arXiv:2501.12816v2 [math.NA] 16 Feb 2026

NONLINEAR REDUCTION STRATEGIES FOR DATA COMPRESSION: A COMPREHENSIVE COMPARISON FROM DIFFUSION TO ADVECTION PROBLEMS

ISABELLA CARLA GONNELLA[1], FEDERICO PICHI[1], GIANLUIGI ROZZA[1]

Abstract. This work presents an overview of several nonlinear reduction strategies for data compression from various research fields, and a comparison of their performance when applied to problems characterized by diffusion and/or advection terms. We aim to create a common framework by unifying the notation referring to a common two-stage pipeline. At the same time, we underline their main differences and objectives by highlighting the diverse choices made for each stage. We test the considered approaches on three test cases belonging to the family of Advection-Diffusion problems, also focusing on the pure Advection and pure Diffusion benchmarks, studying their reducibility while varying the latent dimension. Finally, we interpret the numerical results under the lens of the discussed theoretical considerations, offering a comprehensive landscape for nonlinear reduction methods for general Advection-Diffusion dynamics.

1. Introduction

In recent years, the topic of *Data Reduction* has become pervasive in the scientific computing community, as it contributes to popular ideas spread across many related fields, e.g. Numerical Analysis, Manifold Learning, Representation Learning, Generative Modeling, and more. To pursue a comprehensive analysis of the topic, we start by defining what is meant here by *reduction*: identifying a low dimensional set of coordinates in the data space that encodes most of the information contained in the full dataset. The main assumption behind a reduction task is that the data manifold $\mathcal{M}$ at hand lives, with a small approximation error, on a lower dimensional manifold $\mathcal{Z}_N$ of dimension N embedded in the higher dimensional space of dimension D, with $N \ll D$. Identifying such reduced space is convenient for interpretability but also helps speed up demanding computations, as it offers a suitable lower-dimensional dataset to process.

This work aims to uncover some important relations between reduction strategies developed in different field, trying to align the main perspectives by underlying their common ground and identifying their differences. Here, we want to address the characterization of nonlinear reduction strategies, generally adopted as a way to overcome the typical limitations of linear reduction when addressing highly nonlinear manifolds. To present a broad overview of different nonlinear approaches, we refer to the two-stage scheme reported in Figure 1 interpreting them as decomposable in two main stages, shaping their differences only by adopting different choices for the maps Φ and Z_N and the reduced space $\mathcal{Z}_N$.

[1] mathLab, Mathematics Area, SISSA, via Bonomea 265, I-34136 Trieste, Italy

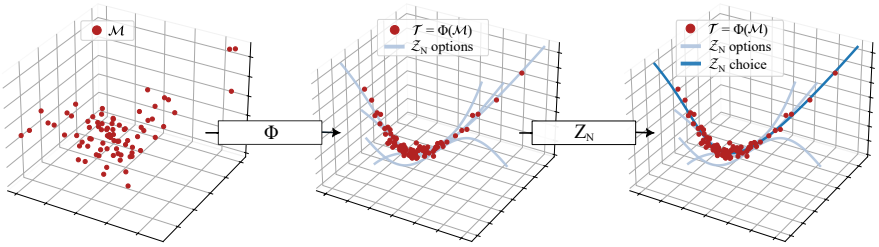

FIGURE 1. Nonlinear reduction two-stage pipeline: considering the original manifold $\mathcal{M}$, the map Φ transforms the data into the new manifold $\mathcal{T}$, on which reduction is performed through the operator Z_N.

In particular, in Figure 1, we represent the initial data-manifold $\mathcal{M}$ undergoing a series of transformations. The first one is given by the map Φ such that $\mathcal{T} = \Phi(\mathcal{M})$, and stands for a generic *pre-processing step* that adjusts the original manifold to make it more suitable to the actual *reduction step*, represented by the map Z_N in the right part of the scheme. This map brings the transformed data manifold $\mathcal{T}$ into a low-dimensional space $\mathcal{Z}_N$ spanned by N "bases" that well capture the structure of $\mathcal{T}$, and associates each datum with its low-dimensional coefficients, for instance belonging to $\mathbb{R}^N$, which characterize its reduced representation with respect to the chosen bases. The bright blue line precisely pictures the space $\mathcal{Z}_N$, embedded in the space of $\mathcal{T}$, that has been chosen between an ensemble of possible N-dimensional options as the one that best represents the (transformed) data.

In this work, we address some well-known nonlinear reduction methodologies developed in the context of Numerical Analysis with Model Order Reduction techniques, Manifold Learning, and Representation Learning. Each one of them has

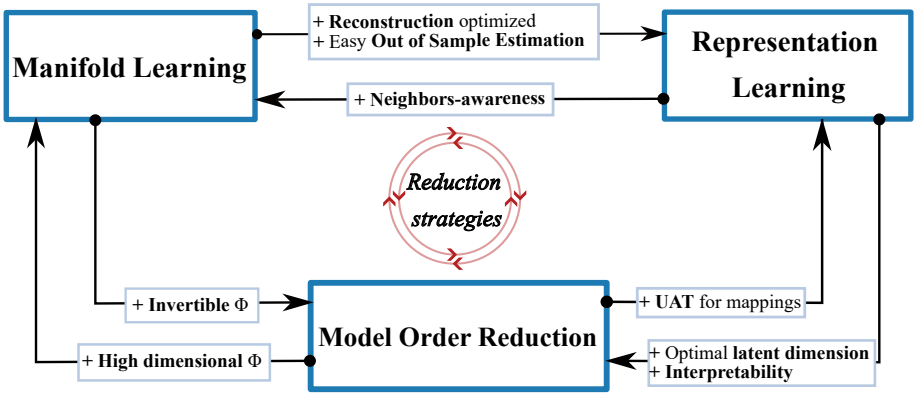

FIGURE 2. Bi-directional flow comparing the properties gathered respectively with methods of Manifold Learning, Representation Learning, and Model Order Reduction in the context of nonlinear strategies for data compression.

some traditional strategies to perform data reduction, therefore, to give a glance at their connection and strengths, we report in Figure 2 a general scheme representing their links. Precisely, a double flow is proposed to report the properties gained in the reduction process passing from one methodology to another. These comments and connections will be deeply described and clarified in the following sections, nevertheless, it is worth giving a short brief overview of their main differences. Representation Learning usually builds the reduction problem as a supervised optimization of deep networks having strong approximation capabilities demonstrated in the Universal Approximation Theorem (UAT) [25], aiming at minimizing the reconstruction error of each datum independently from its neighbors. Manifold Learning learns a latent space by exploiting some data-structure awareness to properly cluster the data in the new low-dimensional latent space. Finally, Reduced Order Models (ROMs) in Numerical Analysis, generally design an interpretable reduction pipeline providing convergence estimates on its reduced representations. On the contrary, learning-based algorithms usually lack interpretability, not clarifying the meaning of the obtained latent spaces, i.e. not providing information on the chosen bases for $\mathcal{Z}_N$ corresponding to the low-dimensional coefficients provided as latent representations of the data.

We start our analysis by introducing the main concepts and presenting *linear reduction*, together with its main difficulties when dealing with some specific problems, for which we report a case test in Section 2. In particular, we refer to the setting of Advection-Diffusion problems, which notoriously model various physical phenomena such as heat and mass transfer, fluid motion, oceanography, air pollution, meteorology, and so on [53, 21]. Moreover, by parametrizing both the Advection and the Diffusion contributions in the differential model, one can obtain a set of significantly different dynamics to test reduction strategies on: *pure Advection*, *pure Diffusion*, and *Advection-Diffusion* problems. Successively, in Section 3, we dive into the details of the two-stage characterization for well-known ROM approaches (see Subsection 3.1), Manifold Learning (see Subsection 3.2) and Representation Learning (see Subsection 3.3), pointing out in each case the construction strategies adopted for Φ, Z_N and $\mathcal{Z}_N$. Finally, in Section 4, we report some numerical results regarding the application of the aforementioned reduction strategies to Advection-Diffusion problems, highlighting the pros and cons of each approach to the different simple, yet meaningful, test cases presented.

