

---

# PDE-FREE MASS-CONSTRAINED LEARNING OF COMPLEX SYSTEMS WITH HIDDEN STATES

---

**Gianmaria Viola<sup>1</sup>, Alessandro Della Pia<sup>2</sup>, Ioannis Kevrekidis<sup>3</sup>, Lucia Russo<sup>1,\*</sup>, Constantinos Siettos<sup>4,\*</sup>**

<sup>(1)</sup>Institute of Science and Technology for Energy and Sustainable Mobility (STEMS),  
Consiglio Nazionale delle Ricerche (CNR), Naples 80125, Italy

<sup>(2)</sup>Modeling and Engineering Risk and Complexity,  
Scuola Superiore Meridionale, School for Advanced Studies, Naples 80138, Italy

<sup>(3)</sup>Department of Chemical and Biomolecular Engineering, Department of Applied Mathematics and  
Statistics & Department of Urology, Johns Hopkins University, Baltimore, MD 21218, USA

<sup>(4)</sup>Dipartimento di Matematica e Applicazioni “Renato Caccioppoli”,  
Università degli Studi di Napoli “Federico II”, Naples 80126, Italy

February 10, 2026

## ABSTRACT

We propose a three-tier machine learning (ML) framework based on the so-called *next-generation Equation-free* algorithms for learning the spatio-temporal dynamics of mass-constrained complex systems with hidden states, whose dynamics can in principle be described by partial differential equations (PDEs), but lack explicit models. In the first step, we employ Diffusion Maps (DMs), a nonlinear manifold learning algorithm, to extract low-dimensional latent representations of the complex spatio-temporal evolution. In the second step, we learn manifold-informed reduced-order models (ROMs) with Sparse Identification of Nonlinear Dynamics (SINDy) and standard linear Multivariate Autoregressive models (MVARs) to approximate the solution operator on the latent space. In the final step, the latent dynamics are lifted back to the original high-dimensional space by solving an (ill-posed) pre-image problem via a convex interpolation based on the  $k$ -NN algorithm. In doing so, the proposed framework reconstructs the solution operator of the unknown mass-constrained PDE, without explicitly identifying the PDE itself. For comparison purposes, we also evaluated the performance of the scheme for constructing ROMs based on Proper Orthogonal Decomposition (POD) and prove that both POD and the  $k$ -NN lifting operators preserve the mass. We illustrate the approach using two benchmark problems: (a) the Hughes model of crowd dynamics, which minimizes walking time while avoiding obstacles and high-density regions, and (b) a CFD problem involving the spatio-temporal evolution of a passive tracer advected by a periodic Navier–Stokes velocity field. We show that ROMs informed by DMs yield parsimonious models that consistently outperform the best POD-informed ROMs, yielding stable and accurate approximations of the solution operator in the latent space and, via reconstruction, in the original high-dimensional space over long time horizons.

## 1 Introduction

Many real-world systems exhibit complex spatio-temporal dynamics that, in principle, can be described by partial differential equations (PDEs) but also depend on hidden variables [1–4] that are not directly observable. These hidden variables often represent unknown closures, as in the case of computational fluid dynamics (CFD) [5–10], or internal states, as in biological systems [11–14], or optimal transport strategies—such as those governing crowd dynamics [15–20]. For modeling such systems, one may consider a general data-driven setting in which the specific functional forms or entire operators of the macroscopic PDEs governing the emergent crowd dynamics are not known explicitly. The goal in this case is to use data, either from high-fidelity microscopic simulations or from real-world measurements, to

---

\*Corresponding authors, emails: lucia.russo@stems.cnr.it, constantinos.siettos@unina.it

approximate the right-hand sides or closures of the effective macroscopic equations within a finite-dimensional space. This can be achieved through black-box or gray-box surrogate models, typically implemented as (deep) neural networks (DNNs) [13, 21–24]. Alternative machine learning schemes, such as convolutional neural networks (CNNs) [25, 26] and Gaussian processes [27, 28], have also been explored in similar contexts.

However, approximating the right-hand side of PDEs in finite spaces using DNNs poses various challenges and limitations. For example, the accurate approximation of spatial and temporal derivatives requires access to high-resolution samples, and additional pre-processing is needed in the case of noisy data [13, 21]. Moreover, in real-world scenarios, data are typically available as discrete-time snapshots, limiting the accuracy of temporal derivative approximations. This can introduce significant modeling biases and numerical instabilities. Another approach is to approximate the integral evolution (solution) operator of PDEs, as defined by the corresponding Cauchy problem, using Neural Operators (NOs) [29–34]. NOs approximate continuous evolution operators of PDEs in infinite-dimensional spaces, theoretically enabling mesh-independent solutions and broad generalization. However, the NO paradigm requires learning a highly nonlinear mapping between function spaces, which can demand large training datasets and may suffer from the curse of dimensionality.

To circumvent the “curse of dimensionality” inherent in the training of surrogate DNN- or NO-based PDE models, learning Reduced Order Models (ROMs) [35, 36] in well-constructed latent spaces is crucial to significantly lowering computational cost while capturing the essential dynamics using surrogate DNNs, Gaussian Processes, or NOs [3, 23, 37–39]. These ROMs must also inherently preserve fundamental principles, such as stability for long-run simulations and “physical” accuracy when their dynamics are lifted back to the reconstructed space.

Within this context, we propose a fully data-driven “*PDE-free*” machine learning framework based on nonlinear manifold learning and in particular Diffusion Maps [40, 41], inspired by our previous work on the *Equation-Free multiscale framework* [42] and “*Next-Generation Equation-Free*” algorithms [3, 4, 9, 10, 37, 43–46]. The proposed framework aims to learn the solution operator of the unavailable PDE for high-dimensional systems involving hidden or unobservable dynamics, without explicitly constructing a surrogate PDE per se, but by reconstructing the spatio-temporal dynamics based on ROMs in latent spaces learned with SINDy and with vanilla linear MVARs. We illustrate the methodology via two benchmark problems: (i) the Hughes PDE for crowd dynamics, (ii) the spatio-temporal evolution of a passive scalar under advection by a Navier–Stokes flow. The objective is to learn appropriate *parsimonious* low-dimensional *manifold-informed* ROMs that can implicitly represent hidden dynamics in latent spaces, possibly by augmenting observables with delayed coordinates thus exploiting Takens’ and Whitney’s embedding theorems [47]. Spatio-temporal dynamics, usually modeled by PDEs, evolve on infinite-dimensional state spaces, but their dynamics often collapse to finite-dimensional attractors. Therefore, by applying time-delay embeddings to latent spaces, Whitney’s/Takens’ theorem establishes a diffeomorphic equivalence between the delay-coordinate latent model and the original system dynamics (see results and discussion in [3, 37, 47–55]). In fact, while Takens’ theorem guarantees that a single generic scalar observable with sufficiently many delays can reconstruct the emergent dynamics, Whitney-type embedding theorems (see [47]) indicate that a sufficiently expressive multivariate representation can result in fewer delays (even only a single delay per variable). This idea for constructing such surrogate models can be traced back to the 1990s [48–51, 56–58], where POD was used to construct ROMs from spatio-temporal simulations of PDEs, while for large-scale complex systems, it was encoded in the *Equation-Free multiscale framework* [42], bypassing the need to construct models in explicit form to enable numerical analysis tasks such as bifurcation and stability analysis for macroscopic (emergent) dynamics. In particular, to our knowledge, the first time that nonlinear ODEs were learned using NNs in latent spaces (discovered using autoencoders) based on time delays was in [48]—and it was performed on experimental electrodisolution data; while, the first time that this was done for distributed systems (PDEs) in latent spaces, using POD and time delays of POD coordinates, was in [49], in which the data were experimental spatiotemporal movies of PEEM of CO oxidation on Pt. The theoretical link of the spatiotemporal work with the theory of Approximate Inertial Manifolds [45, 57, 59, 60], which underpins the existence of low-dimensional models for infinite-dimensional spatiotemporal systems, was used in [50] and especially in [51], where NNs were used to identify the nonlinear Galerkin equations on the PDE inertial manifold. In recent years, this approach has been extended through the application of other manifold learning techniques such as Laplacian Eigenmaps [61, 62], Diffusion Maps (DMs) [1, 40, 41, 44, 63–67], ISOMAP [68, 69], and autoencoders (also introduced in the 1990s [70]), combined with techniques for solving the pre-image problem [37, 38, 44, 71, 72], and with surrogate machine learning models [23, 26, 37, 45, 54, 73–77], including multivariate autoregressive models (MVARs) and Gaussian Processes [37], SINDy [75, 78, 79], Koopman Operator [54], Neural Operators [39], and Operator inference for handling non-polynomial nonlinearities [80, 81].

Our method proceeds in three main steps. In the first step, we identify the intrinsic latent coordinates using nonlinear manifold learning (DMs). For comparison, POD, widely used in the CFD community, is also employed to highlight the advantages of DMs. In the second step, we learn the latent dynamics via surrogate models (here for our illustrations via SINDy but also with vanilla linear MVARs with delayed coordinates.) In the third step, we reconstruct the spatio-temporal dynamics by lifting the learned latent dynamics back into the high-dimensional physical space. While

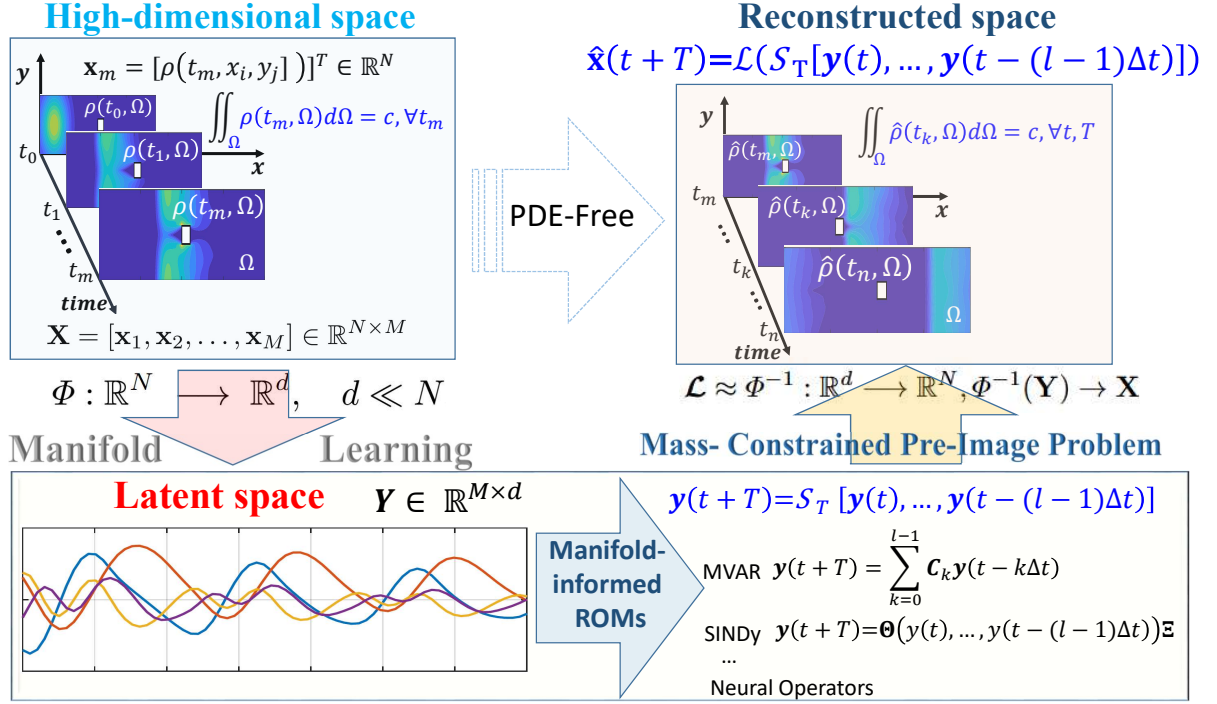


Figure 1: Schematic of the PDE-free mass-preserving approach. (1) Manifold learning: projection of high-dimensional observations onto latent spaces; (2) Learning manifold-informed ROMs (i.e., learning the solution operator in the latent space); (3) Solution of the pre-image problem: construction of the map between the learned latent-space dynamics and the spatio-temporal dynamics (i.e., reconstruction of the solution operator of the unavailable PDE).

POD lifting reduces to a straightforward linear projection, lifting from DMs (or from any other nonlinear manifold learning algorithm) involves solving an ill-posed nonlinear pre-image problem, which we address here through a  $k$ -nearest neighbors ( $k$ -NN)-based decoder with convex interpolation [67]. First, we prove that both the linear POD and  $k$ -NN lifting operators preserve the mass. We then demonstrate that DMs-informed ROMs yield parsimonious models that consistently outperform the best POD-informed ROMs. Remarkably, SINDy achieves this using only a single time delay, thereby demonstrating that DMs provide a rich representation of the embedded manifold. Overall, the proposed framework provides stable and accurate approximations of the solution operator in the latent space and, through reconstruction, in the original high-dimensional space over long time horizons.

The paper is organized as follows. In Section 2, we present the proposed methodology. We first prove that both the POD lifting operator preserves the mass in the reconstructed space. We then briefly describe the DMs nonlinear manifold learning algorithm and prove that the corresponding  $k$ -NN (nonlinear) decoder preserves both mass and positivity of the reconstruction state. Manifold-informed MVARs and SINDy surrogate models are presented in Sections 2.2.1 and 2.2.2. In Section 3, we describe the two benchmark case studies. Numerical results are presented in Section 4, and conclusions are drawn in Section 5. More details on benchmark models, and numerical simulations are provided in Appendices A-D.

## 2 The Methodology

A schematic of the proposed framework is shown in Fig. 1. Assuming a given set of  $M$  spatio-temporal snapshots of a field, say  $\{\mathbf{x}_m\}_{m=1, \dots, M}$ , the aim is to learn a solution operator on a low-dimensional latent space using manifold-

informed ROMs, and then to reconstruct—in a fully data-driven, *Equation-free* manner—an approximation of the apparent solution operator in the high-dimensional input space.

Each snapshot at time  $t_m$  may contain, for example, the density across the spatial domain; i.e.,  $\mathbf{x}_m = [\rho(t_m, x_i, y_j)]^\top \in \mathbb{R}^N$ , where  $i = 1, \dots, N_x, j = 1, \dots, N_y$ , and  $N = N_x \times N_y$  is the number of points in the computational grid. Let us represent the dataset as

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M] \in \mathbb{R}^{N \times M}, \quad (1)$$

with columns  $\mathbf{x}_k \in \mathbb{R}^N$  being the density fields. The key assumption of the methodology is that these data points lie on a smooth  $d$ -dimensional manifold  $\mathcal{M} \subset \mathbb{R}^N$ , and we seek a data-driven mapping

$$\Phi : \mathbb{R}^N \longrightarrow \mathbb{R}^d, \quad d \ll N, \quad (2)$$

such that  $\mathbf{y}_m = \Phi(\mathbf{x}_m) \in \mathbb{R}^d$  captures the dominant coherent structures and emergent dynamics in the latent space, where  $d$  approximates the intrinsic dimension of  $\mathcal{M}$ . The construction of this mapping via manifold learning and in particular Diffusion Maps (POD is also used for comparison purposes) constitutes the first step of the proposed framework.

Once  $\Phi$  is obtained, one can learn surrogate ROMs of the high-dimensional dynamics embedded in the latent space, i.e., learn the solution operator  $\mathcal{S}_T$  of the reduced-order dynamics as:

$$\mathbf{y}(t+T) = \mathcal{S}_T[\mathbf{y}(t), \dots, \mathbf{y}(t - (l-1)\Delta t)], \quad (3)$$

via machine-learning surrogates, where  $T$  is the time horizon of the prediction,  $l$  is the number of delays,  $\Delta t$  is the sampling interval of the original data. Here, we employed SINDy and standard linear MVAR models with delayed coordinates to both demonstrate the framework and provide an indicative comparison of the two approaches.

In the final step, we solve the so-called pre-image problem, i.e., we seek a ‘lifting operator’  $\mathcal{L}$  to reconstruct “on demand” the apparent solution operator  $\mathcal{S}_T^X$  of the high-dimensional spatio-temporal dynamics from the latent dynamics. This allows us to construct the map on the reconstructed space as:

$$\hat{\mathbf{x}}(t+T) = \mathcal{L}(\mathcal{S}_T[\mathbf{y}(t), \dots, \mathbf{y}(t - (l-1)\Delta t)]) := \mathcal{S}_T^X = \mathcal{L} \circ \mathcal{S}_T, \quad (4)$$

thus bypassing the need, as in the Equation-free framework [42], to learn explicitly the high-dimensional map itself. Importantly, we theoretically show that the linear POD lifting operators, as well as the  $k$ -NN algorithm used to construct the lifting operator for DMs, conserve the mass in the input/high-dimensional space. This property is crucial in a wide range of complex systems governed by conservation laws—such as in crowd dynamics and CFD.

Our approach yields compact, computationally efficient surrogates that are non-intrusive (they do not require any explicit knowledge of the governing PDE or parts of it) and are well suited to the Equation-free workflow [42]: fast prediction in latent space followed by data-driven reconstruction to recover high-fidelity solutions.

## 2.1 Manifold Learning and the Pre-image/Reconstruction Problem

In the first step, we determine the map  $\Phi$  from the high-dimensional space to the latent space in Eq. (2) via manifold learning. As discussed, here, for comparison purposes, we consider both POD, and Diffusion Maps (DMs), equipped with a  $k$ -NN decoder. Within the POD context, one assumes a linear map  $\Phi(\mathbf{x}_m) = \mathbf{W}^\top \mathbf{x}_m$ , with  $\mathbf{W} \in \mathbb{R}^{N \times d}$  containing the leading POD modes computed from the covariance matrix of  $\mathbf{X}$  over all  $t_m$ , yielding a projection that maximizes variance in the  $L^2$  sense. Nonlinear manifold learning methods (e.g., kernel-PCA, Laplacian Eigenmaps [61, 62], Diffusion Maps [9, 38, 40, 63, 71, 82], and Autoencoders [26, 83]) generalize this approach by constructing a nonlinear map  $\Phi$  that preserves intrinsic geometric properties of the data. Here, we used both POD and Diffusion Maps.

