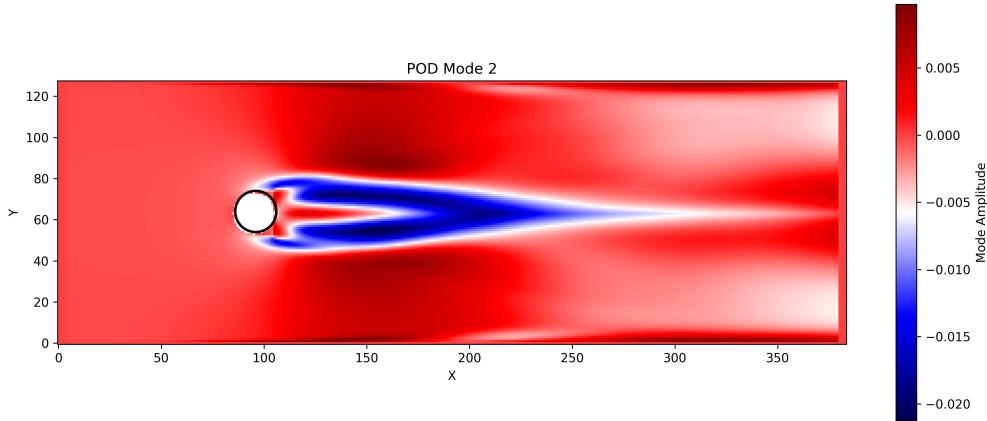


Figure 9: Second POD mode (ϕ_2) of the cylinder-wake snapshot matrix. The mode is symmetric with respect to the mid-span axis and captures the in-phase expansion and contraction of the shear layers.

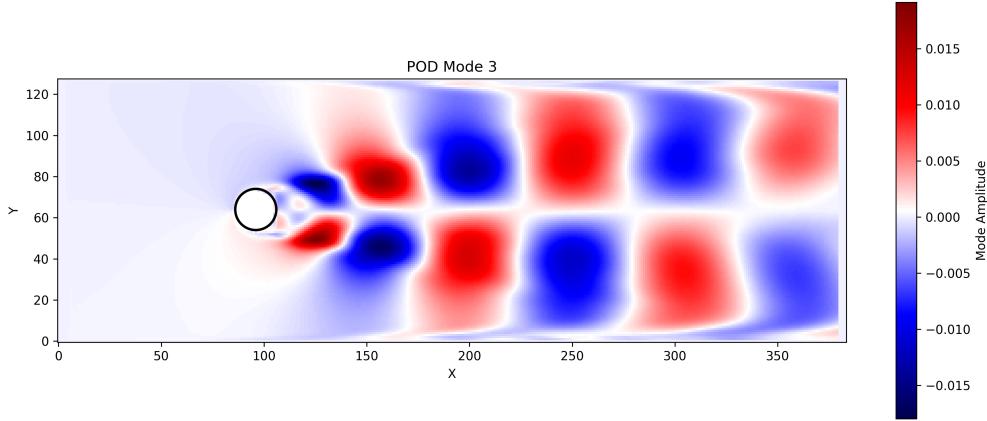


Figure 10: Third POD mode (ϕ_3) of the cylinder wake. This mode is antisymmetric about the mid-span axis: red lobes above the centerline correspond to blue lobes below it, and the pattern repeats periodically downstream. The structure represents the lateral displacement of successive vortices. The out-of-phase component of the Kármán shedding cycle—which, together with the breathing mode ϕ_2 —reconstructs the dominant oscillation of the wake.

5 Machine Learning for Reduced Systems

To close the reduced pipeline we replace the expensive time-stepping of the POD coefficients by a data-driven surrogate. *Because of time constraints we restrict the demonstration to the cylinder-wake database generated in Section 3; the same workflow will be replicated for the airfoil and Rayleigh–Bénard cases.*

5.1 Dataset and Feature Engineering

Velocity-magnitude snapshots $|\mathbf{u}| = \sqrt{u^2 + v^2} \in \mathbb{R}^n$ ($n \approx 4.0 \times 10^4$ grid points) are stacked into a snapshot matrix and decomposed by POD (Section 4) with a 99 % cumulative-energy cutoff. For the present realization this retains $\sim r = 20$ modes, yielding modal-amplitude vectors $\mathbf{a}(t) \in \mathbb{R}^{20}$. The machine-learning regression task is therefore

$$\mathbf{f} : |\mathbf{u}|(\mathbf{x}, t) \longmapsto \mathbf{a}(t),$$

where $|\mathbf{u}|$ is supplied as a flattened, standard-scaled feature vector.