2. Challenges in linear reduction

Given the parametric data manifold $\mathcal{M} := \{u_{\mu_1}, \dots, u_{\mu_n}\}$, where $u_{\mu_i} \in \mathcal{X},\ \forall \mu_i \in \mathcal{P} := \{\mu_1, \dots, \mu_n\}$ with $\mathcal{X}$ Hilbert space, traditional approaches to reduction rely on linear approximation spaces of the form $\mathcal{Z}_N = \text{span}\{\psi_j\}_{j=1}^N$ with $\psi_j \in \mathcal{X},\ \forall j \in \{1, \dots, N\}$, so that

$$u_{\mu_i} \simeq \Pi_{\mathcal{Z}_N} u_{\mu_i} := \overline{u} + \sum_{j=1}^{N} z_j(\mu_i)\psi_j,\ \forall \mu_i \in \mathcal{P},$$

given the parameter-dependent reduced variables $z_j : \mathcal{P} \to \mathbb{R},\ \forall j \in \{1, \dots, N\}$ and $\overline{u} := \frac{1}{n}\sum_{i=1}^{n} u_{\mu_i}$.

A well-known linear reduction method is the Principal Component Analysis (PCA) or Proper Orthogonal Decomposition (POD), depending on the field of

application. It is particularly important because it offers the best-reduced representation of $\mathcal{M}$ in terms of L_2 error. Indeed, it uses as $\{\psi_j\}_{j=1}^N$ the N orthogonal directions that maximize the variance of the data manifold $\mathcal{M}$. It can be proven that they are the first N eigenfunctions of the covariance operator $\mathcal{K} : \mathcal{X} \to \mathcal{X}$ defined as

$$\mathcal{K}(v) := \frac{1}{n}\sum_{i=1}^{n}\langle u_{\mu_i} - \overline{u}, v\rangle_{\mathcal{X}}(u_{\mu_i} - \overline{u}) \quad \forall v \in \mathcal{X},$$

so that

$$\mathcal{K}(\psi_j) = \lambda_j\psi_j, \quad \forall j \in \{1, \dots, N\}. \tag{1}$$

Consequently, the coefficients $\{z_j(\mu_i)\}_{j=1}^N$ are obtained by projecting the data on the new basis:

$$z_j(\mu_i) := \langle u_{\mu_i}, \psi_j\rangle_{\mathcal{X}}, \quad \forall i \in \{1, \dots, n\},\ \forall j \in \{1, \dots, N\}.$$

For instance, in the case of $\mathcal{X} \equiv \mathbb{R}^D$, if one defines the matrix of centered data as $U \in \mathbb{R}^{n\times D}$ (i.e. the mean of each column is zero), then $\{\psi_j\}_{j=1}^N$ are exactly the first N eigenvectors of the covariance matrix $K = U^TU$.

Linear reduction thus relies on the availability of low-dimensional accurate representation of the manifold $\mathcal{M}$ itself. Nevertheless, this is not always the case, as the linear reducibility of $\mathcal{M}$ can be investigated via the Kolmogorov N-width $d_N(\mathcal{M})$:

$$d_N(\mathcal{M}) := \inf_{\mathcal{Z}_N}\sup_{u\in\mathcal{M}} \|u - \Pi_{\mathcal{Z}_N}u\|_{\mathcal{X}},$$

where $\Pi_{\mathcal{Z}_N} : \mathcal{M} \to \mathcal{Z}_N$ is the orthogonal projection operator onto the reduced space $\mathcal{Z}_N$. In particular, the Kolmogorov N-width can be seen as the worst approximation error that can be made on data belonging to $\mathcal{M}$ using the best linear subspace $\mathcal{Z}_N$ of dimension N in terms of $\|\cdot\|_{\mathcal{X}}$. Furthermore, a good approximation of the decay of $d_N(\mathcal{M})$ while increasing N is notoriously provided by the decay of the eigenvalues of the covariance operator $\mathcal{K}$ used in PCA [14, 2, 19].

While under certain circumstances, depending on the properties of $\mathcal{M}$, it is possible to demonstrate that $d_N(\mathcal{M})$ rapidly decays for increasing N [3, 8], there exists a large class of problems for which linear approximation does not offer an accurate reduced representation for a reasonably small amount "modes". It is, for instance, the case of data collected from Advection-dominated dynamics [50, 35], which are commonly described as the prototypical linearly nonreducible $\mathcal{M}$, i.e. characterized by a slow decay of $d_N(\mathcal{M})$.

This can be seen by comparing the decay of PCA eigenvalues of the manifold $\mathcal{M}$ obtained by solving the following Advection-Diffusion system for different values of the parameters c_T and c_D in the following equation in the spatial domain $\Omega \subset \mathbb{R}^D$:

$$\begin{cases} \partial_t u(\boldsymbol{x}, t) + c_T\nabla_{\boldsymbol{x}} \cdot u(\boldsymbol{x}, t) - c_D\Delta_{\boldsymbol{x}}u(\boldsymbol{x}, t) = 0, & \boldsymbol{x} \in \Omega,\ t > 0 \\ u(\boldsymbol{x}, 0) = u_0, & \boldsymbol{x} \in \Omega. \end{cases} \tag{2}$$

Indeed, for $c_T = 0, c_D \neq 0$ one obtains a purely diffusive system, while for $c_T \neq 0, c_D = 0$ a purely advective one. Note that, when dealing with time-dependent systems, one may choose to include the time variable in the parameters' set. In Figure 3 we compare the eigenvalues decay of the Advection, Diffusion, and Advection-Diffusion manifolds, both obtained by collecting solutions of Equation

(2) for $t \in \mathcal{P} = [0, T]$ with $T = 0.5$ for fixed values of the other parameters: respectively $(c_T = 4, c_D = 0)$, $(c_T = 0, c_D = 0.1)$, and $(c_T = 4, c_D = 0.1)$. Thus, it is clear that, even in the "non-parametric setting" (only evolving the systems in time), linear reduction is not well-suited for Advection-dominated problems, since the correspondent PCA eigenvalues exhibit a significantly slower decay w.r.t. the other models, for which diffusive effects are present.