When using POD, the lifting task is trivial: the full state is reconstructed as a linear combination of POD basis modes weighted by the latent coefficients. In the case of nonlinear manifold learning algorithms, such as Diffusion Maps, we aim to learn an approximation of the inverse map (the lifting operator/decoder):

$$\mathcal{L} \approx \Phi^{-1} : \mathbb{R}^d \longrightarrow \mathbb{R}^N, \quad \Phi^{-1}(\mathbf{Y}) \rightarrow \mathbf{X}, \quad (5)$$

for new samples on the manifold, say  $\mathbf{y}^* \notin \Phi(\mathbf{X})$ , i.e., those produced by the MVAR models. This inverse problem is referred to as the “out-of-sample extension pre-image” problem, which is ill-posed. Compactly, this can be written as

$$\Phi(\mathcal{L}(\mathbf{y}^*)) \approx \mathbf{y}^*. \quad (6)$$

### 2.1.1 POD-based reconstruction conserves mass.

Considering a density field  $\rho(t, x, y)$  as a paradigm, in POD, we decompose the fluctuations of the density field  $\rho(t, x, y)$  with respect to the temporal mean  $\bar{\rho}(x, y)$  as

$$\rho'(t, x, y) = \rho(t, x, y) - \bar{\rho}(x, y) = \sum_{i=1}^{\infty} y_i(t) \varphi_i(x, y), \quad (7)$$

where the spatial modes  $\varphi_i(x, y)$  are mutually orthogonal. The discrete modes  $\varphi_i$  are obtained via the method of snapshots [84], namely by solving the eigenvalue problem for the covariance matrix  $\mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{M \times M}$ :

$$\mathbf{X}^\top \mathbf{X} \psi_i = \lambda_i \psi_i, \quad i = 1, \dots, M, \quad (8)$$

where  $\varphi_i = \mathbf{X} \psi_i / \sqrt{\lambda_i}$  is the  $i$ th mode, and  $\lambda_i, \psi_i$  are the corresponding eigenvalue and eigenvector of the covariance matrix, sorted in descending order such that  $\lambda_1 > \dots > \lambda_M$ . Here,  $\mathbf{X} \in \mathbb{R}^{N \times M}$  is the snapshot matrix. By retaining the leading  $d \ll M$  modes, we construct the reduced-order POD basis parameterizing the manifold, which is subsequently used to project the high-dimensional density field onto the temporal coordinates  $y_i(t)$ , for  $i = 1, \dots, d$ .

Using the POD basis, we define the mapping  $\Phi$  in Eq. (2) as  $\mathbf{y} = \Phi(\mathbf{x}) = \mathbf{W}^\top \mathbf{x}$ , where  $\mathbf{W} = [\varphi_1, \dots, \varphi_d] \in \mathbb{R}^{N \times d}$ , thus defining the latent variables as the leading POD coefficients  $\mathbf{y} = [y_1, \dots, y_d]^\top$ . Note that the POD basis provides a parameterization of the manifold for both seen points  $\mathbf{x}_m$  and unseen ones  $\mathbf{x}_m^*$ . Hence, it can be employed for any point in the high-dimensional space, providing the projection subsequently used for ROM construction via the MVAR model.

We now prove that the reconstructed solution  $\hat{\mathbf{x}}$  via POD preserves the total mass in each snapshot.

**Proposition 1.** *Let the dataset be*

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M] \in \mathbb{R}^{N \times M}, \quad (9)$$

*with columns  $\mathbf{x}_k \in \mathbb{R}^N$  representing normalized to sum to 1 density fields. If the total mass of the original data is preserved, i.e.,*

$$\left[ \sum_{i=1}^N x_{i1} \quad \sum_{i=1}^N x_{i2} \quad \dots \quad \sum_{i=1}^N x_{iM} \right] = \mathbf{1}_M^\top, \quad (10)$$

*then the mass of the reconstructed/decoded field computed by POD as*

$$\hat{\mathbf{X}} = \mathbf{U}_d \mathbf{U}_d^\top \tilde{\mathbf{X}} + \mathbf{X}(\mathbf{I}_M - \mathbf{H}), \quad \mathbf{H} = \mathbf{I}_M - \frac{1}{M} \mathbf{1}_M \mathbf{1}_M^\top, \quad (11)$$

*where  $\mathbf{U}_d = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d] \in \mathbb{R}^{N \times d}$  is the orthonormal basis formed by the first  $d$  left singular vectors of the SVD of the centered matrix  $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{H}$ , is also preserved, i.e.,*

$$\left[ \sum_{i=1}^N \hat{x}_{i1} \quad \sum_{i=1}^N \hat{x}_{i2} \quad \dots \quad \sum_{i=1}^N \hat{x}_{iM} \right] = \mathbf{1}_M^\top. \quad (12)$$

*Proof.* Since the sum of each column of  $\mathbf{X}$  is 1 by hypothesis, we have

$$\mathbf{1}_N^\top \mathbf{X} = \mathbf{1}_M^\top.$$

Then, for the centered matrix  $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{H}$ , we obtain

$$\mathbf{1}_N^\top \tilde{\mathbf{X}} = \mathbf{1}_N^\top \mathbf{X}\mathbf{H} = \mathbf{1}_M^\top \mathbf{H} = \mathbf{1}_M^\top - \frac{1}{M} \mathbf{1}_M^\top \mathbf{1}_M \mathbf{1}_M^\top = \mathbf{1}_M^\top - \mathbf{1}_M^\top = \mathbf{0}_M^\top. \quad (13)$$

This implies that  $\mathbf{1}_N$  is orthogonal to the column space of  $\tilde{\mathbf{X}}$ , and therefore to the left singular vectors of its SVD, which provide an orthogonal basis for that column space. Thus,

$$\mathbf{1}_N^\top \mathbf{U}_d = \mathbf{0}_d^\top, \quad (14)$$

where  $d = 1, 2, \dots, r$ , and  $r$  is the rank of  $\mathbf{X}$ . Multiplying Eq. (11) by  $\mathbf{1}_N^\top$  yields

$$\mathbf{1}_N^\top \hat{\mathbf{X}} = \mathbf{1}_N^\top \mathbf{U}_d \mathbf{U}_d^\top \tilde{\mathbf{X}} + \mathbf{1}_N^\top \mathbf{X}(\mathbf{I}_M - \mathbf{H}) = \mathbf{0}_M^\top + \frac{1}{M} \mathbf{1}_N^\top \mathbf{X} \mathbf{1}_M \mathbf{1}_M^\top = \mathbf{1}_M^\top, \quad (15)$$

since  $\mathbf{1}_M^\top \mathbf{1}_M = M$ . We therefore obtain

$$\mathbf{1}_N^\top \hat{\mathbf{X}} = \left[ \sum_{i=1}^N \hat{x}_{i1} \quad \sum_{i=1}^N \hat{x}_{i2} \quad \dots \quad \sum_{i=1}^N \hat{x}_{iM} \right] = \mathbf{1}_M^\top, \quad (16)$$

hence showing that the total mass in each snapshot is preserved under POD reconstruction.  $\square$

The above result follows from the well-known property of POD to preserve symmetries [85]. At this point, we note that mass conservation and pointwise positivity are distinct properties. While POD-based reconstruction preserves total mass, it does not guarantee non-negativity, and small negative values may appear despite positive input data. Since the mean is added back, such values typically occur in low-density regions. For visualization purposes only, one can set negative entries to zero and renormalize the field by the mass of its nonnegative part, thereby enforcing positivity while preserving total mass. This does not influence the training process, since learning is performed entirely in the latent space and the reconstruction is applied only a posteriori.

### 2.1.2 Encoding with Diffusion Maps and Decoding with the k-NN algorithm

The purpose of DMs is to construct a nonlinear mapping from a high-dimensional space to a low-dimensional subspace while preserving the intrinsic geometry of the underlying manifold. We follow the theoretical formulation and numerical implementation of DMs presented in earlier works [44, 54, 64, 67, 86].

Assume that the data lie on a smooth, low-dimensional manifold  $\mathcal{M} \subset \mathbb{R}^N$ . Diffusion Maps then aim to obtain low-dimensional embeddings  $\mathbf{y} \in \mathbb{R}^d$ , with  $d \ll N$ , collected in the matrix  $\mathbf{Y} \in \mathbb{R}^{M \times d}$ , such that Euclidean distances between points  $\mathbf{y}$  approximate the diffusion distances between the original points [82].

The algorithm begins by defining a similarity measure between pairs of data points  $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}, \forall i, j = 1, \dots, M$ , in the high-dimensional space. Using the Euclidean norm  $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ , we construct a Gaussian kernel  $k(\mathbf{x}_i, \mathbf{x}_j)$ , which defines the affinity matrix:

$$\mathbf{A} = [a_{ij}] = [k(\mathbf{x}_i, \mathbf{x}_j)] = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\epsilon^2}\right), \quad (17)$$

where  $\epsilon$  controls the local neighborhood size in the high-dimensional space. In our implementation, we set  $\epsilon = \text{median}(d_{ij})$ , which promotes a relatively large neighborhood. Other strategies for selecting  $\epsilon$  exist [87, 88].

Next, the  $M \times M$  Markov transition matrix  $\mathbf{M}$  is formed by row-normalizing the affinity matrix:

$$\mathbf{M} = \mathbf{D}^{-1}\mathbf{A}, \quad \text{with } \mathbf{D} = \text{diag}\left(\sum_{j=1}^M a_{ij}\right). \quad (18)$$

Each entry  $\mu_{ij}$  of  $\mathbf{M}$  represents the probability of moving from point  $i$  to point  $j$  in the high-dimensional space:

$$\mu_{ij} = \text{Prob}(X_{t+1} = \mathbf{x}_j \mid X_t = \mathbf{x}_i). \quad (19)$$

Equivalently, using the kernel,

$$\mu_{ij} = \frac{k(\mathbf{x}_i, \mathbf{x}_j)}{\text{deg}(\mathbf{x}_i)}, \quad \text{with } \text{deg}(\mathbf{x}_i) = \sum_{j=1}^M k(\mathbf{x}_i, \mathbf{x}_j), \quad (20)$$

recovering Eq. (18).

The transition matrix  $\mathbf{M}$  is similar to the symmetric, positive-definite matrix  $\hat{\mathbf{M}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ , which allows an eigendecomposition

$$\mathbf{M} = \sum_{i=1}^M \lambda_i \mathbf{w}_i \mathbf{u}_i^\top, \quad (21)$$

where  $\lambda_i \in \mathbb{R}$  are eigenvalues and  $\mathbf{w}_i, \mathbf{u}_i \in \mathbb{R}^M$  are left and right eigenvectors, satisfying  $\langle \mathbf{w}_i, \mathbf{u}_j \rangle = \delta_i^j$ . The right eigenvectors  $\mathbf{u}_i$  span an orthonormal basis for the low-dimensional subspace  $\mathbb{R}^d$ , and the best  $d$ -dimensional approximation is obtained from the  $d$  largest eigenvalues.

The standard DMs embedding maps each snapshot  $\mathbf{x}_m$  to

$$\mathbf{y}_m = (\lambda_1 u_{1,m}, \dots, \lambda_d u_{d,m}), \quad m = 1, \dots, M, \quad (22)$$

where  $u_{i,m}$  denotes the  $m$ -th component of the  $i$ -th right eigenvector corresponding to the  $i$ -th largest non-trivial eigenvalue  $\lambda_i$ . This embedding approximates the diffusion distance in the high-dimensional space by Euclidean distance in the embedded space:

$$D_t^2(\mathbf{x}_i, \mathbf{x}_j) = \|\mu_t(\mathbf{x}_i, \cdot) - \mu_t(\mathbf{x}_j, \cdot)\|_{L_2, 1/\text{deg}}^2 = \sum_{k=1}^M \frac{(\mu_t(\mathbf{x}_i, \mathbf{x}_k) - \mu_t(\mathbf{x}_j, \mathbf{x}_k))^2}{\text{deg}(\mathbf{x}_k)}, \quad (23)$$

with  $\mu_t(\mathbf{x}_i, \cdot)$  the  $i$ -th row of  $\mathbf{M}^t$ . In our computations, we use  $t = 1$ .

In practice, the embedded dimension  $d$  is determined by the spectral gap of the eigenvalue ratio of the transition matrix  $\mathbf{M}$ , assuming that the first  $d$  leading eigenvalues are adequate to provide a good approximation of the diffusion distance between all pairs of points. The resulting DMs embedding is constructed from the retained eigenpairs  $\{\lambda_i, \mathbf{u}_i\}_{i=1}^d$ . The restriction operator (encoder)  $\Phi$ , evaluated on a data point  $\mathbf{x}_m$ , is

$$\Phi(\mathbf{x}_m) = (\lambda_1 u_{1,m}, \dots, \lambda_d u_{d,m}) = \mathbf{y}_m \in \mathbb{R}^d, \quad m = 1, \dots, M. \quad (24)$$

For new, unseen points, we employ the Nyström method [38, 44, 71, 72, 89].

**Decoding with  $k$ -NN** Here, for the  $k$ -NN approach for the solution for the pre-image problem [67], we prove the following proposition.

**Proposition 2.** *Let the assumptions for the data matrix  $\mathbf{X}$  in Proposition 1 hold, i.e., columns  $\mathbf{x}_k \in \mathbb{R}^N$  are normalized to sum to 1. Then the solution of the pre-image problem using the  $k$ -NN algorithm with convex interpolation preserves both the total mass and positivity of the reconstructed state.*

*Proof.* Construct the lifting operator (decoder)  $\mathcal{L}$  using  $k$ -NN as (see in [67]):

$$\mathbf{x}^* = \mathcal{L}(\mathbf{y}^*) = \sum_{k=1}^K b_k \mathbf{x}_{S(k)}, \quad (25)$$

where  $\mathbf{x}_{S(k)} \in \mathbf{X}$  are the nearest neighbors in the high-dimensional space corresponding to  $\mathbf{y}^*$  in latent space. The convex weights  $b_k$  satisfy  $\sum_{k=1}^K b_k = 1$  and  $b_k \in [0, 1]$ .

Since  $\mathbf{x}^*$  is a convex combination of normalized columns of  $\mathbf{X}$ , we have

$$\mathbf{1}^\top \mathbf{x}^* = \sum_{k=1}^K b_k \mathbf{1}^\top \mathbf{x}_{S(k)} = \sum_{k=1}^K b_k = 1. \quad (26)$$

Hence, the reconstructed solution preserves the total mass. Furthermore, a weighted sum of this form—nonnegative weights that sum to one—produces a point  $\mathbf{x}^*$  that lies in the convex hull of its neighbors and is positive if its neighbors are positive.  $\square$

## 2.2 Manifold-Informed ROMs

Manifold-informed ROMs (e.g., using multivariate autoregressive models (MVARs), neural networks, Gaussian Processes, SINDy, or NOs) can be used without changing the proposed framework, as shown in following Section 2.2.2. Here, we employed two approaches: vanilla linear MVARs with multiple delayed coordinates and Sparse Identification of Nonlinear Dynamics (SINDy) [78, 79] with a single delay. For both approaches, we optimized embedding parameters and model hyperparameters (e.g., model order for MVAR, sparsity threshold and library for SINDy) on the training sets, and selected the SINDy candidate library via cross-validation to balance predictive accuracy and model parsimony. More details are given in B and D.

### 2.2.1 Manifold-Informed linear MVARs

As shown in [37], MVARs in delay-coordinate latent spaces [3, 50, 52–55] can outperform nonlinear machine learning surrogates when the latent manifold is well parameterized and the dynamics evolve around a single nonlinear basin of attraction. In this regime, the latent dynamics are approximately linearizable, and MVAR models capture the evolution with high fidelity. Training MVARs requires only linear regression, offering a computationally efficient alternative to deep nonlinear models such as neural networks, LSTMs, or GRUs. Furthermore, the learned MVAR coefficients provide interpretability of the mode interactions, and the framework naturally supports uncertainty quantification.

For one-step-ahead prediction ( $T = \Delta t$ ), an MVAR model of order  $l$  reads:

$$\mathbf{y}(t + \Delta t) = \sum_{k=0}^{l-1} \mathbf{C}_k \mathbf{y}(t - k\Delta t), \quad (27)$$

where  $\mathbf{C}_k \in \mathbb{R}^{d \times d}$  are the autoregressive coefficient matrices at lag  $k$ . In compact form,

$$\mathbf{y}(t + \Delta t) = \mathbf{C} \mathbf{r}(t), \quad (28)$$

with  $\mathbf{C} \in \mathbb{R}^{d \times (d \cdot l)}$  concatenating the  $\mathbf{C}_k$ , and the regressor vector

$$\mathbf{r} = [\mathbf{y}(t)^\top \quad \mathbf{y}(t - \Delta t)^\top \quad \dots \quad \mathbf{y}(t - (l-1)\Delta t)^\top]^\top \in \mathbb{R}^{d \cdot l}. \quad (29)$$

To estimate the MVAR parameters, we minimize the total squared prediction error over all sequences corresponding to different initial conditions. Let  $\{\mathbf{y}^{(n)}(t)\}_{t=0}^\tau$  denote the latent sequence for the  $n$ -th initial condition,  $n = 1, \dots, N_{ic}$ , and  $\tau = t_f - t_0$  the sequence length. The corresponding regressor is

$$\mathbf{r}^{(n)} = [\mathbf{y}^{(n)}(t)^\top \quad \mathbf{y}^{(n)}(t - \Delta t)^\top \quad \dots \quad \mathbf{y}^{(n)}(t - (l-1)\Delta t)^\top]^\top.$$

The Ordinary Least Squares (OLS) solution [90] minimizes

$$\min_{\mathbf{C}} \sum_{n=1}^{N_{ic}} \sum_{t=(l-1)\Delta t}^{\tau-\Delta t} \left\| \mathbf{y}^{(n)}(t + \Delta t) - \mathbf{C} \mathbf{r}^{(n)} \right\|_2^2, \quad (30)$$

yielding the closed-form solution

$$\mathbf{C}^\top = (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top \mathbf{Y}, \quad (31)$$

where  $\mathbf{R} \in \mathbb{R}^{M \times (d \cdot l)}$  stacks all regressor vectors and  $\mathbf{Y} \in \mathbb{R}^{M \times d}$  contains the target latent states. Temporal stationarity of the latent states was verified via the Augmented Dickey–Fuller (ADF) test [91].