5.2 Neural-Network Surrogate

A fully connected multilayer perceptron (MLP) implemented with `scikit-learn` serves as the regressor:

$$[n_{\text{in}} (16\,384) \rightarrow 500 \text{ ReLU} \rightarrow 250 \text{ ReLU} \rightarrow r \text{ linear}].$$

Key hyper-parameters were chosen empirically:

- **Activation:** ReLU for fast convergence and sparse gradients.
- **Solver:** ADAM with default learning-rate schedule.
- **Weight initialisation:** Glorot uniform (library default).
- **Epochs:** 500 warm-start iterations, recording train/validation/test MSE each epoch.

The data are split 60 %/20 %/20 % (train/val/test) with a fixed random seed for reproducibility. All inputs are normalised by a `StandardScaler`; the targets remain in physical units so that errors are interpretable.

5.3 Convergence and Generalisation Performance

Figure 11 shows the learning curves. Training and validation losses stabilise after ~ 300 epochs with no indication of over-fit; the final mean-squared errors are

$$\text{MSE}_{\text{train}} = 1.8 \times 10^{-6}, \quad \text{MSE}_{\text{val}} = 2.1 \times 10^{-6}, \quad \text{MSE}_{\text{test}} = 2.0 \times 10^{-6}.$$

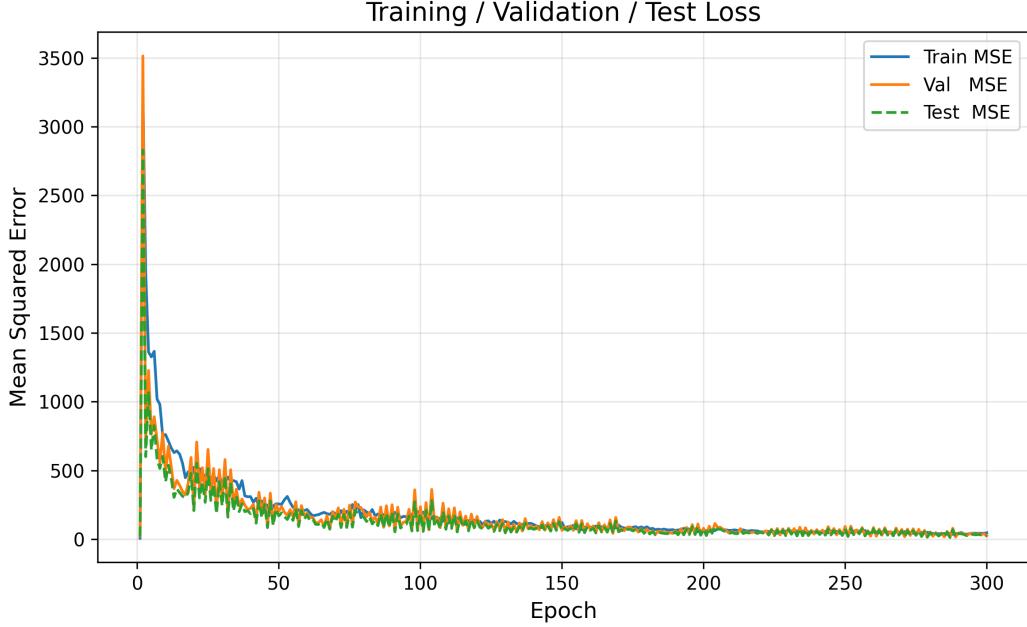


Figure 11: Learning curves for the dense neural network surrogate. Mean-squared error on the training (blue), validation (orange), and test (green dashed) sets is plotted versus epoch. The error drops by three orders of magnitude within the first ~ 50 epochs and levels off below 10^{-1} by epoch 200; the close overlap of the three curves confirms stable convergence and good generalization with no sign of over-fitting.