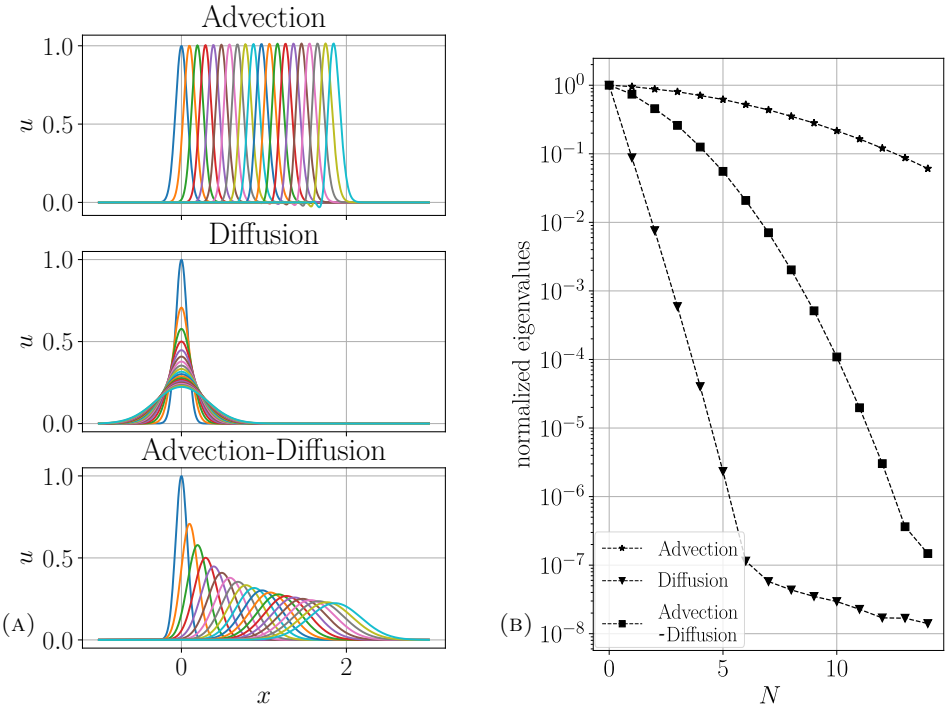

FIGURE 3. On the left, some solutions at different times of the pure Advection $(c_T = 4, c_D = 0)$, pure Diffusion $(c_T = 0, c_D = 0.1)$, and Advection-Diffusion systems $(c_T = 4, c_D = 0.1)$. On the right, their eigenvalues decay for increasing values of $N \in \{1, \dots, 15\}$.

3. Nonlinear Reduced Models: different Learning Strategies, same Reduction Pipeline

In this section, we address well-known nonlinear reduction strategies from different research fields by linking them to the scheme reported in Figure 1. In particular, we identify a nonlinear reduction process with a two-stage pipeline composed of a possible *pre-processing* step transforming the manifold $\mathcal{M}$ into $\mathcal{T}$ through the operator $\Phi : \mathcal{M} \to \mathcal{T}$, and of a standard *reduction step*, which aims to find the operator $Z_N : \mathcal{T} \to \mathcal{Z}_N \subseteq \mathbb{R}^N$ such that the reduced manifold $\mathcal{Z}_N := Z_N(\mathcal{T})$ has dimension N and approximates sufficiently well $\mathcal{T}$. Thus, the main goal of the first step is to obtain a "simpler" representation of the manifold $\mathcal{T}$ which is more suitable for reduction, while the second one actually searches for the more convenient N-dimensional description of the resulting manifold $\mathcal{T} := \Phi(\mathcal{M})$. For instance, in

the case of projection-based ROMs, the last step aims to find N bases $\{\psi_j\}_{j=1}^N$ on which to project $\mathcal{T}$, so that Z_N is taken as a projection operator. In addition, since it is generally desirable to reconstruct the original data $\mathcal{M}$ from $\mathcal{Z}_N$, the methods should also provide the operators $\Phi^{-1} : \mathcal{T} \to \mathcal{M}$ and $Z_N^{-1} : \mathcal{Z}_N \to \mathcal{T}$. Nevertheless, we underline the abuse of notation for Φ^{-1} and Z_N^{-1}, as they are defined only to provide a way to map back the reduced representation to the original manifold, being Φ and Z_N not necessarily invertible operators. Therefore, each method gives its own characterization to Φ^{-1} and Z_N^{-1} (see for instance Section 3.3 for an example of non-invertible Z_N).

Recalling Section 2, in the linear PCA/POD approach one can identify Φ with the identity operator, so that $\mathcal{T} = \mathcal{M}$, and $Z_N : \Phi(u_{\mu_i}) \mapsto \{z_j(\mu_i)\}_{j=1}^N \in \mathcal{Z}_N$ with the operator assigning to each transformed datum the reduced variables $\boldsymbol{z}(\mu_i) := \{z_j(\mu_i)\}_{j=1}^N$ for a given parameter value $\mu_i \in \mathcal{P}$ corresponding to the projection of $\Phi(u_{\mu_i})$ on the bases ψ_j. Then the "inverse" operator $Z_N^{-1}(\boldsymbol{z}) = \sum_{i=j}^N z_j \psi_j$ projects back the reduced variables on $\mathcal{T}$, being $\{\psi_j\}_{j=1}^N$ defined as in (1) and corresponding to the chosen bases of $\mathcal{T}(\equiv \mathcal{M})$. Moreover, we highlight that the same reasoning applies for methods that rely on the interpolation of different local projection-based ROMs correspondent to different parameters values [1, 36]. Indeed, such approaches can be represented by taking Φ as the identity map (similarly to the classical PCA case), and choosing Z_N to be a projection-operator on a set of parameter dependent bases $\{\psi_j(\mu)\}_{j=1}^N$, so that $\mathcal{Z}_N \equiv \mathcal{Z}_N(\mu)$.

Now, we continue by discussing how different nonlinear reduction techniques fit in the above pipeline. With this aim, we present below the characterization of these two stages for some numerical nonlinear ROMs, for well-known Manifold Learning techniques, and for Auto-Encoders, belonging to the Representation Learning field.

3.1. **Techniques from Model Order Reduction: Registration methods.** Nonlinear approaches to reduction, as underlined in [48, 54], can be subdivided into *Eulerian* and *Lagrangian* ones. Eulerian methods search for possibly nonlinear parameter-dependent maps $Z_N \equiv Z_{N,\mu}$ and $Z_N^{-1} \equiv Z_{N,\mu}^{-1}$, with $\mu \in \mathcal{P}$, while keeping $\Phi \equiv Id$. Lagrangian approaches instead aim to perform classical linear reduction on $\mathcal{T} = \Phi(\mathcal{M})$ with a general nonlinear map Φ, obtained by properly optimizing the image set $\mathcal{T}$ to make it well-suited for linear reduction [35].

Concerning Eulerian approaches, examples might rely on Grassmannian learning [58], displacement interpolation [41, 10], or transformed snapshot methods [32, 39, 42]. Nevertheless, in this section, we are focusing on Lagrangian methods, given their capability to exploit the reduction pipeline of Figure 1.

Regarding Lagrangian nonlinear approaches, we start by reporting the example of *Registration Methods*, which generalizes the linear Reduced Basis (RB) technique [20, 38, 43] by coupling it with a previous transformation of the data space. The term *registration* has been used in the fields of computer vision and pattern recognition, to refer to the process of finding a spatial transformation that aligns two datasets. In the context of the Numerical Analysis of parametric Partial Differential Equations (pPDEs) it has been adopted to indicate the process of finding a transformation Φ that improves the linear reduction of a given manifold exploiting such alignment strategy.

Indeed, *Registration Methods* [13, 51, 49] deal with the task of finding a parameter-dependent bijection $\Phi \equiv \Phi_\mu$ such that, given the initial manifold $\mathcal{M}$ of sampled

solutions of a pPDE $\{u_{\mu_i}(\boldsymbol{x})\}_{i=1}^n$ with $\boldsymbol{x} \in \Omega \subset \mathbb{R}^D$, for an ensemble of different values of the parameter $\{\mu_i\}_{i=1}^n$, the image manifold $\mathcal{T} := \Phi(\mathcal{M})$ is suitable for linear reduction, i.e. has a nice Kolmogorov decay. This is done by defining the *optimal map* Φ_μ as the one that minimizes the L_2 distance between the transformed data and the reference solution $u_{\overline{\mu}} \in \mathcal{M}$, for which $\Phi_{\overline{\mu}}$ is the identity map. In particular, Φ_μ is searched as an affine map of the form

$$\Phi_\mu^{\hat{\boldsymbol{a}}}(\boldsymbol{x}) := \boldsymbol{x} + \sum_{m=1}^{M} \hat{a}_m(\mu)\varphi_m(\boldsymbol{x}),$$