The optimal lag order  $l$  is selected using the Bayesian Information Criterion (BIC) [90]:

$$\text{BIC}(l) = -2 \cdot \log(\hat{\mathcal{L}}_h(l)) + \log(N_\tau) \cdot T_c, \quad (32)$$

where  $\hat{\mathcal{L}}_h(l)$  is the maximum likelihood for lag  $l$  and  $T_c = l \cdot d^2$  is the number of autoregressive coefficients. The optimal lag minimizes  $\text{BIC}(l)$ . Once trained, the MVAR model provides a compact, interpretable representation of the latent temporal dynamics, suitable for forecasting, simulation, and uncertainty quantification.

### 2.2.2 Manifold-Informed SINDy

SINDy seeks a parsimonious description of the governing dynamics by representing the time evolution of the latent variables as a sparse linear combination of candidate nonlinear functions drawn from a predefined library [78, 92]. Within the present framework, SINDy is applied directly in the learned latent space, thereby avoiding the curse of dimensionality associated with discovering nonlinear PDEs in the original high-dimensional space (see also the discussion in [79]).

Let  $\mathbf{y}(t) \in \mathbb{R}^d$  denote the latent coordinates obtained via POD or DMs. Within the discrete map context, for one-step-ahead predictions, SINDy seeks to approximate the latent dynamics as

$$\mathbf{y}(t + \Delta t) = \Theta(\mathbf{r}) \Xi. \quad (33)$$

where  $\Theta(\mathbf{r}) \in \mathbb{R}^p$  is a library of  $p$  candidate nonlinear functions and  $\Xi \in \mathbb{R}^{p \times d}$  is a sparse coefficient matrix whose nonzero entries identify the active terms governing the dynamics. This continuous-time formulation provides conceptual motivation for the method.

Typical libraries include polynomial functions up to a prescribed order, Fourier basis functions, or combinations thereof, optionally augmented with cross terms. In the numerical experiments performed in this work, we consider polynomial libraries of order one, two and three, denoted by  $p(1)$ ,  $p(2)$  and  $p(3)$ , respectively, as well as Fourier libraries, denoted by  $f$ . When explicitly excluding cross-coupling polynomial terms, we use the notation  $p_{nc}(\cdot)$ . When a Fourier library is employed,  $\Theta(\cdot)$  consists of sine and cosine functions of the latent coordinates and their delays, enabling the identification of oscillatory and advective structures in the latent dynamics.

The sparse coefficient matrix  $\Xi$  is identified via sparse regression using sequentially thresholded least squares or equivalent  $\ell_1$ -regularized optimization procedures [78]. As in the MVAR case, SINDy operates entirely in the latent space and is therefore fully compatible with the proposed PDE-free, manifold-informed framework. Both families of ROMs are systematically evaluated and compared in Section 4.

### 2.3 Assessing the Reconstruction Error

Once latent coordinates are obtained (via POD or DMs), a high-dimensional state  $\rho \in \mathbb{R}^N$  can be projected into the latent space and then lifted back to the high-dimensional space to obtain a reconstructed field  $\hat{\rho} \in \mathbb{R}^N$ . The reconstruction quality is evaluated against the ground-truth  $\rho$  using the absolute and relative  $\ell_2$  errors:

$$\varepsilon_2(t) = \|\rho(t, \cdot, \cdot) - \hat{\rho}(t, \cdot, \cdot)\|_2, \quad \varepsilon_2^r(t) = \frac{\|\rho(t, \cdot, \cdot) - \hat{\rho}(t, \cdot, \cdot)\|_2}{\|\rho(t, \cdot, \cdot)\|_2}, \quad (34)$$

as well as the *Wasserstein-1 distance* ( $W_1$ ) [93–96] defined as

$$W_1(t) = \inf_{\gamma \in \Gamma(\rho, \hat{\rho})} \int_{\Omega \times \Omega} \|x - y\|_2 d\gamma(x, y), \quad (35)$$

where  $\Gamma(\rho, \hat{\rho})$  is the set of joint distributions (couplings) with marginals  $\rho$  and  $\hat{\rho}$ , and  $\|x - y\|_2$  is the Euclidean distance.

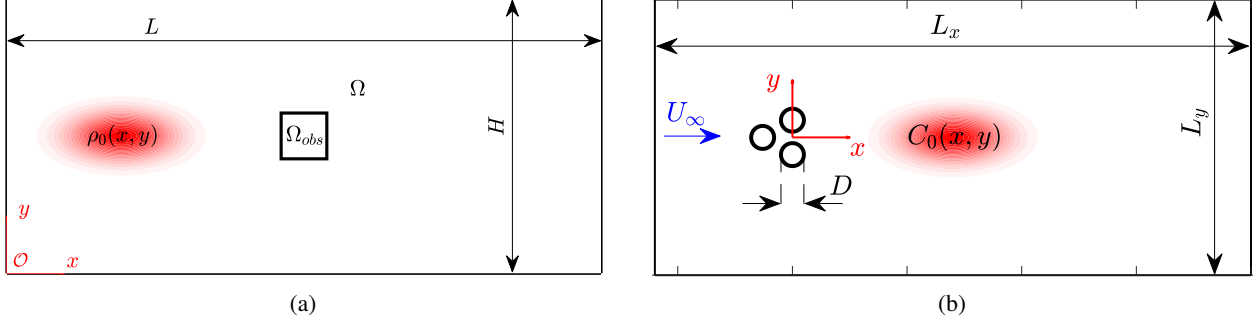


Figure 2: Schematic representation of the case studies: (a) pedestrian motion in a corridor with an obstacle is modeled using the Hughes model; (b) a fluid dynamics setup, where a passive tracer is transported by the Navier–Stokes velocity field produced by a three-cylinder “fluidic pinball” arrangement.

The Wasserstein distance is particularly suitable for spatio-temporal dynamics as it accounts for both magnitude and spatial rearrangements: small translations of a density pattern yield small  $W_1$ , unlike  $l_2$  or  $l_\infty$  norms that can overestimate error due to local misalignments. Here, the computation of the Wasserstein distances is performed in MATLAB using a simplified, discrete approach by summing absolute differences between sorted samples [97].

Here, based on the distributions of  $\varepsilon_2(t)$ ,  $\varepsilon_2^r(t)$ ,  $W_1(t)$ , across all snapshots and initial conditions, we report the mean values and the 10<sup>th</sup>–90<sup>th</sup> percentiles.

### 3 Case Studies

For our illustration, we considered two benchmark problems with mass conservation: (a) The Hughes model of crowd dynamics describing individuals minimizing travel time, formalized as a coupled PDE system in which the density evolves according to a continuity equation, while motion follows the gradient of an eikonal potential that naturally guides agents around obstacles and high-density regions; in the present case study, we considered the Hughes model applied to pedestrian flow in a corridor with an obstacle (see Fig. 2(a)); (b) a fluid dynamics problem, which involves the spatio-temporal evolution of a passive tracer advected by a Navier–Stokes velocity generated by a three-cylinder (fluidic pinball) configuration (see Fig. 2(b)).

#### 3.1 Hughes Model for Crowd Dynamics

A seminal model for approximating behavioral crowd dynamics via PDEs is the continuum Hughes model [98–101], which combines a conservation law for the crowd density with an Eikonal equation to compute shortest-time paths to a common goal. Crucially, the *optimal* path selection is assumed to depend on the global distribution of pedestrians in the environment, necessitating a dynamic resolution of the Eikonal equation at every time step. Although the local density–velocity relationship appears explicitly in the conservation law, the coupling with the Eikonal equation imparts a global character to the dynamics. This allows the model to reproduce complex behavioral features such as obstacle avoidance and re-routing in congested environments [102]. Both theoretical investigations and empirical studies support the notion that pedestrian speed and direction are influenced by the full spatial distribution of the crowd, rather than just local surroundings [17]. Consequently, while such global dependencies enhance model fidelity, they also introduce substantial challenges in terms of analytical tractability and computational cost. This motivates the need for efficient and compact representations of crowd motion—specifically, ROMs accounting for the *unknown hidden behavioral* dynamics [1, 17, 20, 20, 103, 104]—so as to retain the essential features required to reproduce its global evolution.

The Hughes model [98] relies on three behavioral hypotheses: (i) pedestrian speed  $f(\rho)$  depends solely on the local density  $\rho$ , following empirical pedestrian-flow relationships [105, 106]; (ii) pedestrians move in the direction that minimizes a potential function  $\phi$ , representing perceived travel effort or time; and (iii) individuals aim to minimize travel time while avoiding high-density regions, modeled via a separable cost function combining speed  $f(\rho)$  and a discomfort factor  $g(\rho)$ . This yields a coupled PDE system for crowd density  $\rho$  and potential  $\phi$ :

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x} \left( \rho f(\rho) \frac{\partial \phi}{\partial x} \frac{1}{\|\nabla \phi\|} \right) + \frac{\partial}{\partial y} \left( \rho f(\rho) \frac{\partial \phi}{\partial y} \frac{1}{\|\nabla \phi\|} \right), \quad (36a)$$

$$\|\nabla \phi\| = \frac{1}{f(\rho)g(\rho)}, \quad (36b)$$

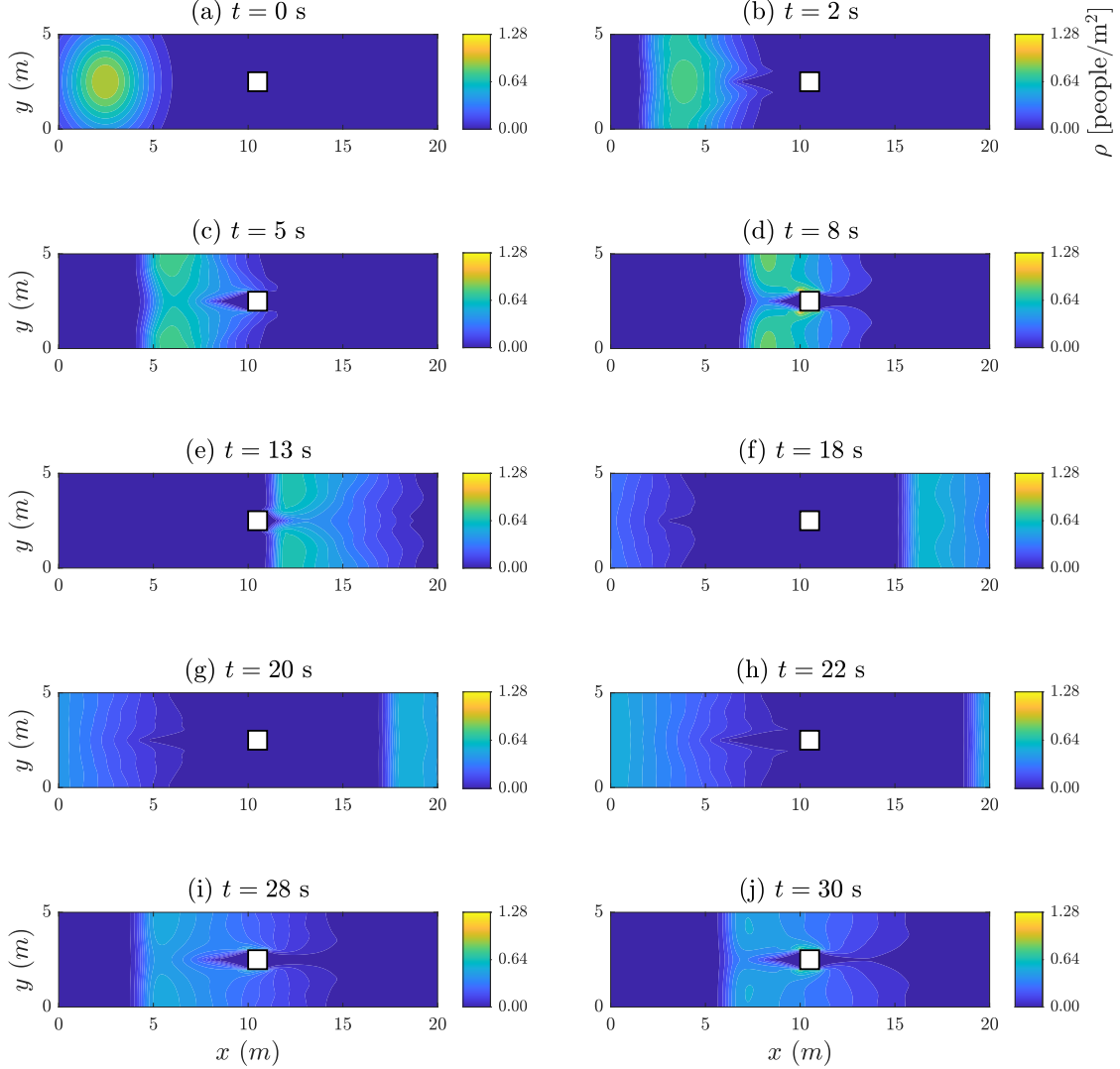


Figure 3: Contour maps of the Hughes PDE solution  $\rho(t, x, y)$  for the Gaussian initial condition  $\rho_0(x, y)$  (see Eq. (40) in A) with  $x_0 = 2.5$  m,  $y_0 = 2.5$  m,  $\sigma_x = 1.8$  m,  $\sigma_y = 1.8$  m, at different time instants: (a)  $t = 0$  s; (b)  $t = 2$  s; (c)  $t = 5$  s; (d)  $t = 8$  s; (e)  $t = 13$  s; (f)  $t = 18$  s; (g)  $t = 20$  s; (h)  $t = 22$  s; (i)  $t = 28$  s; (j)  $t = 30$  s.

with  $\|\nabla\phi\| = \sqrt{(\partial\phi/\partial x)^2 + (\partial\phi/\partial y)^2}$ . Here,  $f(\rho)$  represents the pedestrian speed, while  $g(\rho)$  accounts for the pedestrian discomfort at high crowd densities. A linear velocity–density relationship is considered, expressed as

$$f(\rho) = v_f \left(1 - \frac{\rho}{\rho_m}\right), \quad (37)$$

where  $v_f$  denotes the free-flow speed and  $\rho_m$  the maximum pedestrian density. In this study, these parameters are set to  $v_f = 1$  m/s and  $\rho_m = 5$  people/m<sup>2</sup>, respectively, and the discomfort factor is  $g(\rho) = 1$ . Additional details on data generation, as well as initial and boundary conditions, are provided in A.

A typical evolution of the density field is shown in Fig. 3. Initially, the crowd density (see Eq. (40) in A) exhibits dominant streamwise motion (Figs. 3 (a),(b)), driven by the collective intent to reach the exit. Interestingly, the initial splitting of the crowd occurs before physically encountering the obstacle. This reflects the tendency of pedestrians to adjust trajectories in order to minimize perceived travel cost. As the crowd approaches the obstacle, the density bifurcates into two nearly symmetric (due to the given initial condition) streams (Figs. 3(c),(d)), corresponding to two primary flows circumventing the obstruction. After passing the obstacle (Fig. 3(e)), the two main streams gradually merge (Fig. 3(f)), reestablishing a unified front-like wave, and so on (Figs. 3(g),(h),(i),(j)). Periodic boundaries reintroduce the crowd at the inlet, generating an almost cyclic pattern of inflow, splitting, and merging.

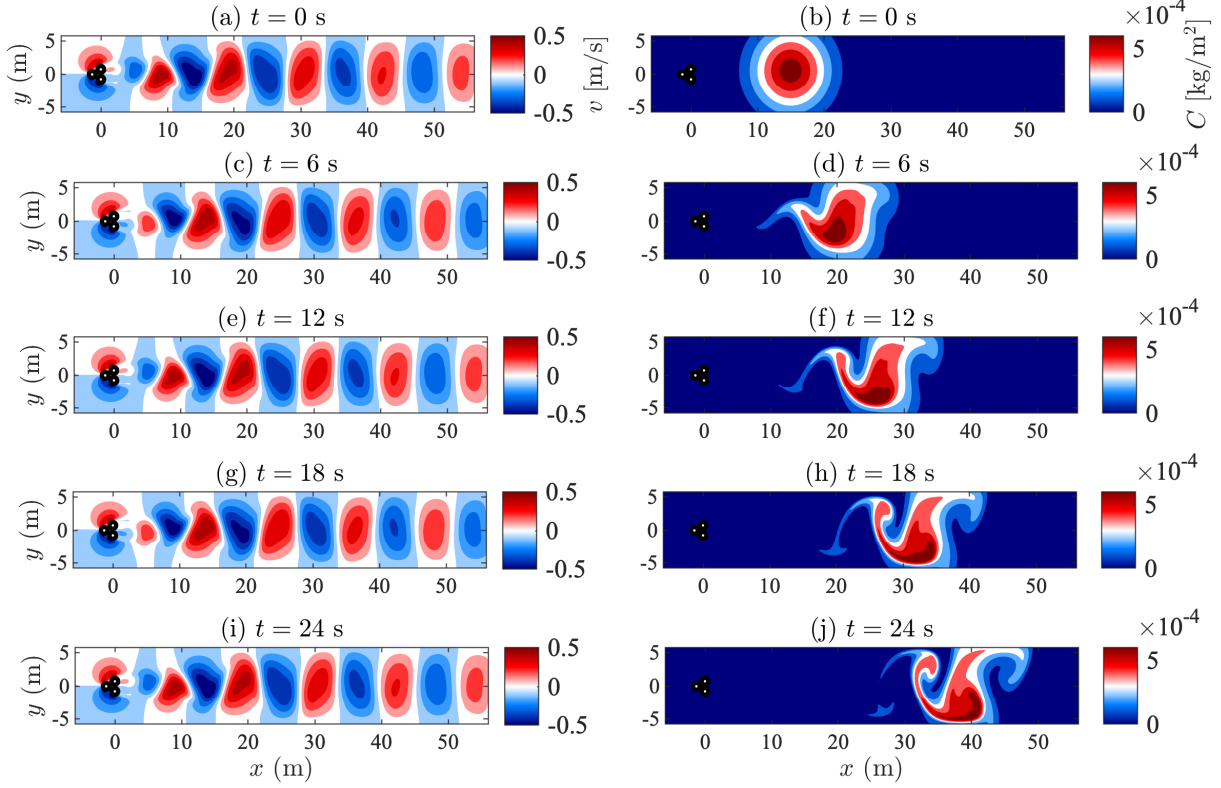


Figure 4: Contour maps of the Navier-Stokes cross-stream velocity component  $v(t, x, y)$  (left panels) and the passive tracer distribution  $C(t, x, y)$  corresponding to the Gaussian initial condition  $C_0(x, y)$  (see Eq. (53) in C) with  $x_0 = 14.83$  m,  $y_0 = 0.44$  m,  $\sigma_x = 4$  m,  $\sigma_y = 4.3$  m, at different time instants: (a)-(b)  $t = 0$  s; (c)-(d)  $t = 6$  s; (e)-(f)  $t = 12$  s; (g)-(h)  $t = 18$  s; (i)-(j)  $t = 24$  s.