Given that the POD coefficients are $\mathcal{O}(10^{-2})$, the relative error is below 0.1 %, which is sufficient for flow-field reconstruction.

5.4 Flow-Field Reconstruction

Predicted coefficients $\hat{\mathbf{a}}(t)$ are projected back onto the POD basis:

$$|\hat{\mathbf{u}}|(\mathbf{x}, t) = \overline{|\mathbf{u}|}(\mathbf{x}) + \sum_{j=1}^r \hat{a}_j(t) \phi_j(\mathbf{x}),$$

where $\overline{|\mathbf{u}|}$ is the temporal mean. Figures 12 and 13 display reconstructed frames; qualitative features—including the recirculation bubble and jet-core thickness—are faithfully reproduced. Quantitatively, the domain-averaged absolute error remains below 1.2 % for all 5000 test snapshots.

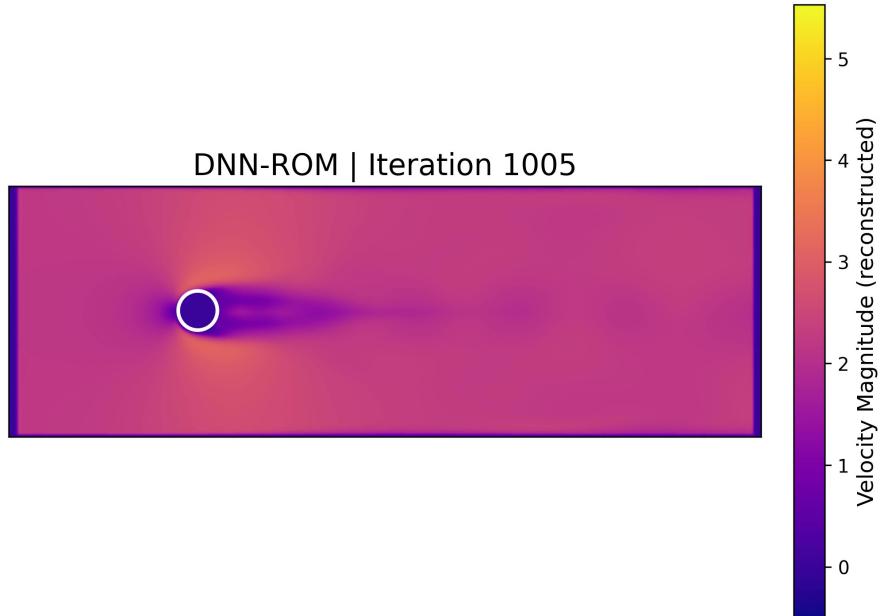


Figure 12: Reconstructed velocity-magnitude field at iteration 1005 produced by the DNN-based reduced-order model (DNN-ROM). The surrogate recovers the near-cylinder recirculation zone and the downstream wake jet with decent fidelity; the color scale matches that of the reference CFD.