where $\{\varphi_m\}_{m=1}^M$ is a given set of bases that well approximate the space $L_2(\Omega)$ (e.g. the set of Legendre polynomials), and $\hat{\boldsymbol{a}} = \{\hat{a}_m\}_{m=1}^M$ are coefficients obtained through the following regularized constrained minimization problem, where ε, C serve to weakly impose $J(\Phi_\mu^{\hat{\boldsymbol{a}}}(\boldsymbol{x})) \in [\varepsilon, \frac{1}{\varepsilon}]$ for the Jacobian, and thus that $\Phi_\mu^{\hat{\boldsymbol{a}}}(x)$ is a bijection from Ω into itself:

$$\begin{aligned} \hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}\in\mathbb{R}^M} & \|u_\mu \circ \Phi_\mu^{\boldsymbol{a}}(\boldsymbol{x}) - u_{\overline{\mu}}(\boldsymbol{x})\|_{L_2(\Omega)}^2 + \xi |\Phi_\mu^{\boldsymbol{a}}(\boldsymbol{x})|_{H_2(\Omega)}^2 \\ \text{s.t.} & \int_\Omega e^{\left(\frac{\varepsilon - J(\Phi_\mu^{\hat{\boldsymbol{a}}}(\boldsymbol{x}))}{C}\right)} + e^{\left(\frac{J(\Phi_\mu^{\hat{\boldsymbol{a}}}(\boldsymbol{x})) - \frac{1}{\varepsilon}}{C}\right)} d\boldsymbol{x} \leq \delta. \end{aligned} \tag{3}$$

The H_2 regularizing term measures the deviations from linear maps, as for $v \in H_2(\Omega)$ linear the seminorm $|v|_{H_2(\Omega)} := \sum_{i,j=1}^D \int_\Omega (\partial_{i,j}^2 v)^2 = 0$. This can be further interpreted as a Tikhonov regularization and has the effect of controlling the gradient of the Jacobian, which is important for ensuring invertibility of $\Phi_\mu^{\hat{\boldsymbol{a}}}(x)$.

Thus, Registration Methods offer a general and interpretable framework, clarifying the optimization process that leads to a given choice of Φ while ensuring its invertibility and the possibility of reconstructing the original samples. Furthermore, since they rely on classical linear methods for the reduction step, they can be coupled with any reduction strategy by selecting the dimension N looking at the eigenvalues decay of the covariance operator computed on the transformed manifold $\mathcal{T} = \Phi(\mathcal{M}) := \{u_\mu \circ \Phi_\mu^{\hat{a}} \mid u_\mu \in \mathcal{M}\}$, not imposing it a-priori.

Moreover, an interesting direction for Registration has been offered by Optimal Transport (OT) theory, which focuses on the existence and computability of *optimal transport maps* between two probability measures with finite second order moments $\rho, \sigma \in \mathcal{P}_2(\Omega, \mathcal{W}_2)$ with respect to the Wasserstein metric $\mathcal{W}_2$. It can be reinterpreted from a Registration perspective by considering $u_{\mu_i}(\boldsymbol{x}) \in \mathcal{P}_2(\Omega, \mathcal{W}_2)$, $\forall \mu_i \in \mathcal{P}$, computing Φ through the transport maps $T_{u_\mu \to u_{\overline{\mu}}}$ between the distributions $u_\mu(\boldsymbol{x})$ and the reference element $u_{\overline{\mu}}(\boldsymbol{x})$. Some works developed and discussed this intuition, proposing frameworks in which OT maps can be useful to find a proper $\mathcal{T} = \Phi(\mathcal{M}) := \{(T_{u_\mu \to u_{\overline{\mu}}})_\# u_\mu \mid u_\mu \in \mathcal{M}\}$ in case of problems with slow decay, such as the Advection-dominated ones [7, 12].

In particular, [7] proposes a way to build a proper transformation from the manifold $\mathcal{M} \subset \mathcal{P}(\Omega)$ to a space as simple as possible, even coinciding with a proper reference solution $u_{\overline{\mu}}$ for a given reference value of the parameter $\overline{\mu}$. In this setting, Φ_μ coincides with the OT map $T_{u_\mu \to \overline{u}}$ between u_μ and $u_{\overline{\mu}}$, as it minimizes the following problem

$$T_{u_\mu \to u_{\overline{\mu}}} = \arg\min_T \int_\Omega \frac{1}{2} |T(u_\mu) - u_{\overline{\mu}}|^2 d\boldsymbol{x},$$

thus corresponding to the *optimal map* according to the definition of Wasserstein distance, which reads

$$\mathcal{W}_2(u_\mu, u_{\overline{\mu}}) = \min_T \int_\Omega \frac{1}{2}|T(u_\mu) - u_{\overline{\mu}}|^2 d\boldsymbol{x}.$$

3.2. **Techniques from Manifold Learning: Kernel and Spectral methods.** Manifold Leaning belongs to the Metric Learning field, which, by definition, aims at optimizing a problem-dependent metric that provides a well-suited similarity measure to represent the given dataset by means of a few *clusters*. This task can be approached both in the supervised and unsupervised context, and in the latter case it precisely takes the name of Manifold Learning. Such techniques aim to understand the manifold geometrical and topological features. Precisely, this is often done by identifying the bases spanning an informative lower dimensional space, on which to project the original data manifold to both preserve and emphasize possibly hidden core information. As a result, manifold learning often provides significant hints for the reduction, as it searches for a low dimensional representation of the high dimensional data while preserving relevant information of its original topology [30, 27, 22, 5].

For instance, PCA (see Section 2) belongs to this class of methods, as it finds new bases maximizing the variance of the reduced representation so that the original data can be treated with a reduced number of coordinates while still preserving its intrinsic variance. A nonlinear generalization to PCA is offered by its "kernelized" version [46, 6], which only modifies the Φ. Indeed, in kernel PCA (kPCA), Φ is assumed to be a high-dimensional function defining the correspondent kernel k used in computations. The idea behind a high-dimensional $\mathcal{T}$ is to obtain a sparsification of data on new directions advantageous to the subsequent PCA reduction.

To enable the computation of the PCA decomposition of the high-order nonlinear transformation of the original manifold, the so-called *kernel trick* is introduced. More precisely, if each datum u_{μ_i} belongs to $\mathcal{X} \equiv \mathbb{R}^D$ so that $\boldsymbol{U} := \{u_{\mu_i}\}_{i=1}^n \in \mathbb{R}^{n\times D}$ and $\Phi : \mathbb{R}^D \to \mathbb{R}^M$ with M arbitrarily large, the kernel is thus defined as a dot product in the *feature space* $\mathbb{R}^N$

$$k(u_{\mu_i}, u_{\mu_j}) = \langle \Phi(u_{\mu_i}), \Phi(u_{\mu_j}) \rangle ,$$

and the *kernel trick* enables to compute the kPCA reduction algorithm without explicitly knowing Φ, but only relying on k.

Indeed, performing PCA on the *feature space*, we study the spectral decomposition of the matrix $\boldsymbol{C} := \boldsymbol{\Phi}^T\boldsymbol{\Phi}$, where $\boldsymbol{\Phi} \in \mathbb{R}^{(n\times M)}$ is the evaluation matrix of the function Φ at the elements of the data manifold $\mathcal{M} = \{u_{\mu_i}\}_{i=1}^n$. It can be verified that, being $\{\lambda_j^{\boldsymbol{C}}\}_{j=1}^R$ and $\{\boldsymbol{v}_j^{\boldsymbol{C}}\}_{j=1}^R$ respectively the eigenvalues and eigenvectors of the matrix $\boldsymbol{C}$ of rank R

$$\boldsymbol{C}\boldsymbol{v}_j^{\boldsymbol{C}} = \lambda_j^{\boldsymbol{C}}\boldsymbol{v}_j^{\boldsymbol{C}}, \quad \forall j \in \{1, \dots, R\},$$

they can be put in relation with the ones $\{\lambda_j^{\boldsymbol{K}}\}_{j=1}^R$ and $\{\boldsymbol{v}_j^{\boldsymbol{K}}\}_{j=1}^R$ of the *kernel matrix* $\boldsymbol{K}_{i,j} := k(u_{\mu_i}, u_{\mu_j}) = \boldsymbol{\Phi}\boldsymbol{\Phi}^T \in \mathbb{R}^{(n\times n)}$ which by definition has the same rank R. Thus, one obtains

$$\boldsymbol{v}_j^{\boldsymbol{C}} = \boldsymbol{\Phi}^T\boldsymbol{v}_j^{\boldsymbol{K}} \quad \text{and} \quad \lambda_j^{\boldsymbol{C}} = \lambda_j^{\boldsymbol{K}} \quad \forall j \in \{1, \dots, R\},$$

and the normalizing conditions $\lambda_j^{\boldsymbol{K}}(\boldsymbol{v}_j^{\boldsymbol{K}} \cdot \boldsymbol{v}_j^{\boldsymbol{K}}) = 1,\ \forall j \in \{1, \dots, R\}$, are applied to obtain $(\boldsymbol{v}_k^{\boldsymbol{C}} \cdot \boldsymbol{v}_k^{\boldsymbol{C}}) = 1$.