### 3.2 Passive Tracer Transport in Navier–Stokes Flow

The fluidic pinball is a flow configuration consisting of three rotatable cylinders of equal diameter  $D$ , whose axes are located at the vertices of an equilateral triangle, as sketched in Fig. 2(b). The triangle has a centre-to-centre side length equal to  $1.5D$  and is immersed in a viscous, incompressible flow with uniform upstream velocity  $U_\infty$ . The governing parameter of this configuration is the Reynolds number,

$$Re = \frac{U_\infty D}{\nu}, \quad (38)$$

where  $\nu$  denotes the fluid kinematic viscosity. As  $Re$  increases, the flow undergoes a sequence of bifurcations—including Andronov–Hopf, pitchfork, and Neimark–Sacker bifurcations—eventually leading to chaotic dynamics [107]. In this work, we focus on the regime at  $Re = 30$ , where the dynamics are characterized by spatio-temporal periodic oscillations.

The dataset is obtained from direct numerical simulations of the incompressible two-dimensional Navier–Stokes equations coupled with a linear advection equation for a passive scalar tracer,

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad (39a)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (39b)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right), \quad (39c)$$

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial x} + v \frac{\partial C}{\partial y} = 0. \quad (39d)$$

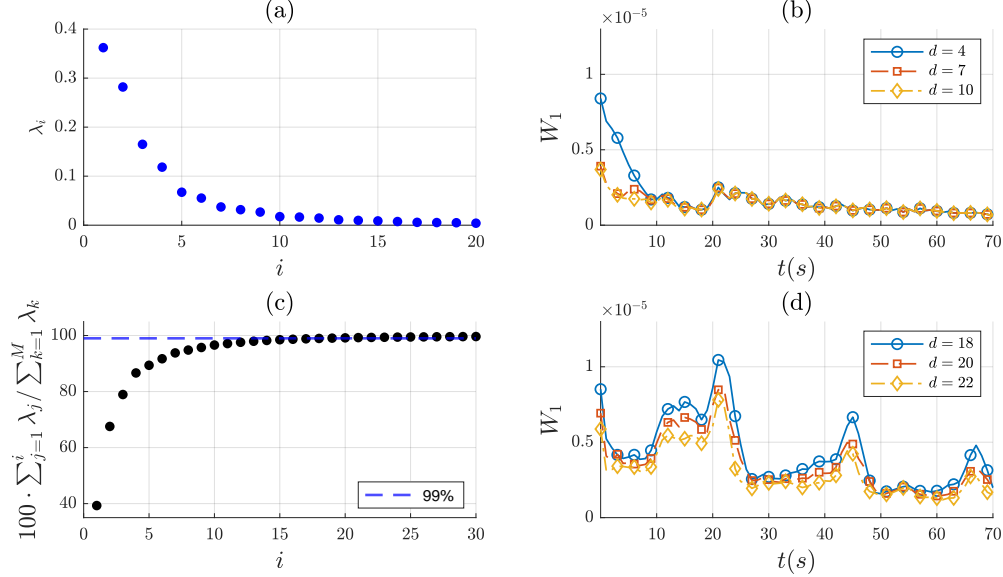


Figure 5: Crowd dynamics paradigm: (a) DMs eigenvalues  $\lambda_i$ ; (b) baseline mean reconstruction error  $W_1$  computed over  $d = 4, 7,$  and  $10$  DMs coordinates for the  $N_{\text{train}} = 40$  training simulations; (c) cumulative distribution of POD eigenvalues (black circles) and 99% cumulative variance threshold (blue dashed line); (d) baseline mean reconstruction error  $W_1$  for  $d = 18, 20,$  and  $22$  POD coordinates on the training set.

Here,  $u$  and  $v$  denote the streamwise ( $x$ ) and cross-stream ( $y$ ) velocity components, respectively,  $p$  is the pressure,  $\rho$  is the (constant) density, and  $C$  is the passive tracer concentration. Additional details on data generation, as well as initial and boundary conditions, are provided in C.

In the Navier–Stokes system coupled with a passive tracer, the tracer concentration alone does not provide full observability of the underlying dynamics. The velocity field and pressure naturally act as hidden variables, mediating the transport, mixing, and deformation processes that govern the tracer evolution. As a result, the tracer dynamics effectively evolve on a latent manifold shaped by these unobserved/hidden flow variables.

In Fig. 4, typical evolutions of the cross-stream velocity field  $v$  (left panels) and of the passive tracer density  $C$  (right panels) are reported. Note that the time  $t = 0$  corresponds to the instant at which the tracer is injected into the domain (see Eq. (53) in C). The velocity field exhibits global spatio-temporal oscillations associated with the periodic vortex shedding that develops downstream of the pinball configuration in this Reynolds number regime (left panels in Fig. 4). These oscillations arise from the *Andronov–Hopf bifurcation* undergone by the pinball flow at  $Re \approx 18$ , beyond which the steady base flow loses stability and a time-periodic limit cycle emerges [107]. The resulting von Kármán-type vortex street dominates the wake dynamics. As a consequence, the passive tracer is advected by the underlying velocity field and progressively deforms, stretches, and mixes as it is entrained by the coherent vortical structures. In particular, the tracer distribution closely follows the alternating vortex-shedding pattern, with regions of high tracer concentration being wrapped around the vortices and transported downstream. This process leads to an increasingly filamented tracer field whose spatial organization mirrors that of the underlying vorticity field, highlighting the strong coupling between wake dynamics and scalar transport in this regime.

## 4 Numerical Results

### 4.1 The Crowd Dynamics Paradigm

Here, we evaluate the proposed methodology using numerical simulations of the Hughes PDE model for pedestrian dynamics. For our illustration, we have created  $N_{ic} = 100$  distinct initial conditions by uniformly sampling the parameters of the 2D Gaussian distribution (see Eq. (40) in A) within the ranges  $x_0, y_0 \in [1.5, 3.5]$  and  $\sigma_x, \sigma_y \in [1.6, 2]$ , and integrate each simulation until  $t_f = 70$  s. From these, we constructed three datasets: a training set  $X_{\text{train}}$  containing  $N_{\text{train}} = 40$  randomly selected simulations, a validation set with  $N_{\text{val}} = 20$  samples for model selection, and a test set  $X_{\text{test}}$  including the remaining  $N_{\text{test}} = 40$  simulations for out-of-sample evaluation. The construction of  $X_{\text{train}}, X_{\text{val}},$  and  $X_{\text{test}}$ —including parameter ranges and temporal resolution—follows the protocol detailed in Section 3.1.

In the first stage, as previously discussed, we employed both POD and DMs to learn a low-dimensional parametrization of the manifold. Given the high dimensionality of each snapshot ( $N = N_x \times N_y = 200 \times 50$ ) and the large ensemble of training data ( $N_{\text{train}} \times N_t = 40 \times 700$ ), performing manifold learning on the full dataset would be computationally prohibitive. Moreover, in the present crowd dynamics configuration, there exists a single nonlinear basin of attraction: trajectories originating from different initial conditions converge toward similar long-term behavior, with most variability concentrated in the initial transient phase and around the obstacle. During this transient—when eikonal-driven optimal paths bifurcate and spatial patterns (e.g., splitting flows and front-wave formations) emerge—the dynamics are particularly informative and of primary interest (see Fig. 3 and related discussion in Section 3.1). Consequently, we learn the restriction and lifting operators from a subset of snapshots emphasizing the transient dynamics: two-thirds of the selected snapshots are drawn from the first 10 s of each simulation, and one-third from the remaining time. In total,  $X_{\text{train}}$  contains 6,000 snapshots for operator construction, with approximately 4,000 from the initial 10 s and 2,000 from the later interval. This design keeps the computational cost manageable while enhancing the reconstruction of early, fast evolving dynamics without neglecting the slow, more homogeneous regime that follows.

Model	$\varepsilon_2 (\times 10^{-3})$	$\varepsilon_2^r (\times 10^{-1})$	$W_1 (\times 10^{-6})$	#params
POD-MVAR ( $d = 20, l = 5$ )	1.40 (0.92, 2.00)	0.97 (0.62, 1.31)	6.53 (3.89, 9.94)	2000
DMs-MVAR ( $d = 10, l = 8$ )	1.10 (0.37, 1.80)	0.69 (0.28, 1.15)	1.64 (0.48, 3.20)	800
POD-SINDy ( $d = 22, p(1) + f$ )	2.35 (1.54, 3.36)	1.59 (1.06, 2.23)	8.52 (6.29, 11.43)	1470
DMs-SINDy ( $d = 7, p_{nc}(2)$ )	1.32 (0.74, 2.35)	0.88 (0.50, 1.56)	3.59 (0.84, 11.12)	250

Table 1: Crowd dynamics paradigm. Summary of the selected POD-informed ROMs and DMs-informed ROMs performance on the test set  $X_{\text{test}}$ : absolute  $\ell_2$  error  $\varepsilon_2$ , relative  $\ell_2$  error  $\varepsilon_2^r$ , and Wasserstein-1 distance  $W_1$  in terms of average over time and 10<sup>th</sup>–90<sup>th</sup> percentiles. The last column reports the number of parameters of each ROM.

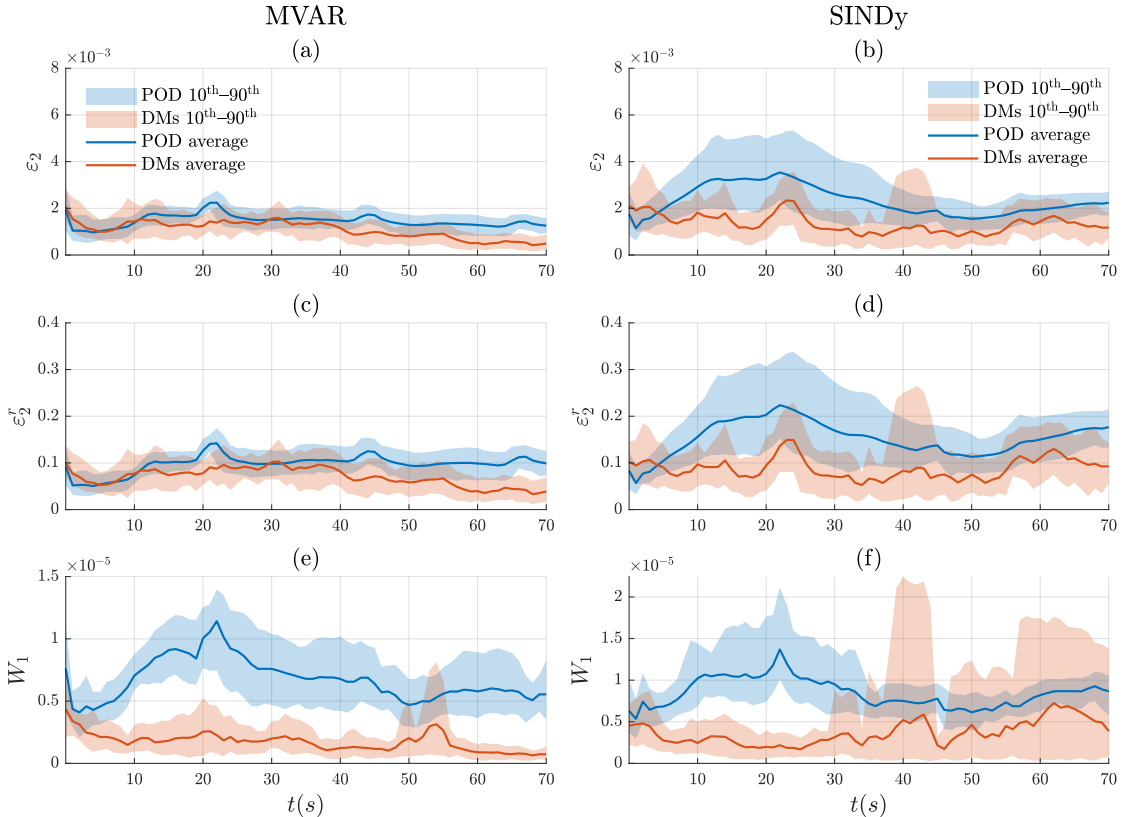


Figure 6: Crowd dynamics paradigm: time evolution of reconstruction errors on the test set  $X_{\text{test}}$ . (a,b) Absolute  $\ell_2$  error ( $\varepsilon_2$ ); (c,d) relative  $\ell_2$  error ( $\varepsilon_2^r$ ); (e,f) Wasserstein distance ( $W_1$ ) for the selected POD-MVAR ( $d = 20, l = 5$ ), DMs-MVAR ( $d = 10, l = 8$ ), POD-SINDy ( $d = 22, p(1) + f$ ), and DMs-SINDy ( $d = 7, p_{nc}(2)$ ) reduced-order-models across the  $N_{\text{test}} = 40$  test simulations. In all panels, solid lines denote the mean error between the high-dimensional Hughes PDE solution (ground truth) and the ROM predictions, while shaded bands indicate the 10<sup>th</sup>–90<sup>th</sup> percentiles.

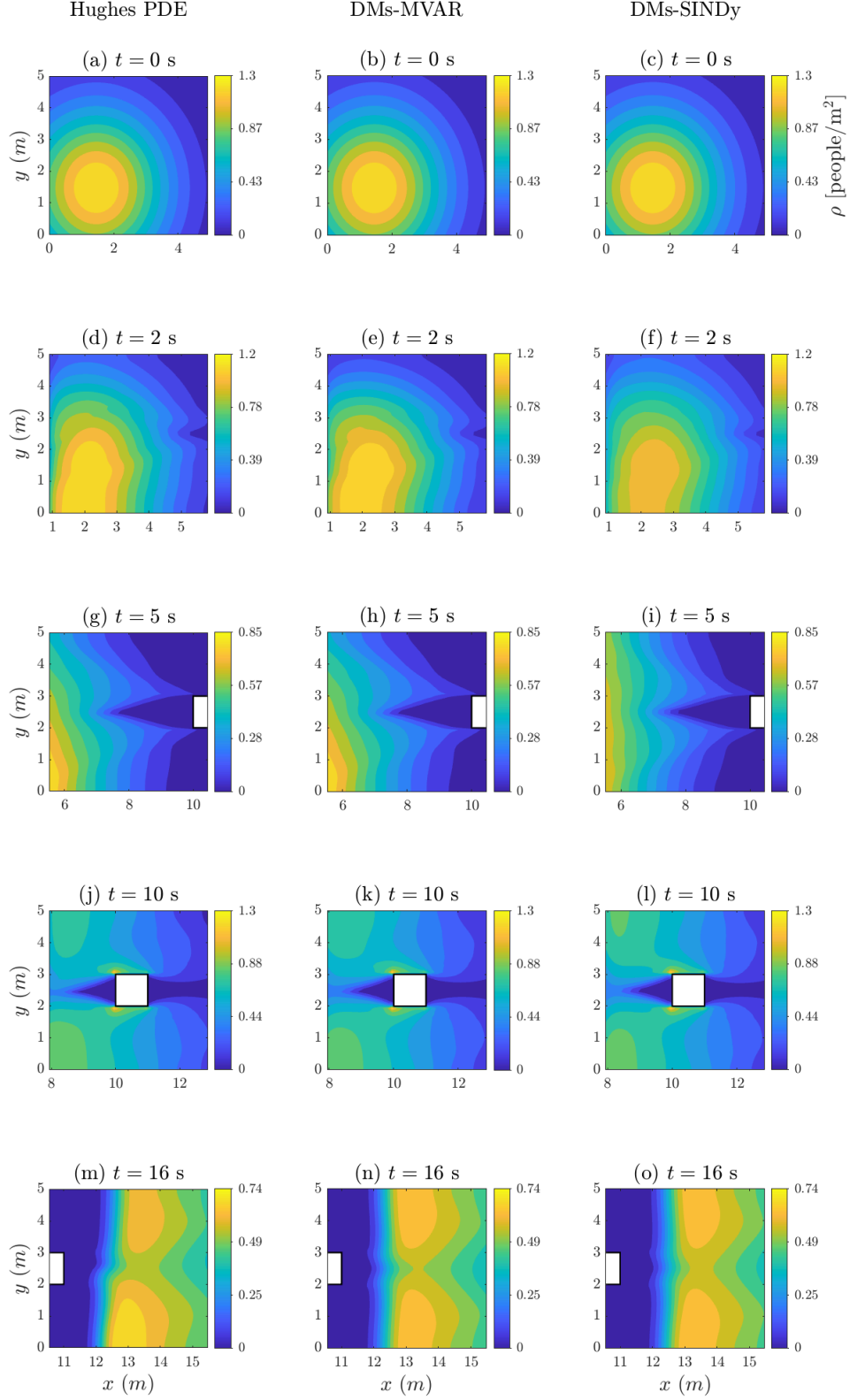


Figure 7: Comparison between the ground-truth Hughes PDE solution (left column), the DMs-informed MVAR ( $d = 10$ ,  $l = 8$ ) prediction (center column), and the DMs-informed SINDy ( $d = 7$ ,  $p_{nc}(2)$ ) prediction (third column) at different time instants  $t$  for an unseen Gaussian initial condition  $\rho_0(x, y)$  (see Eq. (40)) with parameters  $x_0 = 1.5$  m,  $y_0 = 1.5$  m,  $\sigma_x = 1.6$  m, and  $\sigma_y = 1.8$  m, belonging to the test set  $X_{\text{test}}$ : ((a)–(c))  $t = 0$  s; ((d)–(f))  $t = 2$  s; ((g)–(i))  $t = 5$  s; ((j)–(l))  $t = 10$  s; ((m)–(o))  $t = 16$  s.