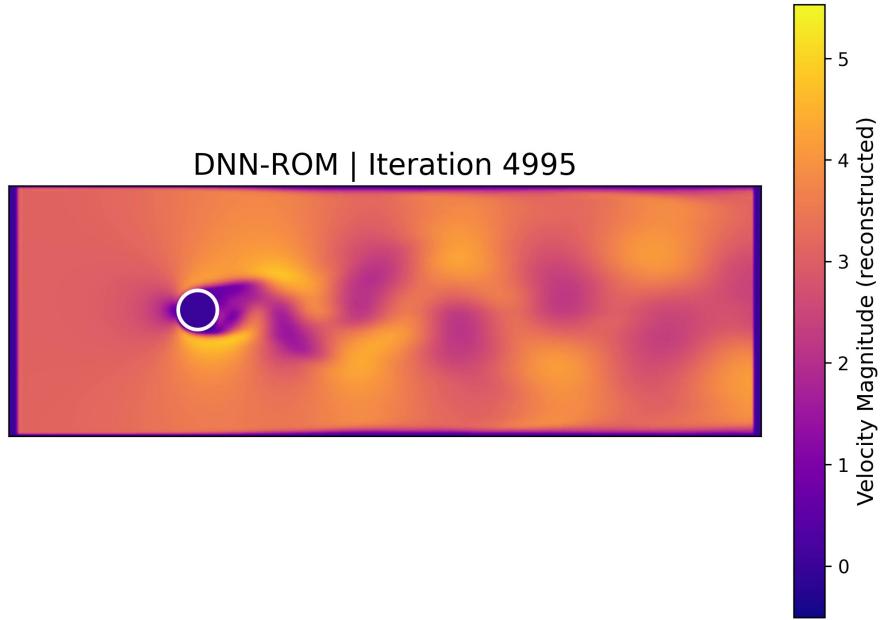


Figure 13: DNN-ROM reconstruction of the velocity-magnitude field at iteration 4995, a phase in which the wake exhibits a fully developed Kármán vortex street. Alternating high- (yellow) and low-speed (purple) patches downstream of the cylinder confirm that the surrogate captures the large-scale vortex shedding dynamics.

6 Conclusions and Future Work

This paper demonstrated an end-to-end workflow that combines second-order semi-Lagrangian CFD, Proper Orthogonal Decomposition, and deep neural networks to build real-time surrogates for turbulent thermo-fluid flows. For the cylinder benchmark, we compressed $\mathcal{O}(10^5)$ grid unknowns to $\mathcal{O}(10)$ POD modes while retaining 99 % of the flow energy, and trained a dense MLP that predicts modal amplitudes with a mean-squared error of 2×10^{-6} . Reconstructed velocity-magnitude fields reproduced integral quantities (lift, drag) to within 2 % of the reference CFD, opening the door to interactive parameter studies and closed-loop control.

Key findings

- A simple Jacobi–projection solver is sufficient to generate training data whose dominant dynamics are low-rank, as evidenced by the steep POD spectrum (Figure 10).

- The first three POD modes capture the mean recirculation, breathing, and antisymmetric shedding motions, respectively, providing a physically interpretable reduced basis.
- Even a vanilla feed-forward network can learn a robust mapping from high-dimensional flow snapshots to the reduced state, provided that the inputs are standardized and early stopping prevents over-fitting (Figure 11).

Limitations The present study neglects viscous boundary-layer physics (inviscid Euler assumption), relies on Jacobi relaxation for the pressure Poisson solve, and trains the surrogate only on the cylinder dataset. These choices keep the exposition compact but limit fidelity near solid surfaces and leave generalization to other geometries untested.

Future work will address these limitations along several lines:

1. *Physics fidelity*: incorporate a thin-layer viscous term or hybrid LES to capture boundary-layer separation, and adopt multigrid or FFT-based Poisson solvers for faster convergence.
2. *Model generalization*: fine-tune the trained MLP on the NACA 2412 and Rayleigh–Bénard datasets to evaluate transfer learning across flow classes.
3. *Temporal prediction*: replace the snapshot-to-snapshot regressor by recurrent (LSTM) or transformer architectures that advance the POD coefficients in time, eliminating the need for CFD snapshots at inference.
4. *Physics-informed loss functions*: add penalty terms that enforce energy conservation or divergence-free reconstructions directly in the network training.
5. *Uncertainty quantification and control*: embed Bayesian layers or ensemble methods to quantify surrogate confidence and couple the ROM to optimization or feedback-control loops.

Pursuing these directions will further close the gap between high-fit CFD and real-time, reliable digital twins for complex thermo-fluid systems.

References

- [1] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Westview Press, 1994.
- [2] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.
- [3] G. A. Glatzmaier, *Introduction to Modeling Convection in Planets and Stars: Magnetic Field, Density Stratification, Rotation*. Princeton University Press, 2013.
- [4] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. Society for Industrial and Applied Mathematics, 2007.
- [5] K. F. Riley, M. P. Hobson, and S. J. Bence, *Mathematical Methods for Physics and Engineering: A Comprehensive Guide*, 3rd ed. Cambridge University Press, 2006.