This way, the reduced representation of a given data $u \in \mathbb{R}^D$ can be identified only using the kernel k without explicitly knowing $\Phi(u)$, but only computing its projection of the transformed onto the first $N < R$ principal components $\{\boldsymbol{v}_j^{\boldsymbol{C}}\}_{j=1}^N$:

$$\Phi(\boldsymbol{u})\boldsymbol{v}_j^{\boldsymbol{C}} = \Phi(\boldsymbol{u})\boldsymbol{\Phi}^T v_j^{\boldsymbol{K}} = \sum_{l=1}^{n} k(u, u_l)(\boldsymbol{v}_j^{\boldsymbol{K}})_l, \quad \forall j \in \{1, \dots, N\}.$$

Nevertheless, it is to be underlined that without Φ we can not reconstruct the original samples $u \in \mathcal{M}$ starting from their reduced representation. In particular, given a kernel k satisfying some general assumptions, Mercer's theorem [47] ensures the existence and uniqueness of Φ, but it is in general too complicated to be explicitly written.

However, for the identification of relevant clusters of significantly nonlinear data manifolds, many different kernels have proved to be beneficial, and in this section we report some well-known Manifold Learning techniques that can be reconducted to kernelized versions of PCA [17], as it is the case of the ones listed in Table 1.

Method	**Kernel**
Principal Component Analysis	$\boldsymbol{U}^T\boldsymbol{U}$
Multi-Dimensional Scaling	$-\frac{1}{2}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}$
Isomap	$-\frac{1}{2}\boldsymbol{H}\boldsymbol{D}^{(g)}\boldsymbol{H}$
Spectral Clustering	$\boldsymbol{L}^\dagger$
Local Linear Embedding	$\boldsymbol{M}^\dagger$

TABLE 1. Manifold Learning methods written as kPCA ones with different kernel functions.

3.2.1. *Multi-Dimensional Scaling (MDS).* MDS [17, 44] aims to preserve a notion of distance between points, which is encoded in the matrix $\boldsymbol{D}$ and then *double-centered* via the matrix $\boldsymbol{H} := \boldsymbol{I}_n - \frac{1}{n}\boldsymbol{1}_n$. This is done by projecting the data on the directions which maximise the distance between the *training points*, similarly to what is done with the matrix K in the classical case of PCA, to preserve the variance in the reduced representation.

3.2.2. *Isomap.* It follows the same principles as MDS, only changing the distance used to compute matrix $\boldsymbol{D}^{(g)}$, which is not the Euclidean distance, but the geodesic one, defined as the length of the shortest path between two points on the possibly curvy (i.e., nonlinear) manifold. It can be computed by approximating it with a sum of local Euclidean distances between neighbour points [9, 52, 4].

3.2.3. *Spectral Clustering.* The main idea behind Spectral Clustering (SC) is to divide the data $\boldsymbol{U}$ into clusters which are as separated as possible [55, 34]. It is done through an optimization procedure based on the so-called *adjacency matrix* $\boldsymbol{W}$, defined as

$$\boldsymbol{W}_{i,j} = \begin{cases} w(\boldsymbol{u}_i, \boldsymbol{u}_j) & \boldsymbol{u}_j \in \text{Neighbours}(\boldsymbol{u_i}), \\ 0 & \boldsymbol{u}_j \notin \text{Neighbours}, \\ 0 & \boldsymbol{u}_i = \boldsymbol{u}_j, \end{cases} \tag{4}$$

accordingly to a *weighting function* $w : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. The optimization process can be translated in a minimization of the sum of the weights of the points belonging to different clusters

$$\min_{\boldsymbol{v}} \sum_{i=1}^{n} \sum_{j=1}^{n} w(\boldsymbol{u}_i, \boldsymbol{u}_j)(\boldsymbol{v}_i - \boldsymbol{v}_j),$$

where $\boldsymbol{v}$ stands for a discriminator function for a specific cluster. In this way, the reduced representation can be more suitable to be analyzed with simple clustering algorithms.

It has been proven that such problem is equivalent to the eigenvalue problem of the Laplacian matrix

$$\boldsymbol{L} := \boldsymbol{W}_D - \boldsymbol{W},$$

where $\boldsymbol{W}_D := diag(\{d_1, \ldots, d_n\})$ with $d_i := \sum_{j=1}^{n} w_{i,j}$. Thus, this directly translates in the kPCA maximization problem taking the pseudo-inverse of the Laplacian $\boldsymbol{L}^\dagger$ as *kernel matrix*. Therefore, SC is equivalent to a kPCA version in which particular attention is given to the adjacency links of *training data*, wanting to maximise the information encoded by the Laplacian. Giving an intuitive interpretation of $\boldsymbol{L}$, we remark that it can be seen as a measure of how *curved* the neighborhood of a point is, so the SC approach aims at preserving the geometrical "second order" characteristics of the points neighborhoods.

3.2.4. *Locally Linear Embedding.* Among the techniques searching for a reduced representation of $\boldsymbol{U}$, the goal of Locally Linear Embedding (LLE) methods is to preserve the linear reconstruction weights of data neighbors [45, 18, 57]. Indeed such approach is divided in two fundamental optimization steps: the first in the high-dimensional space, and the second in the low-dimensional one. During the first stage, one searches for the weight matrix $\tilde{\boldsymbol{W}}$ defined as

$$\tilde{\boldsymbol{W}} := \arg\min_{\tilde{\boldsymbol{W}}} \sum_{i=1}^{n} \left\| \boldsymbol{u}_i - \sum_{j \in \mathrm{kNN}(\boldsymbol{x}_i)} \tilde{w}_{i,j} \boldsymbol{u}_j \right\|, \tag{5}$$

so that a local linear reconstruction of each point is obtained. Then, the next stage exploits $\tilde{\boldsymbol{W}}$ to search for the optimal points $\boldsymbol{Y}$ in the reduced space, thus solving an second minimization problem of the form

$$\boldsymbol{Y} := \arg\min_{\boldsymbol{Y}} \sum_{i=1}^{n} \left\| \boldsymbol{y}_i - \sum_{j \in \mathrm{kNN}(\boldsymbol{u}_i)} \tilde{w}_{i,j} \boldsymbol{y}_j \right\|,$$

that can actually be seen as an eigenvalue problem for the matrix $\boldsymbol{M}$ defined as

$$\boldsymbol{M} := (\boldsymbol{I} - \tilde{\boldsymbol{W}})^T (\boldsymbol{I} - \tilde{\boldsymbol{W}}).$$

Also in this case, in order to cast the problem as a kPCA one, the minimization can be transformed into a maximization problem by taking the pseudo-inverse $\boldsymbol{M}^\dagger$.