Fig. 5 summarizes the results of the DMs and POD embeddings computed from the  $N_{\text{train}} = 40$  training simulations. Fig. 5(a) reports the DMs eigenvalues  $\lambda_i$ , while Fig. 5(b) shows the mean baseline reconstruction error in terms of the Wasserstein–1 distance ( $W_1$ ) for  $d = 4, 7$ , and 10 DMs coordinates. It can be noted that all error distributions converge to the same value for  $t > 20$  s, but only the embeddings with  $d = 7$  and  $d = 10$  exhibit low errors during the transient phase ( $t < 20$  s). Hence, we primarily consider the first  $d = 7$  DMs and later test  $d = 10$  to evaluate whether additional coordinates improve latent-space predictions. Fig. 5(c) displays the cumulative distribution of the POD eigenvalues, and Fig. 5(d) reports the corresponding mean reconstruction errors  $W_1$  for  $d = 18, 20$ , and 22 POD modes. We find that  $d = 20$ –22 POD modes—capturing more than 99% of the cumulative variance—are required to achieve reconstruction accuracy comparable to that obtained with only  $d = 7$ –10 DMs coordinates. Overall, DMs achieve equivalent, and in the transient regime even superior, reconstruction accuracy using far fewer coordinates than POD.

In the second step, we trained two families of manifold-informed ROMs: one based on POD coordinates and another based on Diffusion Maps (DMs) coordinates. For each family, we employed both MVAR and SINDy models. A detailed analysis of the different ROMs performance on the training and validation datasets, including the dependence on the number of retained coordinates and modes, is provided in B. This analysis allowed us to select the four best models, based on the validation set, one for each combination of embedding algorithm (POD and DMs) and surrogate model (MVAR and SINDy). In the following, we focus on these selected models and their performance on the test set.

Fig. 6 shows the temporal evolution of the test-set error metrics, for the selected POD- and DMs-informed MVARs and SINDy ROMs: absolute  $\ell_2$  error ( $\varepsilon_2$ ), relative  $\ell_2$  error ( $\varepsilon_2^r$ ), and Wasserstein–1 distance ( $W_1$ ) in the reconstructed high-dimensional space, together with 10<sup>th</sup>–90<sup>th</sup> percentiles. We note that all error metrics in these diagrams are initialized at the first predicted snapshot.

The time-averaged errors and associated percentiles for the four models are also reported in Table 1. The POD-MVAR and DMs-MVAR models are constructed with  $d = 20$  and  $d = 10$  variables and with  $l = 5$  and  $l = 8$  time delays, respectively. The POD-SINDy and DMs-SINDy models are constructed with  $d = 22$  and  $d = 7$  variables, respectively, and with a polynomial library of order one augmented with Fourier features ( $p(1) + f$ ) and a polynomial library of order two without cross terms ( $p_{nc}(2)$ ), respectively. As shown, the DMs-informed models consistently outperform the POD-informed models on average across all metrics.

In Fig. 7, we provide representative snapshots of the ground-truth Hughes PDE solution (left panels), the DMs-informed MVAR (center panels), and the DMs-informed SINDy (right panels) predictions, for an unseen initial condition from the test set. As shown, the DMs-informed ROMs are able to capture spatial asymmetries and nonlinear features of the evolving field with high fidelity. It is interesting to highlight that the DMs-informed SINDy (with just one delay) and DMs-informed MVAR (with multiple delays) exhibit comparable predictive accuracy. We note that, as detailed in Tables 3 and 4 in B summarizing the results on the training and validation sets, respectively, while the POD-informed SINDy performs relatively well—even with just one delay—though still worse than the DMs-informed SINDy, the POD-informed MVAR with one delay fails entirely to capture the dynamics.

## 4.2 The Fluid dynamics Paradigm

Here, we evaluate the proposed methodology using numerical simulations of the Navier-Stokes flow with passive tracer. We have created  $N_{ic} = 80$  distinct initial conditions by uniformly sampling the parameters of the 2D Gaussian distribution (see Eq. (53) in C) within the ranges  $x_0 \in [14, 16]$  m,  $y_0 \in [-1, 1]$  m, and  $\sigma_x, \sigma_y \in [4, 5]$  m, and integrate each simulation until  $t_f = 24$  s. From these, we constructed three datasets: a training set  $X_{\text{train}}$  containing  $N_{\text{train}} = 35$  randomly selected simulations, a validation set with  $N_{\text{val}} = 10$  samples for model selection, and a test set  $X_{\text{test}}$  including the remaining  $N_{\text{test}} = 35$  simulations for out-of-sample evaluation. The construction of  $X_{\text{train}}$ ,  $X_{\text{val}}$ , and  $X_{\text{test}}$ —including parameter ranges and temporal resolution—follows the protocol detailed in Section 4.1 for the crowd dynamics paradigm.

In the first stage, as previously discussed, we employed both POD and DMs to learn a low-dimensional parametrization of the manifold. Fig. 8 summarizes the results of the DMs and POD embeddings computed from the  $N_{\text{train}} = 40$  training simulations. Fig. 8(a) reports the DMs eigenvalues  $\lambda_i$ , while Fig. 8(b) shows the mean baseline reconstruction error in terms of the Wasserstein–1 distance  $W_1$  for  $d = 2, 4$ , and 6 DMs coordinates. Fig. 8(c) displays the cumulative distribution of the POD eigenvalues, and Fig. 8(d) reports the corresponding mean reconstruction errors  $W_1$  for  $d = 18, 20$ , and 22 POD modes.

In the second step, as for the crowd dynamics paradigm, we trained again two families of manifold-informed ROMs: one based on POD embedding and another based on DMs embedding, and for each family of embeddings, we employed both MVARs and SINDy models. A detailed analysis of the different ROMs performance on the training and validation datasets, including the dependence on the number of retained coordinates and modes, is provided in D. From these, we

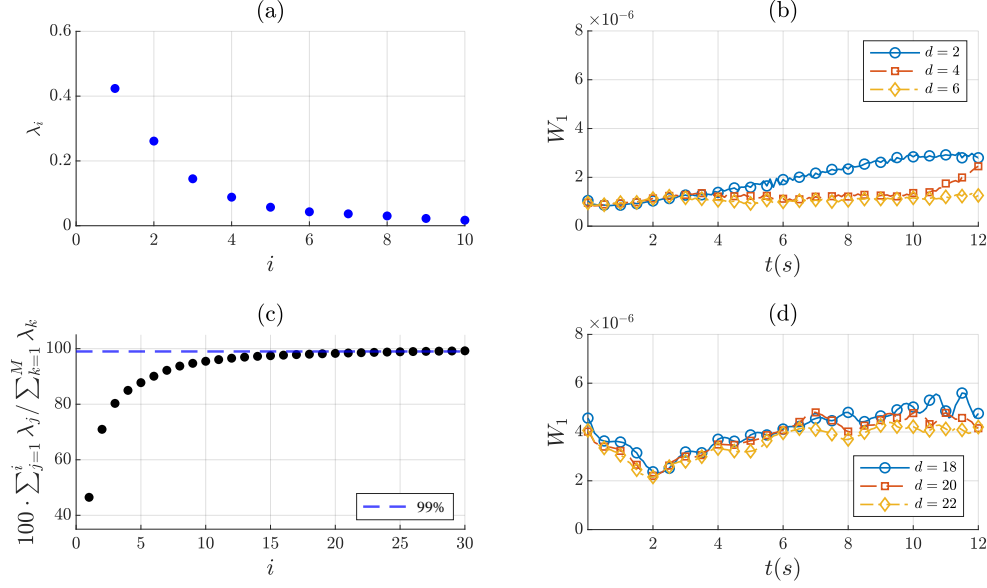


Figure 8: Fluid dynamics paradigm: (a) DMs eigenvalues  $\lambda_i$ ; (b) baseline mean reconstruction error  $W_1$  computed over  $d = 2, 4,$  and  $6$  DMs coordinates for the  $N_{\text{train}} = 35$  training simulations; (c) cumulative distribution of POD eigenvalues (black circles) and 99% cumulative variance threshold (blue dashed line); (d) baseline mean reconstruction error  $W_1$  for  $d = 18, 20,$  and  $22$  POD coordinates on the training set.

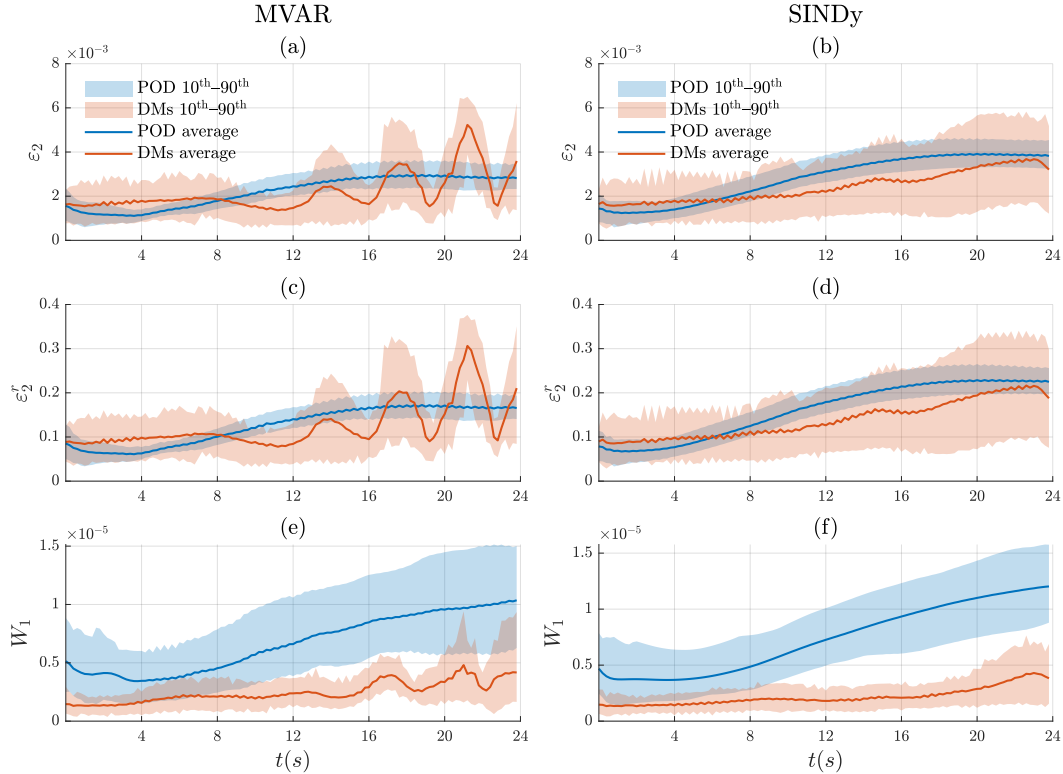


Figure 9: Fluid dynamics paradigm: time evolution of reconstruction errors on the test set  $X_{\text{test}}$ . (a,b) Absolute  $\ell_2$  error ( $\varepsilon_2$ ); (c,d) relative  $\ell_2$  error ( $\varepsilon_2^r$ ); (e,f) Wasserstein distance ( $W_1$ ) for the selected POD-MVAR ( $d = 16, l = 3$ ), DMs-MVAR ( $d = 6, l = 3$ ), POD-SINDy ( $d = 20, p(1) + f$ ), and DMs-SINDy ( $d = 4, p(3)$ ) reduced-order-models across the  $N_{\text{test}} = 35$  test simulations. In all panels, solid lines denote the mean error between the high-dimensional Navier-Stokes PDE solution (ground truth) and the ROM predictions, while shaded bands indicate the 10<sup>th</sup>–90<sup>th</sup> percentiles.

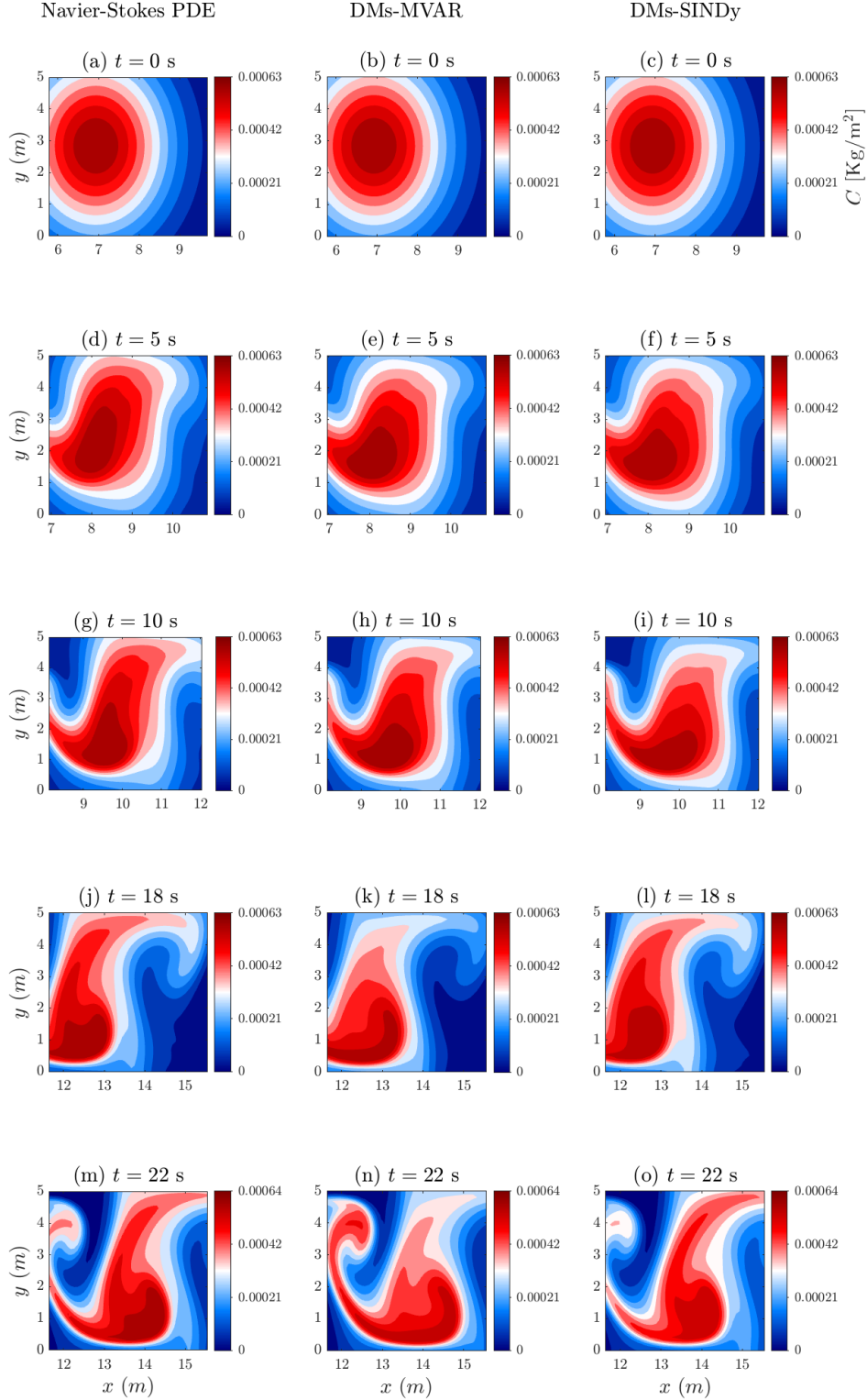


Figure 10: Comparison between the ground-truth Navier-Stokes PDE solution (left column), the DMs-informed MVAR prediction ( $d = 6$ ,  $l = 3$ ) (center column), and the DMs-informed SINDy prediction ( $d = 4$ ,  $p(3)$ ) (right column) at different time instants  $t$  for an unseen Gaussian initial condition  $\rho_0(x, y)$  (see Eq. (53)) with parameters  $x_0 = 14.50$  m,  $y_0 = -0.79$  m,  $\sigma_x = 4.19$  m, and  $\sigma_y = 4.50$  m, belonging to the test set  $X_{\text{test}}$ : ((a)–(c))  $t = 0$  s; ((d)–(f))  $t = 5$  s; ((g)–(i))  $t = 10$  s; ((j)–(l))  $t = 18$  s; ((m)–(o))  $t = 22$  s.

Model	$\varepsilon_2 (\times 10^{-3})$	$\varepsilon_2^r (\times 10^{-1})$	$W_1 (\times 10^{-6})$	#params
POD-MVAR ( $d = 16, l = 3$ )	2.20 (1.74, 2.74)	1.26 (1.04, 1.52)	6.67 (3.84, 10.50)	768
DMS-MVAR ( $d = 6, l = 3$ )	2.23 (1.19, 3.42)	1.27 (0.68, 1.92)	2.48 (1.19, 4.32)	108
POD-SINDy ( $d = 20, p(1) + f$ )	2.81 (2.29, 3.48)	1.61 (1.35, 1.94)	7.32 (4.69, 10.36)	1219
DMS-SINDy ( $d = 4, p(3)$ )	2.41 (1.04, 3.96)	1.38 (0.61, 2.23)	2.17 (0.98, 3.56)	98

Table 2: Fluid dynamics paradigm. Summary of the selected POD-informed ROMs and DMs-informed ROMs performance on the test set  $X_{\text{test}}$ : absolute  $\ell_2$  error  $\varepsilon_2$ , relative  $\ell_2$  error  $\varepsilon_2^r$ , and Wasserstein–1 distance  $W_1$  in terms of average over time and 10<sup>th</sup>–90<sup>th</sup> percentiles. The last column reports the number of parameters of each ROM.

have selected the four best models according to the validation test, one for each combination of embedding algorithm (POD and DMS) and ROMs (MVAR and SINDy). In the following, we focus on these selected models and their performance on the test set.