3.3. **Techniques from Representation Learning: Auto-Encoders.** From the field of Representation Learning, we include in the analysis Auto-Encoders (AE), one of the main Deep Learning (DL) architectures used for reduction [31, 37, 16, 15, 26]. Indeed, Data Reduction has become a popular pre-processing step, prior to the actual learning algorithms, as it has been understood that the effectiveness and interpretability of machine learning methods largely relies on the selection of data

features they are applied to. As a result, a significant part of the research work has been focusing on designing pipelines for data transformations that produce a suitable *representation* for successful learning. To this aim, Auto-Encoders have been presented as a peculiar deep learning architecture to reduce the inputs dimension, in order to build an informative surrogate representation of them. In particular, Auto-Encoders are composed by an encoder and a decoder, which are layers that respectively reduce and reconstruct the original data manifold.

We can refer to the pipeline reported in Figure 1 also for their description. Indeed, they search for the reduced representation which *minimize* the reconstruction error on the training dataset, relying on the UAT, that enables a representation through a deep network of a large variety of invertible encoding maps satisfying such minimization constraint. The fixed reduced dimension N is expressed by construction by the number of neurons of the latent representation, and thus has to be established a priori to the training phase. In our notation, the encoder f_e approximates the function $Z_N : \mathcal{X} \to \mathbb{R}^N$ that transforms the datum u_{μ_i} correspondent to the parameter μ_i in its latent representation $\boldsymbol{z}_{\mu_i} \in \mathbb{R}^N$, while the decoder f_d corresponds to Z_N^{-1} that maps back into $\mathcal{X}$ the latent representation of the input. Nevertheless, in contrast to the Registration and Manifold Learning cases, for Auto-Encoders we can not assume $\mathcal{Z}_N$ to be a linear space and Z_N to be a projection operator with respect to some bases $\{\psi_j\}_{j=1}^N$, as it is nontrivial to explicitly identify such bases hypothetically spanning the latent space Z_N. Indeed, even assuming the encoder as a composition of a nonlinear map Φ and a reduction step Z_N, there is no easy way to characterize the latent space [28].

Given the dataset $\boldsymbol{U} = \{\boldsymbol{u}_{\mu_i}\}_{i=1}^n$ with $\boldsymbol{u}_{\mu_i} \in \mathcal{X} \equiv \mathbb{R}^D$, $\forall i \in \{1, \dots, n\}$, the training procedure follows the minimization of the loss $\mathcal{L}$:

$$\mathcal{L}(\boldsymbol{U}) := \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{u}_i - f_d \circ f_e(\boldsymbol{u}_i)\|_{\mathcal{X}}, \tag{6}$$

where f_e and f_d are respectively denoting the encoder and the decoder functions.

Additional terms can be inserted into the loss formulation as regularizing factors, as it is the case of Sparse Auto-Encoders [33], and of Contractive Auto-Encoders [40]. The former class enforces sparsity in the hidden activations of the layers $\boldsymbol{A}$ inserting a regularized L_1 norm term:

$$\mathcal{L}_{sparse}(\boldsymbol{U}) := \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{u}_i - f_d \circ f_e(\boldsymbol{u}_i)\|_{\mathcal{X}} + \lambda \|\boldsymbol{A}\|_{L_1}. \tag{7}$$

The second class instead aims to build an architecture resistant to input non-informative fluctuations, with the goal of inferring a stable latent representation. In order to do that, a regularizing term is inserted regarding the Jacobian matrix of the hidden layer $\boldsymbol{J}_A$, which represents the derivative of each hidden node with respect to each input component:

$$\mathcal{L}_{contractive}(\boldsymbol{U}) := \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{u}_i - f_d \circ f_e(\boldsymbol{u}_i)\|_{\mathcal{X}} + \lambda \|\boldsymbol{J}_A\|_{L_2}, \tag{8}$$

so that high derivatives w.r.t. inputs are discouraged.

Therefore, Auto-Encoders obtain a latent representation through loss minimizations that are not explicitly aware of the topological characteristics of the manifold, as instead happened in Section 3.2 with Manifold Learning techniques. On the

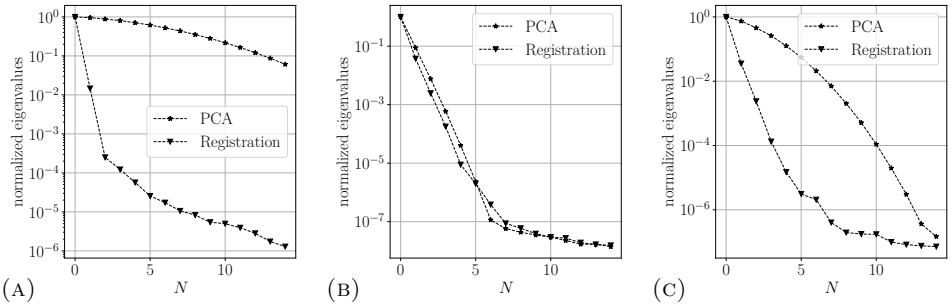

FIGURE 4. Eigenvalues decay for PCA and Registration methods for pure Advection, pure Diffusion, and Advection-Diffusion problems respectively on the left, in the middle, and on the right.

other hand, such DL architecture focuses on enabling an accurate reconstruction of samples, which, in contrast, is not always possible in the previous cases due to the lack of an explicit Φ^{-1} operator.

Furthermore, in many cases Auto-Encoders provide an advantageous reduction architecture for Out Of Sample (OOS) estimation, i.e. the identification of the reduced representation of data not belonging to the *training set*. Indeed, for all the approaches presented in Sections 3.1 and 3.2, one has to perform some sort of multivariate regression strategy on the set of "train reduced variables" $\{\mu_i, \boldsymbol{z}_i\}_{i=1}^n$ to obtain the reduction operator $Z_N : u_\mu \mapsto \boldsymbol{z}_\mu \ \forall \mu \in \mathcal{P}$. In the DL case, instead, Z_N is the encoder itself, which automatically maps any new input in a new point of the learnt reduced space.

4. Numerical Results

In this section, we apply the nonlinear reduction methods described above to the Advection-Diffusion problem described in Equation (2) (see Section 2). In particular, we test them on the pure Advection problem ($c_D = 0, c_T = 4$), pure Diffusive problem ($c_D = 0.1, c_T = 0$), and Advection-Diffusion one ($c_D = 0.1, c_T = 4$). For simplicity, in each test case, the parameters are kept fixed and the time is evolved in the range $[0, T]$ with $T = 0.5s$, and we fix $\Omega := [-1, 3]$. The *training set* is composed by 20 solution samples u_{t_i} for t_i uniformly distributed in $[0, T]$, while the *testing set* by 200 samples uniformly distributed in $[0, T]$ [1].

First of all, in Figure 4 we report, in the three test cases, the eigenvalues decay of the PCA applied to the dataset $\mathcal{T}$ transformed through the Registration Method, compared with the decay of the PCA performed on the original $\mathcal{M}$ also reported in Figure 3b. As one can see, the Registration procedure appears to be particularly useful in treating Advective behavior, as in both Figures 4a and 4c its decay appears much faster than the PCA one.

One can explain this result by recalling the optimization process behind Registration (see Equation (3)). Indeed, it aims to find a suitable map Φ to bring each datum u_{t_i} as close as possible to a reference one $u_{\bar{t}}$ (here we choose $\bar{t} = 0$). This

[1]The Python codes with the implementation of the methods exploited for the numerical results will be made available upon request.

helps in the presence of advective dynamics as it builds a "less spread" transformed manifold $\mathcal{T}$.

Furthermore, in Figures 5, 6 and 7, we report the reconstruction with the associated error for the training and testing data, respectively for PCA, Registration methods, and "vanilla" AutoEncoder for small latent dimension ($N = 2$). We construct our "vanilla" AutoEncoder with Linear Layers interleaved by nonlinear activation function (tanh) for both encoder and decoder, made up also by an hidden layer of dimension $\frac{D}{2}$. In all the cases, the test data is reconstructed by applying the backward operators Φ^{-1} and Z_N^{-1} to the "test reduced variables" $z_{\mu_{test}} \in \mathcal{Z}_N$, which are obtained through Kernel Ridge Regression (kRR) based on inverse multiquadratic RBFs [56] applied on the "train reduced variables" $\{\mu_i, z_i\}_{i=1}^n$ collected during the offline phase. We remark that, in such context, no Manifold Learning strategy is reported due to the fact that in general the map Φ is not explicitly known, making it impossible to reconstruct the original sample u_{μ_i} from its latent representation z_{μ_i}. The results highlight the potential of Registration Methods, which, with very few latent variables, provide good approximations in all cases, also comparing to AutoEncoders, while proposing an interpretable reduction pipeline.

To better capture the error behavior with increasing values of the latent dimension N for both Numerical methods and AutoEncoders, we report in Figure 8 the a-posteriori errors on the test dataset. Regarding PCA and Registration methods, one can note that they decrease as long as their eigenvalues significantly decay, according to Figures 4. Indeed, for Registration the error is stable for $N > 5$ in each test case, while the PCA error continues decreasing also for higher N values

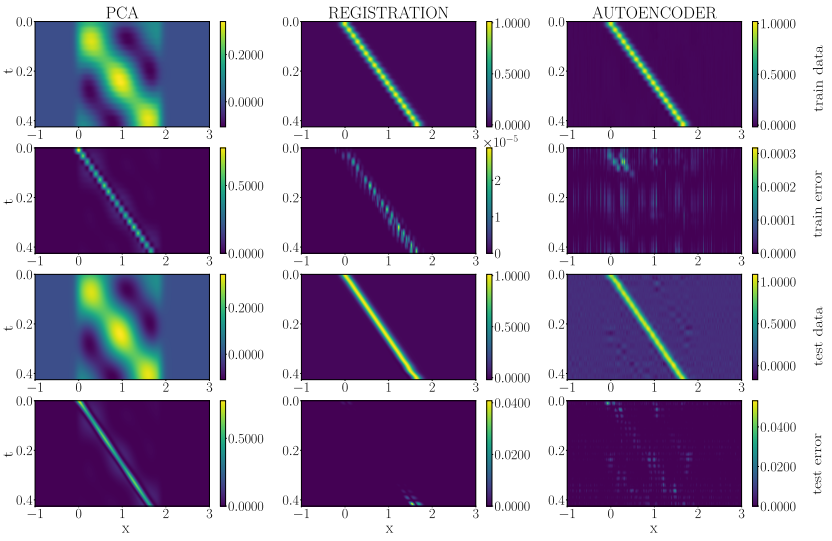

FIGURE 5. Reconstruction of train and test data shown in first and third row, respectively, with the correspondent l_2 errors (second and fourth row) of PCA (left-most column), Registration method (middle column), and AutoEncoder (right-most column) for pure Advection problem with $N = 2$.

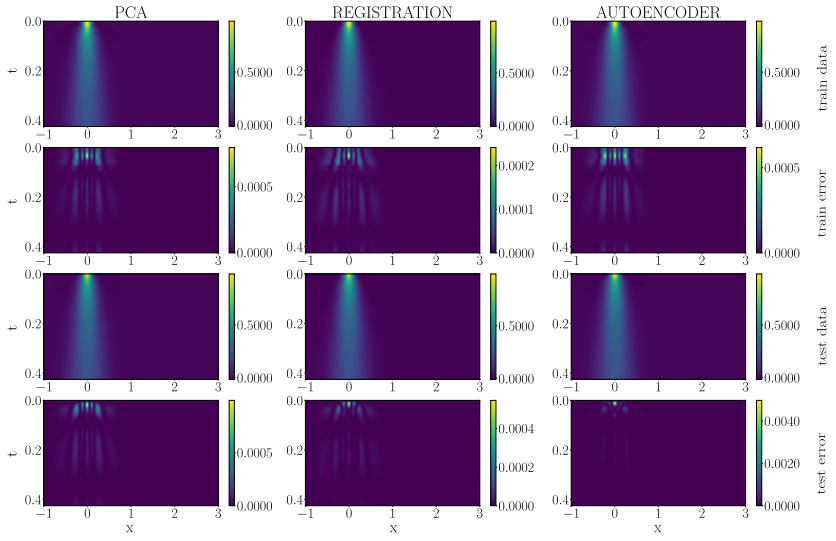

FIGURE 6. Reconstruction of train and test data shown in first and third row, respectively, with the correspondent l_2 errors (second and fourth row) of PCA (left-most column), Registration method (middle column), and AutoEncoder (right-most column) for pure Diffusion problem with $N = 2$

when the problem is not only Diffusive (Figures 8a and 8c). Concerning AutoEncoders, their performance does not reach the accuracy of Registration but they demonstrate, in general, an average approximation capability for both Advection and Diffusion problems.

Nevertheless, it is to be underlined that these results correspond to the "vanilla" forms of the above methods, as many extensions have been proposed in the literature to improve their performance for many test cases [29, 11, 37, 26, 23, 24]. Since in this work we are interested in analyzing the main ideas behind a reduction process, we do not offer further insights in the direction of "hyper-optimizing" the architecture, and we only focus on providing a comprehensive analysis of the main strategies giving an intuition on the motivations behind the successful choices for the identification of Φ and Z_N.

Concerning the Manifold Learning techniques, in Figure 9 we report the eigenvalues decay of the kernel methods of Table 1 for the three test cases. For this class of methods, one has to differ between the strategy to adopt for Diffusion and Advection problems in order to obtain the best performance. Indeed, while linear approaches are well-suited when dealing with the former, more advanced techniques are required to obtain a good representation at a low cost for the latter. Above all, LLE seems to provide a good alternative for Advection-dominated problems. This is particularly interesting when considering its similarity to Registration methods. Indeed, both work with mappings that bring one sample into another: it is the case of Φ such that $u_{\mu_i} \circ \Phi_{\mu_i} \simeq u_{\overline{\mu}}$ for Registration, and of $\tilde{\boldsymbol{W}}$ in Equation (5) for

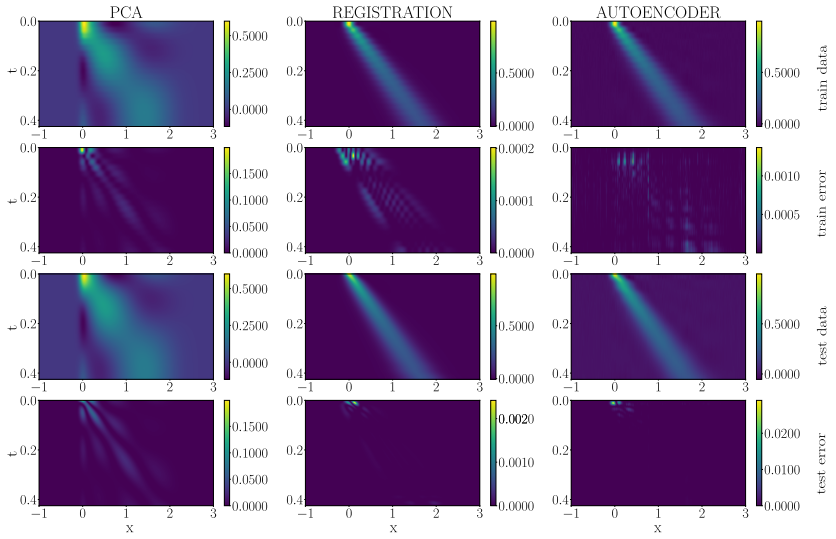

FIGURE 7. Reconstruction of train and test data shown in first and third row, respectively, with the correspondent l_2 errors (second and fourth row) of PCA (left-most column), Registration method (middle column), and AutoEncoder (right-most column) for Advection-Diffusion problem with $N = 2$

LLE. Therefore, the experiments seem to suggest the utility of considering notions of distance between samples in the analysis of Advection-dominated problems.