The time-averaged errors and associated percentiles for the four models are reported in Table 2. Note that the POD-MVAR and DMS-MVAR ROMs are constructed using  $d = 16$  and  $d = 6$  variables and  $l = 3$  time delays, respectively. The POD-SINDy, and DMS-SINDy ROMs are constructed with  $d = 20$ , and only  $d = 4$  variables, respectively, and with a polynomial library of order one augmented with Fourier features ( $p(1) + f$ ) and a polynomial library of order three ( $p(3)$ ).

Fig. 9 shows the temporal evolution of the test-set error metrics for the selected POD- and DMS-informed MVAR and SINDy ROMs: absolute  $\ell_2$  error ( $\varepsilon_2$ ), relative  $\ell_2$  error ( $\varepsilon_2^r$ ), and Wasserstein–1 distance ( $W_1$ ) in the reconstructed high-dimensional space, together with 10<sup>th</sup>–90<sup>th</sup> percentiles. Note that all curves are initialized at the first predicted snapshot. As shown, the DMS-informed MVAR models exhibit oscillatory behavior despite comparable short-term accuracy in all metrics. On average, the DMS-informed SINDy models are more accurate with respect to  $W_1$ , and comparable to the other two metrics albeit with wider 10<sup>th</sup>–90<sup>th</sup> percentiles spreads. We also note that the POD-informed ROMs (both MVARs and SINDy) result in a divergent  $W_1$  metric.

Finally, in Fig. 10, we provide representative snapshots of the ground-truth Navier-Stokes PDE solution (left panels), the DMS-informed MVAR (center panels), and the DMS-informed SINDy (right panels) predictions, for an unseen initial condition from the test set. As illustrated by these snapshots and throughout the test set, the DMS-informed SINDy captures more accurately the spatio-temporal dynamics, a feature that it is not fully reflected by the global metrics reported in Table 2. In fact, the DMS-informed MVAR prediction shows a spatio-temporal shift relative to the ground-truth solution, which explains the oscillatory behavior observed in all the metrics reported in Fig. 9 (red curves in panels (a), (c), and (e)). This can be appreciated in particular by comparing Fig. 10(m) and Fig. 10(n), displaying the ground-truth Navier–Stokes solution and the DMS-MVAR prediction, respectively, at  $t = 22$  s, which is the time instant of maximum reconstruction error.

## 5 Conclusions

For systems with conservation laws, learning the full high-dimensional solution operator for a mass-conserving PDE is not trivial [108–111]: small model errors compound under long-time integration, producing significant drifts or even blow-ups. The proposed PDE-free framework overcomes this limitation and applies broadly to systems that can, in principle, be described by PDEs—including those with hidden or unobserved variables—by avoiding the need to learn the full spatio-temporal operator. Instead, we construct reduced-order models (ROMs) in latent spaces, recovering an effective ODE-like operator using nonlinear manifold learning techniques and in particular Diffusion maps (DMs). This approach is far more computationally efficient than learning the full PDE solution operator, a task that suffers from the curse of dimensionality. Once trained, the manifold-informed ROM allows reconstruction of the original spatio-temporal fields via the manifold *mass-preserving* pre-image problem, reproducing the high-dimensional solution operator *implicitly* and *on demand*. Hence, with such ROMs, one can perform accurate long-horizon simulations.

For these ROMs to achieve high accuracy over long time-horizons, the latent dynamics must be represented by a well-parameterized embedding of the underlying manifold. Here and in other works [112, 113], we have shown that for that task, DMs outperform POD—which remains dominant in the vast majority of such studies. Consequently, significantly fewer coordinates are required to parametrize the low-dimensional latent space where robust and accurate ROMs can be constructed with reduced complexity. However, this advantage over POD comes with a challenge: the pre-image problem—mapping latent coordinates back to high-dimensional fields—is ill-posed and usually computationally more expensive. Crucially, mass conservation must be enforced consistently when lifting back to physical space by using an appropriate lifting scheme, such as the  $k$ -NN algorithm with convex interpolation [67], which, as we have proven

here, preserves mass positivity; as we also prove here, mass preservation is also achieved with POD lifting, even though positivity is not guaranteed. Future work will address systems with multiple basins of attraction— and comparing our manifold-DMs-informed ROMs against autoencoders, neural operators and other ML architectures to quantify robustness, generalization and conservation properties. Another direction is to handle cases where high-dimensional statistics are functions of low-dimensional variables (i.e., low-to-high probability distributions) [114], thus identifying the underlying SDEs, that preserve distributional constraints.

## A Numerical simulation of the Hughes model

High-dimensional crowd density data were generated by numerically solving the Hughes model (Eqs. (36a)–(36b)) with an in-house finite-volume scheme implemented in Python. The spatial domain is discretized into  $N = N_x \times N_y = 200 \times 50$  cells, with  $\rho_{i,j}^n$  denoting the cell-averaged density at  $(x_i, y_j)$  and time  $t^n$ . To capture different crowd regimes,  $N_{ic} = 100$  simulations with varying initial conditions are performed, each producing  $N_t = 700$  snapshots spaced by  $\Delta t = 0.1$  s. Snapshots are collected as column vectors  $\mathbf{x}_m \in \mathbb{R}^N$  and assembled in the snapshot matrix  $\mathbf{X} = \{\mathbf{x}_m\}_{m=1,\dots,M} \in \mathbb{R}^{N \times M}$ , with  $M = N_t \cdot N_{ic}$ .

The domain is a rectangular corridor  $\Omega$  with a central square obstacle  $\Omega_{\text{obs}}$ , length  $L = 20$  m, height  $H = 5$  m, and obstacle side  $L_{\text{obs}} = 1$  m (see Fig. 2(a) in Section 3.1). Initial conditions are 2D Gaussian density distributions:

$$\rho_0(x, y) = \Gamma_0 \exp \left[ -\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{(y - y_0)^2}{2\sigma_y^2} \right], \quad (40)$$

with the total mass,

$$m = \iint_{\Omega} \rho_0(x, y) dx dy, \quad (41)$$

kept constant across all initial conditions by adjusting  $\Gamma_0$ .

The Hughes PDE model (Eqs. (36a)–(36b)) is complemented with the following boundary conditions prescribed on the inlet ( $\Gamma_{\text{in}}$ ), outlet ( $\Gamma_{\text{out}}$ ), the domain walls ( $\Gamma_w$ ), and the obstacle boundary ( $\Gamma_{\text{obs}}$ ):

$$\begin{cases} \left( \rho f(\rho) \frac{\partial \phi / \partial \mathbf{n}}{\|\nabla \phi\|} \right) \cdot \mathbf{n}(x, y) = 0 & \text{on } (\Gamma_{\text{obs}} \cup \Gamma_w) \times [0, t_f], \\ \rho(0, y, t) = \rho(L, y, t) & y \in [0, H], t \in [0, t_f], \\ \phi = 0 & \text{on } \Gamma_{\text{out}} \times [0, t_f], \end{cases} \quad (42)$$

where  $\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_w \cup \Gamma_{\text{obs}}$ , and  $\mathbf{n}$  is the local outward unit normal. Homogeneous Neumann conditions enforce zero flux at the walls, while periodicity in  $x$  ensures inflow–outflow continuity. The potential is set to zero at the exit. Note that each simulation is run until approximately four loops of the domain are completed.

The numerical solution of the Hughes model (Eqs. (36a)–(36b)) is carried out using a finite-volume scheme combined with the Fast Sweeping Method [115] to efficiently solve the nonlinear Eikonal equation (36b). The approach outlined here specifies the spatial and temporal discretization of the governing equations, the evaluation of numerical fluxes via Godunov’s method, and the stability condition adopted for time-step selection.

At each time step, given the current density field  $\rho_{i,j}^n$ , the right-hand side of the Eikonal equation is evaluated, and the potential  $\phi_{i,j}^n$  is computed using the Fast Sweeping Method [115]. Once the potential is available, the system (36a)–(36b) reduces to a scalar nonlinear conservation law:

$$\frac{\partial \rho}{\partial t} + \frac{\partial F_x(\rho, \phi)}{\partial x} + \frac{\partial F_y(\rho, \phi)}{\partial y} = 0, \quad (43)$$

where  $F_x(\rho, \phi)$  and  $F_y(\rho, \phi)$  denote the density fluxes along the  $x$ - and  $y$ -directions:

$$F_x(\rho, \phi) = \rho U_x(\rho, \phi) = \rho g(\rho) v_f \left( 1 - \frac{\rho}{\rho_m} \right) \frac{\phi_x}{\sqrt{\phi_x^2 + \phi_y^2}}, \quad (44)$$

$$F_y(\rho, \phi) = \rho U_y(\rho, \phi) = \rho g(\rho) v_f \left( 1 - \frac{\rho}{\rho_m} \right) \frac{\phi_y}{\sqrt{\phi_x^2 + \phi_y^2}}, \quad (45)$$

with  $\phi_x = \partial\phi/\partial x$  and  $\phi_y = \partial\phi/\partial y$  evaluated by first-order forward finite differences of the discrete potential field.

The density field is then updated in conservative form:

$$\rho_{i,j}^{n+1} = \rho_{i,j}^n - \frac{\Delta t}{\Delta x} \left( \hat{F}_{i+\frac{1}{2},j}^n - \hat{F}_{i-\frac{1}{2},j}^n \right) - \frac{\Delta t}{\Delta y} \left( \hat{F}_{i,j+\frac{1}{2}}^n - \hat{F}_{i,j-\frac{1}{2}}^n \right), \quad (46)$$

where  $\Delta x = \Delta y$  denotes the uniform grid spacing.

The intercell fluxes are computed using Godunov's method. In the  $x$ -direction:

$$\hat{F}_{i+\frac{1}{2},j}^n = \begin{cases} \min_{\rho_{i,j}^n \leq \theta \leq \rho_{i+1,j}^n} F_x(\theta) & \text{if } \rho_{i,j}^n \leq \rho_{i+1,j}^n, \\ \max_{\rho_{i+1,j}^n \leq \theta \leq \rho_{i,j}^n} F_x(\theta) & \text{otherwise,} \end{cases} \quad (47)$$

$$\hat{F}_{i-\frac{1}{2},j}^n = \begin{cases} \min_{\rho_{i-1,j}^n \leq \theta \leq \rho_{i,j}^n} F_x(\theta) & \text{if } \rho_{i-1,j}^n \leq \rho_{i,j}^n, \\ \max_{\rho_{i,j}^n \leq \theta \leq \rho_{i-1,j}^n} F_x(\theta) & \text{otherwise,} \end{cases} \quad (48)$$

and analogously in the  $y$ -direction:

$$\hat{F}_{i,j+\frac{1}{2}}^n = \begin{cases} \min_{\rho_{i,j}^n \leq \theta \leq \rho_{i,j+1}^n} F_y(\theta) & \text{if } \rho_{i,j}^n \leq \rho_{i,j+1}^n, \\ \max_{\rho_{i,j+1}^n \leq \theta \leq \rho_{i,j}^n} F_y(\theta) & \text{otherwise,} \end{cases} \quad (49)$$

$$\hat{F}_{i,j-\frac{1}{2}}^n = \begin{cases} \min_{\rho_{i,j-1}^n \leq \theta \leq \rho_{i,j}^n} F_y(\theta) & \text{if } \rho_{i,j-1}^n \leq \rho_{i,j}^n, \\ \max_{\rho_{i,j}^n \leq \theta \leq \rho_{i,j-1}^n} F_y(\theta) & \text{otherwise.} \end{cases} \quad (50)$$

The fluxes  $F_x(\theta)$  and  $F_y(\theta)$  in Eqs. (46) are evaluated at frozen coefficients  $\phi_x, \phi_y$ . The optimization is carried out with respect to  $\theta \in [\rho_L, \rho_R]$ , where  $\rho_L$  and  $\rho_R$  are the left and right states at the interface.

Finally, the time step  $\Delta t$  is chosen according to a Courant–Friedrichs–Lewy (CFL) condition:

$$\Delta t = CFL \cdot \min \left( \frac{\Delta x}{\max |U_x|}, \frac{\Delta y}{\max |U_y|} \right), \quad (51)$$

where the maxima of  $|U_x|$  and  $|U_y|$  (see Eqs. (44)–(45)) are taken over all grid points. In all simulations, a Courant number of  $CFL = 0.25$  was adopted.

## B Crowd dynamics paradigm: optimal model selection

Model	$\varepsilon_2 (\times 10^{-3})$	$\varepsilon_2^r (\times 10^{-1})$	$W_1 (\times 10^{-6})$
POD-MVAR ( $d = 18, l = 5$ )	1.60 (1.00, 2.20)	1.07 (0.70, 1.42)	7.15 (3.89, 11.60)
POD-MVAR ( $d = 20, l = 5$ )	1.40 (1.00, 1.90)	0.97 (0.66, 1.12)	6.22 (3.74, 9.49)
POD-MVAR ( $d = 22, l = 5$ )	1.40 (0.91, 1.80)	0.94 (0.59, 1.26)	6.47 (3.82, 9.52)
POD-MVAR ( $d = 22, l = 1$ )	27.3 (24.4, 30.2)	18.7 (17.0, 20.6)	122.0 (106.8, 140.9)
POD-SINDy ( $d = 18, p(1) + f$ )	2.53 (1.78, 3.59)	1.69 (1.22, 2.34)	10.06 (7.78, 13.38)
POD-SINDy ( $d = 20, p(1) + f$ )	2.58 (1.81, 3.70)	1.71 (1.24, 2.41)	10.29 (7.75, 13.86)
POD-SINDy ( $d = 22, p(1) + f$ )	2.46 (1.60, 3.57)	1.64 (1.10, 2.34)	8.75 (6.31, 12.25)
DMs-MVAR ( $d = 7, l = 8$ )	1.30 (0.42, 2.30)	0.87 (0.31, 1.50)	2.24 (0.56, 4.32)
DMs-MVAR ( $d = 10, l = 8$ )	1.00 (0.37, 1.70)	0.67 (0.26, 1.22)	1.67 (0.50, 3.18)
DMs-SINDy ( $d = 7, pnc(2)$ )	1.44 (0.63, 2.48)	0.97 (0.43, 1.67)	4.63 (0.70, 11.84)
DMs-SINDy ( $d = 10, pnc(2)$ )	1.96 (0.78, 3.33)	1.32 (0.53, 2.22)	3.86 (1.02, 8.08)

Table 3: Crowd dynamics paradigm: summary of the manifold-informed ROMs performance on the training set  $X_{\text{train}}$ . Columns report absolute  $\ell_2$  error ( $\varepsilon_2$ ), relative  $\ell_2$  error ( $\varepsilon_2^r$ ), and Wasserstein–1 distance ( $W_1$ ) in terms of average over time and 10<sup>th</sup>–90<sup>th</sup> percentiles.

We detail here the procedure used to select the optimal reduced-order models for the crowd dynamics paradigm. In particular, we investigate the dependence of reconstruction and forecasting accuracy on the number of retained POD modes and DMs coordinates, and compare the performance of the MVAR and SINDy learning algorithms applied to these latent coordinates.

Model	$\varepsilon_2 (\times 10^{-3})$	$\varepsilon_2^r (\times 10^{-1})$	$W_1 (\times 10^{-6})$
POD-MVAR ( $d = 18, l = 5$ )	1.50 (0.95, 2.10)	1.01 (0.67, 1.32)	5.67 (3.03, 9.21)
POD-MVAR ( $d = 20, l = 5$ )	1.30 (0.91, 1.80)	0.91 (0.57, 1.18)	4.75 (2.70, 7.30)
POD-MVAR ( $d = 22, l = 5$ )	1.30 (0.79, 1.80)	0.88 (0.53, 1.17)	5.03 (2.82, 7.96)
POD-MVAR ( $d = 22, l = 1$ )	27.50 (26.0, 29.0)	19.0 (18.0, 20.0)	127.1 (118.0, 136.0)
POD-SINDy ( $d = 18, p(1) + f$ )	2.23 (1.85, 2.64)	1.52 (1.28, 1.79)	9.36 (8.21, 10.72)
POD-SINDy ( $d = 20, p(1) + f$ )	2.21 (1.84, 2.66)	1.50 (1.25, 1.79)	9.51 (8.26, 10.99)
POD-SINDy ( $d = 22, p(1) + f$ )	2.03 (1.60, 2.49)	1.40 (1.11, 1.70)	7.67 (6.48, 9.08)
DMS-MVAR ( $d = 7, l = 8$ )	1.20 (0.34, 2.30)	0.81 (0.24, 1.49)	2.18 (0.41, 4.79)
DMS-MVAR ( $d = 10, l = 8$ )	0.91 (0.32, 1.60)	0.60 (0.23, 1.02)	1.39 (0.36, 3.11)
DMS-SINDy ( $d = 7, pnc(2)$ )	1.14 (0.68, 1.70)	0.77 (0.46, 1.14)	2.66 (0.72, 6.51)
DMS-SINDy ( $d = 10, pnc(2)$ )	1.31 (0.54, 2.24)	0.88 (0.36, 1.51)	2.69 (0.90, 5.29)
DMS-MVAR ( $d = 7, l = 5$ )	1.21 (0.72, 1.73)	0.82 (0.50, 1.16)	1.93 (0.79, 3.22)

Table 4: Crowd dynamics paradigm: summary of manifold-informed ROMs performance on the validation set  $X_{\text{val}}$ . Columns report absolute  $\ell_2$  error ( $\varepsilon_2$ ), relative  $\ell_2$  error ( $\varepsilon_2^r$ ), and Wasserstein-1 distance ( $W_1$ ) in terms of average over time and 10<sup>th</sup>–90<sup>th</sup> percentiles. Rows shaded in green highlight the selected POD-informed and DMS-informed ROMs.

For the MVAR models, the lag order  $l$  was selected using the Bayesian Information Criterion (BIC). The BIC yields  $l = 5$  for POD-informed ROMs with  $d = 18, 20$ , and  $22$  modes, and  $l = 8$  for DMS-informed ROMs with  $d = 7$  and  $10$  coordinates. Tables 3 and 4 summarize performance on the training and validation datasets, respectively, in terms of absolute error ( $\varepsilon_2$ ), relative error ( $\varepsilon_2^r$ ), and Wasserstein-1 distance ( $W_1$ ). All metrics are averaged over time with 10<sup>th</sup>–90<sup>th</sup> percentiles. For the reconstruction of the DMS-informed ROMs predictions in the high-dimensional space, we have used the  $k$ -NN algorithm with  $k = 8$ .

Within the DMS-informed ROMs, reconstruction accuracy of the MVARs improves with the number of retained coordinates. In contrast, for the SINDy, more parsimonious models yield better accuracy, with the best performance achieved for  $d = 7$ . For the MVARs, although  $d = 7$  and  $d = 10$  exhibit comparable projection-stage accuracy, predictive errors are consistently lower for  $d = 10$  across all metrics and datasets.

For the POD-informed ROMs, both MVAR and SINDy models show increasing accuracy as the number of retained modes grows. The best performance is achieved with  $d = 20$  modes for the MVAR and  $d = 22$  modes for the SINDy-ROM, based on the  $W_1$  metric evaluated on both the training and validation datasets. Accordingly, we selected  $d = 10$  as the optimal DMS-informed MVAR ROM and  $d = 20$  as the optimal POD-informed MVAR ROM for test-set evaluation, and  $d = 4$  as the optimal DMS-informed SINDy ROM and  $d = 22$  as the optimal POD-informed SINDy ROM. Results obtained with these four selected models have been presented and discussed in Section 4.1.

## C Numerical simulation of the fluidic pinball

The Navier-Stokes equations (39a)–(39c) are solved by using a projection method: a provisional velocity field is first computed neglecting the pressure gradient, and subsequently projected onto the space of divergence-free fields by solving a pressure Poisson equation. Once the velocity field is obtained, it is used to advect the tracer concentration by solving Eq. (39d) with the second-order Bell–Colella–Glaz (BCG) scheme. Details on the numerical implementation in the open-source code BASILISK can be found in Popinet [116]. All computations are performed on a uniform Cartesian grid with spacing  $\Delta x = \Delta y = 0.2D$ , corresponding to  $N = N_x \times N_y = 15124$  grid cells. As shown in [117], this resolution is sufficient to ensure grid-independent results in the periodic flow regime of the fluidic pinball.

The computational domain is a rectangle of dimensions  $L_x = 52D$  and  $L_y = 12D$  ( $D = 1$  m), from which the interior of the cylinders is excluded (see Fig. 2(b) in Section 3.2). A Cartesian coordinate system  $\mathcal{O}xy$  is adopted, with the origin located at the midpoint between the two rear cylinders. Since the cylinders are fixed in the present study, no-slip boundary conditions are enforced on their surfaces. At the inlet (left boundary), a uniform velocity profile is prescribed ( $u = U_\infty, v = 0$ ), while a standard free-outflow condition is imposed at the outlet (right boundary). Homogeneous Neumann boundary conditions are applied on the remaining boundaries.

The initial velocity field is prescribed as

$$u(t = 0) = U_\infty + u'_0, \quad v(t = 0) = v'_0, \quad (52)$$

where  $u'_0$  and  $v'_0$  are random perturbations modeled as white noise with amplitude equal to 10% of the free-stream velocity ( $U_\infty = 1$  m/s). The tracer concentration is initially set to zero throughout the domain.

After the spatio-temporal periodic oscillations are fully established, the tracer is initialized as a two-dimensional Gaussian distribution,

$$C_0(x, y) = \bar{C}_0 \exp \left[ -\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{(y - y_0)^2}{2\sigma_y^2} \right], \quad (53)$$

as in the previous case study. The total initial tracer mass, defined as the spatial integral of  $C_0(x, y)$  over the domain (see Eq. (41) in A), is kept constant across all simulations by appropriately adjusting the amplitude  $\bar{C}_0$ .

To construct the dataset,  $N_{ic} = 80$  simulations are performed with different initial tracer distributions, obtained by varying the parameters within the ranges  $x_0 \in [14, 16]$  m,  $y_0 \in [-1, 1]$  m, and  $\sigma_x, \sigma_y \in [4, 5]$  m. Since the passive tracer is the mass-preserving quantity of interest in this case study, each simulation is advanced up to  $t_f = 24$  s, beyond which the tracer leaves the domain and mass conservation is no longer satisfied. For each initial condition,  $N_t = 120$  snapshots of the tracer field are stored at uniform time intervals  $\Delta t = 0.2$  s. The snapshots are reshaped into column vectors  $\mathbf{x}_m \in \mathbb{R}^N$  and assembled into the snapshot matrix  $\mathbf{X} = \{\mathbf{x}_m\}_{m=1, \dots, M} \in \mathbb{R}^{N \times M}$ , with  $M = N_t \cdot N_{ic}$ .

## D Fluid dynamics paradigm: optimal model selection

Model	$\varepsilon_2 (\times 10^{-3})$	$\varepsilon_2^r (\times 10^{-1})$	$W_1 (\times 10^{-6})$
POD-MVAR ( $d = 16, l = 3$ )	2.16 (1.66, 2.80)	1.24 (0.99, 1.56)	6.16 (3.60, 9.41)
POD-MVAR ( $d = 18, l = 2$ )	3.13 (2.64, 3.84)	1.80 (1.57, 2.18)	12.36 (10.07, 15.11)
POD-MVAR ( $d = 20, l = 2$ )	3.82 (3.41, 4.39)	2.20 (2.01, 2.48)	17.11 (14.78, 19.70)
POD-SINDy ( $d = 16, p(1) + f$ )	2.96 (2.41, 3.71)	1.71 (1.44, 2.10)	7.52 (4.65, 11.10)
POD-SINDy ( $d = 18, p(1) + f$ )	2.76 (2.21, 3.57)	1.59 (1.31, 2.01)	7.30 (4.68, 10.37)
POD-SINDy ( $d = 20, p(1) + f$ )	2.69 (2.23, 3.39)	1.55 (1.32, 1.91)	6.85 (4.53, 9.55)
DMs-MVAR ( $d = 4, l = 5$ )	2.33 (1.05, 3.78)	1.34 (0.62, 2.16)	2.50 (0.78, 4.55)
DMs-MVAR ( $d = 6, l = 3$ )	1.94 (0.89, 3.11)	1.12 (0.52, 1.77)	2.14 (0.90, 3.61)
DMs-SINDy ( $d = 4, p(3)$ )	1.97 (0.68, 3.64)	1.14 (0.40, 2.11)	1.82 (0.67, 3.01)
DMs-SINDy ( $d = 6, p(1) + f$ )	2.57 (1.64, 3.78)	1.48 (0.95, 2.15)	4.63 (1.74, 7.24)

Table 5: Fluid dynamics paradigm: summary of manifold-informed ROMs performance on the training set  $X_{\text{train}}$ . Columns report absolute  $\ell_2$  error ( $\varepsilon_2$ ), relative  $\ell_2$  error ( $\varepsilon_2^r$ ), and Wasserstein-1 distance ( $W_1$ ) in terms of average over time and 10<sup>th</sup>-90<sup>th</sup> percentiles. Rows shaded in green highlight the selected POD-informed and DMs-informed ROMs.

Model	$\varepsilon_2 (\times 10^{-3})$	$\varepsilon_2^r (\times 10^{-1})$	$W_1 (\times 10^{-6})$
POD-MVAR ( $d = 16, l = 3$ )	3.56 (2.66, 4.49)	2.05 (1.54, 2.55)	8.55 (4.26, 15.01)
POD-MVAR ( $d = 18, l = 2$ )	4.28 (3.25, 5.16)	2.48 (1.87, 3.05)	14.65 (11.08, 20.26)
POD-MVAR ( $d = 20, l = 2$ )	4.89 (3.89, 5.79)	2.84 (2.25, 3.41)	19.14 (15.48, 24.48)
POD-SINDy ( $d = 16, p(1) + f$ )	4.03 (3.19, 4.99)	2.33 (1.84, 2.82)	9.21 (4.58, 17.36)
POD-SINDy ( $d = 18, p(1) + f$ )	3.94 (3.06, 4.98)	2.28 (1.76, 2.82)	9.24 (4.57, 17.21)
POD-SINDy ( $d = 20, p(1) + f$ )	3.59 (2.59, 5.16)	2.07 (1.51, 2.96)	7.89 (3.75, 14.13)
DMs-MVAR ( $d = 4, l = 5$ )	2.69 (1.40, 4.10)	1.56 (0.82, 2.35)	3.02 (0.92, 5.70)
DMs-MVAR ( $d = 6, l = 3$ )	2.21 (1.24, 3.22)	1.28 (0.72, 1.85)	2.48 (0.97, 4.35)
DMs-SINDy ( $d = 4, p(3)$ )	2.43 (1.13, 4.17)	1.41 (0.66, 2.40)	2.05 (0.90, 3.31)
DMs-SINDy ( $d = 6, p(1) + f$ )	2.69 (1.79, 3.71)	1.56 (1.03, 2.13)	4.72 (1.43, 7.36)

Table 6: Fluid dynamics paradigm: summary of manifold-informed ROMs performance on the validation set  $X_{\text{val}}$ . Columns report absolute  $\ell_2$  error ( $\varepsilon_2$ ), relative  $\ell_2$  error ( $\varepsilon_2^r$ ), and Wasserstein-1 distance ( $W_1$ ) in terms of average over time and 10<sup>th</sup>-90<sup>th</sup> percentiles. Rows shaded in green highlight the selected POD-informed and DMs-informed ROMs.

We detail here the procedure used to select the optimal reduced-order models (ROMs) for the fluid dynamics paradigm. In particular, we investigate the dependence of reconstruction and forecasting accuracy on the number of retained POD modes and DMs coordinates, and compare the performance of the MVAR and SINDy learning algorithms applied to these latent coordinates.

For the MVAR models, the lag order  $l$  was selected using the Bayesian Information Criterion (BIC). For the POD-informed models, the BIC selects  $l = 3$  for  $d = 16$  modes, while  $l = 2$  for  $d = 18$  and  $d = 20$ . For the DMs-informed models, the BIC selects  $l = 5$  for the lower-dimensional embedding ( $d = 4$ ), and  $l = 3$  for  $d = 6$ . Tables 5 and 6 summarize ROM performance on the training and validation datasets, respectively, in terms of absolute error ( $\varepsilon_2$ ), relative error ( $\varepsilon_2^r$ ), and Wasserstein-1 distance ( $W_1$ ), reported as time-averaged values with 10<sup>th</sup>-90<sup>th</sup> percentiles. For

the reconstruction of the DMs-informed ROMs predictions in the high-dimensional space, we have used the  $k$ -NN algorithm with  $k = 6$ .

Within the DMs-informed ROMs, the MVAR reconstruction accuracy improves with the number of retained coordinates, with the best validation performance achieved for  $d = 6$  and  $l = 3$  (Table 6). For DMs-informed SINDy, more parsimonious models yield better performance: the cubic polynomial library  $p(3)$  with  $d = 4$  coordinates attains the lowest  $W_1$  on the validation set, while increasing the latent dimension to  $d = 6$  (with a  $p(1) + f$  library) degrades accuracy.

For the POD-informed ROMs, MVARs exhibit increasing errors as  $d$  grows from 16 to 20, indicating that retaining additional POD modes does not translate into improved latent-space predictability for the linear autoregressive model in this setting. Conversely, POD-informed SINDy benefits from richer POD embeddings: increasing the number of modes reduces the validation error, with the best performance obtained at  $d = 20$  using a first-order polynomial library augmented with Fourier features ( $p(1) + f$ ). Based primarily on the  $W_1$  metric evaluated on both the training and validation datasets, we selected  $d = 6$  as the optimal DMs-informed MVAR ROM and  $d = 16$  as the optimal POD-informed MVAR ROM, and  $d = 4$  as the optimal DMs-informed SINDy ROM and  $d = 20$  as the optimal POD-informed SINDy ROM. Results obtained with these four selected models have been presented and discussed in Section 4.2.

## References

- [1] F. P. Kemeth, S. W. Haugland, F. Dietrich, T. Bertalan, K. Höhle, Q. Li, E. M. Bollt, R. Talmon, K. Krischer, and I. G. Kevrekidis, “An emergent space for distributed data with hidden internal order through manifold learning,” *IEEE Access*, vol. 6, pp. 77402–77413, 2018.
- [2] P. A. Reinbold and R. O. Grigoriev, “Data-driven discovery of partial differential equation models with latent variables,” *Physical Review E*, vol. 100, no. 2, p. 022219, 2019.
- [3] F. P. Kemeth, T. Bertalan, T. Thiem, F. Dietrich, S. J. Moon, C. R. Laing, and I. G. Kevrekidis, “Learning emergent partial differential equations in a learned emergent space,” *Nature communications*, vol. 13, no. 1, p. 3318, 2022.
- [4] D. W. Sroczynski, F. Dietrich, E. D. Koronaki, R. Talmon, R. R. Coifman, E. Bollt, and I. G. Kevrekidis, “On learning what to learn: Heterogeneous observations of dynamics and establishing possibly causal relations among them,” *PNAS nexus*, vol. 3, no. 12, p. pgae494, 2024.
- [5] L. Russo, C. I. Siettos, and I. G. Kevrekidis, “Reduced computations for nematic-liquid crystals: A timestepper approach for systems with continuous symmetries,” *Journal of non-newtonian fluid mechanics*, vol. 146, no. 1-3, pp. 51–58, 2007.
- [6] J. N. Kutz, “Deep learning in fluid dynamics,” *Journal of Fluid Mechanics*, vol. 814, pp. 1–4, 2017.
- [7] M. Raissi, A. Yazdani, and G. E. Karniadakis, “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations,” *Science*, vol. 367, no. 6481, pp. 1026–1030, 2020.
- [8] A. S. Joglekar and A. G. Thomas, “Machine learning of hidden variables in multiscale fluid simulation,” *Machine Learning: Science and Technology*, vol. 4, no. 3, p. 035049, 2023.
- [9] N. Evangelou, N. J. Wichrowski, G. A. Kevrekidis, F. Dietrich, M. Kooshkbaghi, S. McFann, and I. G. Kevrekidis, “On the parameter combinations that matter and on those that do not: data-driven studies of parameter (non) identifiability,” *PNAS nexus*, vol. 1, no. 4, p. pgac154, 2022.
- [10] C. P. Martin-Linares, Y. M. Psarellis, G. Karapetsas, E. D. Koronaki, and I. G. Kevrekidis, “Physics-agnostic and physics-infused machine learning for thin films flows: modelling, and predictions from small data,” *Journal of Fluid Mechanics*, vol. 975, no. A41, pp. 1–22, 2023.
- [11] B. Engelhardt, M. Kschischo, and H. Fröhlich, “A bayesian approach to estimating hidden variables as well as missing and wrong molecular interactions in ordinary differential equation-based mathematical models,” *Journal of The Royal Society Interface*, vol. 14, no. 131, p. 20170332, 2017.
- [12] A. Yazdani, L. Lu, M. Raissi, and G. E. Karniadakis, “Systems biology informed deep learning for inferring parameters and hidden dynamics,” *PLoS computational biology*, vol. 16, no. 11, p. e1007575, 2020.
- [13] S. Lee, Y. M. Psarellis, C. I. Siettos, and I. G. Kevrekidis, “Learning black-and gray-box chemotactic pdes/closures from agent based monte carlo simulation data,” *Journal of Mathematical Biology*, vol. 87, no. 1, p. 15, 2023.
- [14] N. Ahmadi Daryakenari, M. De Florio, K. Shukla, and G. E. Karniadakis, “Ai-aristotle: A physics-informed framework for systems biology gray-box identification,” *PLOS Computational Biology*, vol. 20, no. 3, p. e1011916, 2024.

- [15] N. Bellomo and C. Dogbe, “On the modeling of traffic and crowds: A survey of models, speculations, and perspectives,” *SIAM review*, vol. 53, no. 3, pp. 409–463, 2011.
- [16] N. Bellomo, B. Piccoli, and A. Tosin, “Modeling crowd dynamics from a complex system viewpoint,” *Mathematical models and methods in applied sciences*, vol. 22, no. supp02, p. 1230004, 2012.
- [17] E. Cristiani, B. Piccoli, and A. Tosin, *Multiscale Modeling of Pedestrian Dynamics*, vol. 12 of *MS&A: Modeling, Simulation and Applications*. Cham: Springer, 2014.
- [18] E. Cristiani, B. Piccoli, and A. Tosin, “An overview of the modeling of crowd dynamics,” *Multiscale modeling of pedestrian dynamics*, pp. 73–107, 2014.
- [19] N. Bellomo, L. Gibelli, A. Quaini, and A. Reali, “Towards a mathematical theory of behavioral human crowds,” *Mathematical Models and Methods in Applied Sciences*, vol. 32, pp. 321–358, 2022.
- [20] N. Bellomo, J. Liao, A. Quaini, L. Russo, and C. Siettos, “Human behavioral crowds review, critical analysis and research perspectives,” *Mathematical Models and Methods in Applied Sciences*, vol. 33, pp. 1–49, Jun 2023.
- [21] S. Lee, M. Kooshkbaghi, K. Spiliotis, C. I. Siettos, and I. G. Kevrekidis, “Coarse-scale pdes from fine-scale observations via machine learning,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 1, p. 013141, 2020.
- [22] E. Galaris, G. Fabiani, I. Gallos, I. Kevrekidis, and C. Siettos, “Numerical bifurcation analysis of pdes from lattice boltzmann model simulations: a parsimonious machine learning approach,” *Journal of Scientific Computing*, vol. 92, no. 34, pp. 1–30, 2022.
- [23] G. Fabiani, N. Evangelou, T. Cui, J. M. Bello-Rivas, C. P. Martin-Linares, C. Siettos, and I. G. Kevrekidis, “Task-oriented machine learning surrogates for tipping points of agent-based models,” *Nature communications*, vol. 15, no. 1, p. 4117, 2024.
- [24] M. De Florio, I. G. Kevrekidis, and G. E. Karniadakis, “Ai-lorenz: A physics-data-driven framework for black-box and gray-box identification of chaotic systems with symbolic regression,” *Chaos, Solitons & Fractals*, vol. 188, p. 115538, 2024.
- [25] J. Qu, W. Cai, and Y. Zhao, “Learning time-dependent pdes with a linear and nonlinear separate convolutional neural network,” *Journal of Computational Physics*, vol. 453, p. 110928, 2022.
- [26] P. R. Vlachas, G. Arampatzis, C. Uhler, and P. Koumoutsakos, “Multiscale simulations of complex systems by learning their effective dynamics,” *Nature Machine Intelligence*, vol. 4, no. 4, pp. 359–366, 2022.
- [27] G. Pang, M. D’Elia, M. Parks, and G. Karniadakis, “npinns: Nonlocal physics-informed neural networks for a parametrized nonlocal universal laplacian operator. algorithms and applications,” *Journal of Computational Physics*, vol. 422, p. 109760, 2020.
- [28] Y. Chen, B. Hosseini, H. Owhadi, and A. M. Stuart, “Solving and learning nonlinear pdes with gaussian processes,” *Journal of Computational Physics*, vol. 447, p. 110668, 2021.
- [29] X. Li, W. Wang, Q. Liu, *et al.*, “Deep neural networks for simulating complex dynamical systems: A framework combining neural operators and recurrent neural networks,” *arXiv preprint arXiv:2307.07331*, 2023.
- [30] S. Goswami, A. Bora, Y. Yu, and G. E. Karniadakis, “Physics-informed deep neural operator networks,” in *Machine learning in modeling and simulation: methods and applications*, pp. 219–254, Springer, 2023.
- [31] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, “Fourier neural operator for parametric partial differential equations,” *arXiv preprint arXiv:2010.08895*, 2020.
- [32] A. Peyvan, V. Oommen, A. D. Jagtap, and G. E. Karniadakis, “Riemannets: Interpretable neural operators for riemann problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 426, p. 116996, 2024.
- [33] E. Zappala, A. H. d. O. Fonseca, J. O. Caro, A. H. Moberly, M. J. Higley, J. Cardin, and D. v. Dijk, “Learning integral operators via neural integral equations,” *Nature Machine Intelligence*, vol. 6, no. 9, pp. 1046–1062, 2024.
- [34] G. Fabiani, I. G. Kevrekidis, C. Siettos, and A. N. Yannacopoulos, “Randonets: Shallow networks with random projections for learning linear and nonlinear operators,” *Journal of Computational Physics*, vol. 520, p. 113433, 2025.
- [35] W. H. Schilders, H. A. Van der Vorst, and J. Rommes, *Model order reduction: theory, research aspects and applications*, vol. 13. Springer, 2008.
- [36] A. Quarteroni and G. Rozza, *Reduced order methods for modeling and computational reduction*, vol. 9. Springer, 2014.
- [37] P. G. Papaioannou, R. Talmon, I. G. Kevrekidis, and C. Siettos, “Time-series forecasting using manifold learning, radial basis function interpolation, and geometric harmonics,” *Chaos*, vol. 32, no. 8, p. 083113, 2022.

- [38] N. Evangelou, F. Dietrich, E. Chiavazzo, D. Lehmberg, M. Meila, and I. G. Kevrekidis, “Double diffusion maps and their latent harmonics for scientific computations in latent space,” *Journal of Computational Physics*, vol. 485, no. 112072, 2023.
- [39] K. Kontolati, S. Goswami, G. Em Karniadakis, and M. D. Shields, “Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems,” *Nature Communications*, vol. 15, no. 1, p. 5101, 2024.
- [40] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps,” *Proc. Natl. Acad. Sci.*, vol. 102, no. 21, pp. 7426–7431, 2005.
- [41] R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.
- [42] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos, “Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis,” *Communications in Mathematical Sciences*, vol. 1, no. 4, pp. 715–762, 2003.
- [43] H. Arbabi and I. G. Kevrekidis, “Particles to partial differential equations parsimoniously,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 3, 2021.
- [44] D. Patsatzis, L. Russo, I. G. Kevrekidis, and C. Siettos, “Data-driven control of agent-based models: An equation/variable-free machine learning approach,” *Journal of Computational Physics*, vol. 478, no. 111953, 2023.
- [45] E. D. Koronaki, N. Evangelou, C. P. Martin-Linares, E. S. Titi, and I. G. Kevrekidis, “Nonlinear dimensionality reduction then and now: Aims for dissipative pdes in the ml era,” *Journal of Computational Physics*, vol. 506, p. 112910, 2024.
- [46] H. V. Alvarez, D. G. Patsatzis, L. Russo, I. Kevrekidis, and C. Siettos, “Next generation equation-free multiscale modelling of crowd dynamics via machine learning,” *arXiv preprint arXiv:2508.03926*, 2025.
- [47] T. Sauer, J. A. Yorke, and M. Casdagli, “Embedology,” *Journal of statistical Physics*, vol. 65, no. 3, pp. 579–616, 1991.
- [48] R. Rico-Martinez, K. Krischer, I. Kevrekidis, M. Kube, and J. Hudson, “Discrete-vs. continuous-time nonlinear signal processing of cu electrodisolution data,” *Chemical Engineering Communications*, vol. 118, no. 1, pp. 25–48, 1992.
- [49] K. Krischer, R. Rico-Martinez, I. Kevrekidis, H. Rotermund, G. Ertl, and J. Hudson, “Model identification of a spatiotemporally varying catalytic reaction,” *AIChE Journal*, vol. 39, no. 1, pp. 89–98, 1993.
- [50] S. Y. Shvartsman and I. Kevrekidis, “Low-dimensional approximation and control of periodic solutions in spatially extended systems,” *Physical Review E*, vol. 58, no. 1, p. 361, 1998.
- [51] S. Y. Shvartsman, C. Theodoropoulos, R. Rico-Martinez, I. Kevrekidis, E. S. Titi, and T. Mountziaris, “Order reduction for nonlinear dynamic models of distributed reacting systems,” *Journal of Process Control*, vol. 10, no. 2-3, pp. 177–184, 2000.
- [52] D. Dylewsky, E. Kaiser, S. L. Brunton, and J. N. Kutz, “Principal component trajectories for modeling spectrally continuous dynamics as forced linear systems,” *Physical Review E*, vol. 105, no. 1, p. 015312, 2022.
- [53] J. Axås and G. Haller, “Model reduction for nonlinearizable dynamics via delay-embedded spectral submanifolds,” *Nonlinear Dynamics*, vol. 111, no. 24, pp. 22079–22099, 2023.
- [54] I. K. Gallos, D. Lehmberg, F. Dietrich, and C. Siettos, “Data-driven modelling of brain activity using neural networks, diffusion maps, and the koopman operator,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 34, no. 1, 2024.
- [55] P. M. Patil, E. Kaiser, J. Nathan Kutz, and S. L. Brunton, “Separation of periodic orbits in the delay-embedded space of chaotic attractors,” *Proceedings of the Royal Society A*, vol. 481, no. 2322, p. 20240900, 2025.
- [56] A. E. Deane, I. G. Kevrekidis, G. E. Karniadakis, and S. A. Orszag, “Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders,” *Physics of Fluids A: Fluid Dynamics*, vol. 3, no. 10, pp. 2337–2354, 1991.
- [57] R. Temam, *Navier–Stokes equations and nonlinear functional analysis*. SIAM, 1995.
- [58] M. D. Graham and I. G. Kevrekidis, “Alternative approaches to the karhunen-loeve decomposition for model reduction and data analysis,” *Computers & chemical engineering*, vol. 20, no. 5, pp. 495–506, 1996.

- [59] E. S. Titi, “On approximate inertial manifolds to the navier-stokes equations,” *Journal of mathematical analysis and applications*, vol. 149, no. 2, pp. 540–557, 1990.
- [60] P. Constantin, C. Foias, B. Nicolaenko, and R. Temam, *Integral manifolds and inertial manifolds for dissipative partial differential equations*, vol. 70. Springer Science & Business Media, 2012.
- [61] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [62] M. Belkin and P. Niyogi, “Towards a theoretical foundation for laplacian-based manifold methods,” *Journal of Computer and System Sciences*, vol. 74, no. 8, pp. 1289–1308, 2008.
- [63] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler, “Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems,” *Multiscale Model. Simul.*, vol. 7, no. 2, pp. 842–864, 2008.
- [64] C. J. Dsilva, R. Talmon, R. R. Coifman, and I. G. Kevrekidis, “Parsimonious representation of nonlinear dynamical systems through manifold learning: a chemotaxis case study,” *Appl. Comput. Harmon. Anal.*, vol. 44, no. 3, pp. 759–773, 2018.
- [65] C. Soize and R. Ghanem, “Probabilistic learning on manifolds (plom) with partition,” *International Journal for Numerical Methods in Engineering*, vol. 123, no. 1, pp. 268–290, 2022.
- [66] C. Soize and R. Ghanem, “Transient anisotropic kernel for probabilistic learning on manifolds,” *Computer Methods in Applied Mechanics and Engineering*, vol. 432, p. 117453, 2024.
- [67] T. Chin, J. Ruth, C. Sanford, R. Santorella, P. Carter, and B. Sandstede, “Enabling equation-free modeling via diffusion maps,” *Journal of Dynamics and Differential Equations*, vol. 36, no. Suppl 1, pp. 415–434, 2024.
- [68] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [69] E. Bollt, “Attractor modeling and empirical nonlinear model reduction of dissipative dynamical systems,” *International Journal of Bifurcation and Chaos*, vol. 17, no. 04, pp. 1199–1219, 2007.
- [70] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.
- [71] R. R. Coifman and S. Lafon, “Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions,” *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 31–52, 2006.
- [72] E. Chiavazzo, C. W. Gear, C. J. Dsilva, N. Rabin, and I. G. Kevrekidis, “Reduced models in chemical kinetics via nonlinear data-mining,” *Processes*, vol. 2, no. 1, pp. 112–140, 2014.
- [73] A. Charalampopoulos and T. Sapsis, “Uncertainty quantification of turbulent systems via physically consistent and data-informed reduced-order models,” *Physics of Fluids*, vol. 34, no. 7, 2022.
- [74] D. Floryan and M. D. Graham, “Data-driven discovery of intrinsic dynamics,” *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1113–1120, 2022.
- [75] P. Conti, G. Gobat, S. Fresca, A. Manzoni, and A. Frangi, “Reduced order modeling of parametrized systems through autoencoders and sindy approach: continuation of periodic solutions,” *Computer Methods in Applied Mechanics and Engineering*, vol. 411, p. 116072, 2023.
- [76] C. R. Constante-Amores and M. D. Graham, “Data-driven state-space and koopman operator models of coherent state dynamics on invariant manifolds,” *Journal of Fluid Mechanics*, vol. 984, p. R9, 2024.
- [77] F. Faraji, M. Reza, and J. N. Kutz, “Shallow recurrent decoder for reduced order modeling of  $e \times b$  plasma dynamics,” *Machine Learning: Science and Technology*, vol. 6, no. 2, p. 025024, 2025.
- [78] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [79] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Data-driven discovery of partial differential equations,” *Science Advances*, vol. 3, no. 4, 2017.
- [80] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox, “Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 372, p. 113433, 2020.
- [81] B. Kramer, B. Peherstorfer, and K. E. Willcox, “Learning nonlinear reduced models from data with operator inference,” *Annual Review of Fluid Mechanics*, vol. 56, pp. 521–548, 2024.

- [82] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, “Diffusion maps, spectral clustering and reaction coordinates of dynamical systems,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 113–127, 2006.
- [83] F. Romor, G. Stabile, and G. Rozza, “Non-linear manifold reduced-order models with convolutional autoencoders and reduced over-collocation method,” *Journal of Scientific Computing*, vol. 94, no. 3, p. 74, 2023.
- [84] L. Sirovich, “Turbulence and the dynamics of coherent structures, parts i–iii,” *Quarterly of Applied Mathematics*, vol. 45, no. 3, p. 561–571, 1987.
- [85] N. Aubry, W.-Y. Lian, and E. S. Titi, “Preserving symmetries in the proper orthogonal decomposition,” *SIAM Journal on Scientific Computing*, vol. 14, no. 2, pp. 483–505, 1993.
- [86] A. Holiday, M. Kooshkbaghi, J. M. Bello-Rivas, C. W. Gear, A. Zagaris, and I. G. Kevrekidis, “Manifold learning for parameter reduction,” *J. Comput. Phys.*, vol. 392, no. 1, pp. 419–431, 2019.
- [87] A. Singer, R. Erban, I. G. Kevrekidis, and R. R. Coifman, “Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 38, pp. 16090–16095, 2009.
- [88] I. K. Gallos, E. Galaris, and C. I. Siettos, “Construction of embedded fmri resting-state functional connectivity networks using manifold learning,” *Cognitive neurodynamics*, vol. 15, no. 4, pp. 585–608, 2021.
- [89] E. J. Nyström, *Über die praktische Auflösung von linearen Integralgleichungen mit Anwendungen auf Randwertaufgaben der Potentialtheorie*. Akademische Buchhandlung, 1929.
- [90] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*. Springer, 1st ed., 2005.
- [91] D. A. Dickey and W. A. Fuller, “Distribution of the estimators for autoregressive time series with a unit root,” *Journal of the American Statistical Association*, vol. 74, no. 366, pp. 427–431, 1979.
- [92] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, “Data-driven discovery of coordinates and governing equations,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22445–22451, 2019.
- [93] C. Villani, *Optimal Transport: Old and New*, vol. 338 of *Grundlehren der mathematischen Wissenschaften*. Berlin, Germany: Springer-Verlag Berlin Heidelberg, 2009.
- [94] B. Piccoli and F. Rossi, “Generalized wasserstein distance and its application to transport equations with source,” *Archive for Rational Mechanics and Analysis*, vol. 211, no. 1, pp. 335–358, 2014.
- [95] E. Cristiani, B. Piccoli, and A. Tosin, “Evolution in measure spaces with wasserstein distance,” in *Multiscale Modeling of Pedestrian Dynamics*, pp. 169–194, Springer, 2014.
- [96] B. Piccoli and F. Rossi, “On properties of the generalized wasserstein distance,” *Archive for Rational Mechanics and Analysis*, vol. 222, no. 3, pp. 1339–1365, 2016.
- [97] N. Kolbe, “Wasserstein distance,” <https://github.com/nklb/wasserstein-distance>, *GitHub*, 2025.
- [98] R. L. Hughes, “A continuum theory for the flow of pedestrians,” *Transportation Research Part B: Methodological*, vol. 36, pp. 507–535, Jul 2002.
- [99] P. Goatin and M. Mimault, “The wave-front tracking algorithm for hughes’ model of pedestrian motion,” *SIAM Journal on Scientific Computing*, vol. 35, no. 3, pp. B606–B622, 2013.
- [100] D. Amadori, P. Goatin, and M. D. Rosini, “Existence results for hughes’ model for pedestrian flows,” *Journal of Mathematical Analysis and applications*, vol. 420, no. 1, pp. 387–406, 2014.
- [101] D. Amadori, B. Andreianov, M. Di Francesco, S. Fagioli, T. Girard, P. Goatin, P. Markowich, J.-F. Pietschmann, M. D. Rosini, G. Russo, *et al.*, “The mathematical theory of hughes’ model: a survey of results,” in *Crowd Dynamics, Volume 4: Analytics and Human Factors in Crowd Modeling*, pp. 9–53, Springer, 2023.
- [102] V. Coscia and C. Canavesio, “First-order macroscopic modelling of human crowd dynamics,” *Math. Models Methods Appl. Sci.*, vol. 18, pp. 1217–1247, 2008.
- [103] N. Bellomo and C. Dogbé, “On the modelling crowd dynamics from scaling to hyperbolic macroscopic models,” *Mathematical Models and Methods in Applied Sciences*, vol. 18, pp. 1317–1345, August 2008.
- [104] N. Bellomo and L. Gibelli, “Behavioral human crowds and society,” in *Crowd Dynamics, Volume 4: Analytics and Human Factors in Crowd Modeling*, pp. 1–8, Springer, 2023.
- [105] B. D. Greenshields, “A study of traffic capacity,” *Proceedings of the Highway Research Board*, vol. 14, pp. 448–477, 1934.

- [106] M. J. Lighthill and G. B. Whitham, “On kinematic waves. ii. a theory of traffic flow on long crowded roads,” *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.
- [107] N. Deng, B. R. Noack, M. Morzynski, and L. R. Pastur, “Low-order model for successive bifurcations of the fluidic pinball,” *Journal of Fluid Mechanics*, vol. 884, no. A37, pp. 1–41, 2020.
- [108] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, “Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 365, p. 113028, 2020.
- [109] A. J. Huang and S. Agarwal, “On the limitations of physics-informed deep learning: Illustrations using first-order hyperbolic conservation law-based traffic flow models,” *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 279–293, 2023.
- [110] Z. Chen, A. Gelb, and Y. Lee, “Learning the dynamics for unknown hyperbolic conservation laws using deep neural networks,” *SIAM Journal on Scientific Computing*, vol. 46, no. 2, pp. A825–A850, 2024.
- [111] D. G. Patsatzis, M. di Bernardo, L. Russo, and C. Siettos, “Gorinns: Godunov-riemann informed neural networks for learning hyperbolic conservation laws,” *Journal of Computational Physics*, vol. 534, p. 114002, 2025.
- [112] A. Della Pia, D. G. Patsatzis, L. Russo, and C. Siettos, “Learning the latent dynamics of fluid flows from high-fidelity numerical simulations using parsimonious diffusion maps,” *Physics of Fluids*, vol. 36, no. 10, 2024.
- [113] A. Della Pia, D. G. Patsatzis, G. Rozza, L. Russo, and C. Siettos, “Surrogate normal-forms for the numerical bifurcation and stability analysis of navier-stokes flows via machine learning,” *arXiv preprint arXiv:2506.21275*, 2025.
- [114] D. G. Giovanis, E. Crabtree, R. G. Ghanem, and I. G. Kevrekidis, “Generative learning of densities on manifolds,” *arXiv preprint arXiv:2503.03963*, 2025.
- [115] H. Zhao, “A fast sweeping method for eikonal equations,” *Mathematics of Computation*, vol. 74, no. 250, pp. 603–627, 2004.
- [116] S. Popinet, “Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries,” *Journal of Computational Physics*, vol. 190, no. 2, pp. 572–600, 2003.
- [117] A. Della Pia, D. G. Patsatzis, L. Russo, and C. Siettos, “Learning the latent dynamics of fluid flows from high-fidelity numerical simulations using parsimonious diffusion maps,” *Physics of Fluids*, vol. 36, no. 105187, pp. 1–19, 2024.