Finally, we report in Figure 10 the reduced variables dynamic for different reduction strategies, i.e. their evolution with respect to $t \in [0, T]$ for $N = 2$. In general, one can note that the Diffusion problem has a simpler latent dynamic, except that in the subplot corresponding to the Registration method. Indeed, in this case, the dynamics exhibits very similar shapes in all three test cases.

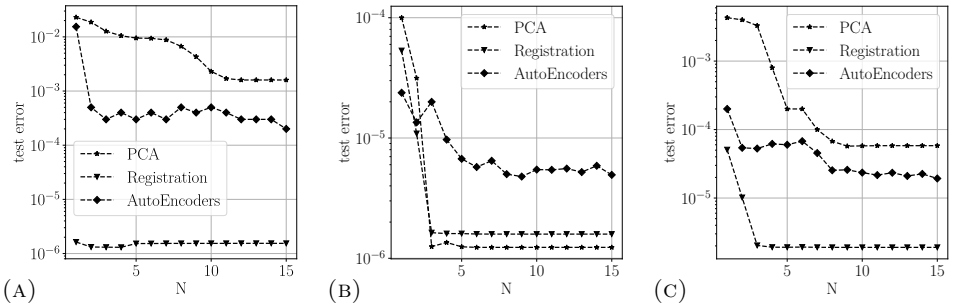

FIGURE 8. A-posteriori error on the test dataset of PCA, Registration method, and AutoEncoder for pure Advection, pure Diffusion, and Advection-Diffusion problems respectively on the left, in the middle, and on the right.

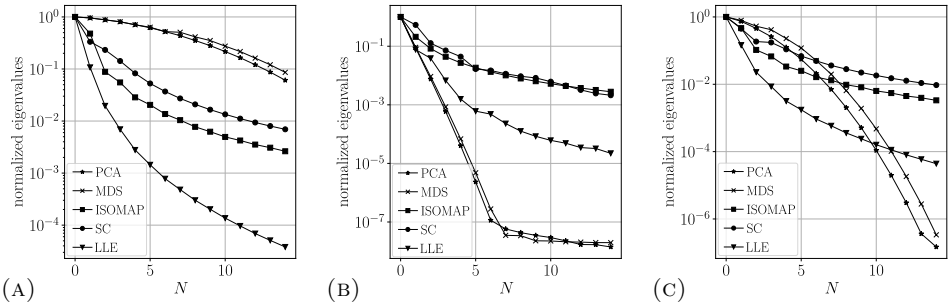

FIGURE 9. Eigenvalues decay of Manifold Learning methods interpretable as kPCA computed for pure Advection, pure Diffusion, and Advection-Diffusion problems respectively on the left, in the middle, and on the right.

5. CONCLUSIONS

In this work, we offer a panoramic view of the main ideas concerning nonlinear reduction. In order to do this, we introduce technical insights on various methods belonging to different research fields: Numerical Analysis, Manifold Learning, and Representation Learning.

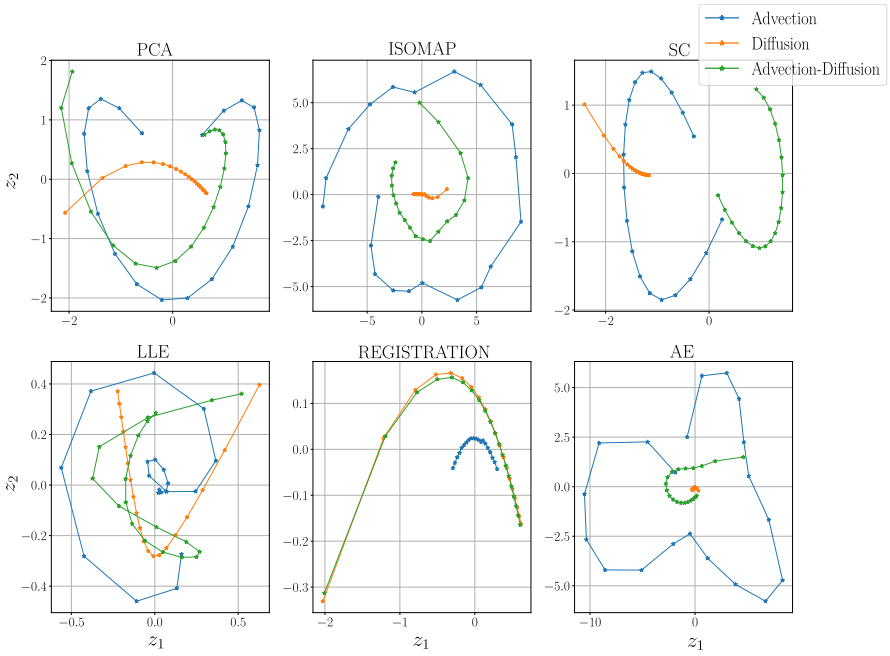

FIGURE 10. Latent variables dynamic of different nonlinear reduction methods, computed for pure Advection, pure Diffusion, and Advection-Diffusion problems.

We highlight their common ground by referring, for their explanation, to the same two-stage pipeline displayed in Figure 1, which we take to be the general scheme of each Nonlinear Reduction strategy. Therefore, interpreting many well-known methods from different fields to the same scheme, we identify their differences by pointing out their diversity in the choices made along the pipeline and explaining their consequences in the effectivity of the final strategy.

Finally, we apply the described methods to the class of Advection-Diffusion problems and analyze their approximation capabilities. In particular, we conduct the analysis on three test cases: pure Diffusion, pure Advection, and Advection-Diffusion problems. We report the reconstruction errors for the classes of Nonlinear ROMs in Numerical Analysis (Registration Methods) and Representation Learning (AutoEncoders), and we analyze the spectral decay for Manifold Learning approaches (kPCA). By analyzing the whole set of approaches, we observed that, as expected, while Diffusion problems seem to be naturally well-suited for linear reduction, Advection problems are successfully reduced only when a proper *pre-processing stage* is applied to the original manifold $\mathcal{M}$. In particular, LLE and Registration Methods seem to lead the way, suggesting to search for a Φ which transforms all the samples in a reference one or, more in general, that embeds some notion of distance between the training samples.

In conclusion, we foresee that such a common interpretation of both numerical and learning-based reduction can help in future research, by opening broader and deeper communication between the fields.

6. Acknowledgements

FP and **GR** acknowledge the support provided by the European Union- NextGenerationEU, in the framework of the iNEST- Interconnected Nord-Est Innovation Ecosystem (iNEST ECS00000043– CUP G93C22000610007) consortium and its CC5 Young Researchers initiative. The authors also like to acknowledge INdAM-GNSC and MIUR (Italian ministry for university and research) through FAREX-AROMA-CFD project, P.I. Prof. Gianluigi Rozza, for their support.

References

[1] D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM Journal on Scientific Computing*, 33(5):2169–2198, 2011.

[2] F. Arbes, C. Greif, and K. Urban. The Kolmogorov N-width for linear transport: Exact representation and the influence of the data. *Advances in Computational Mathematics*, 51(2):13, 2025.

[3] M. Bachmayr and A. Cohen. Kolmogorov widths and low-rank approximations of parametric elliptic PDEs. *Mathematics of Computation*, 86(304):701–724, 2017.

[4] M. Balasubramanian and E. L. Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.

[5] A. Bellet, A. Habrard, and M. Sebban. *Metric learning*. Morgan & Claypool Publishers, 2015.

[6] G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66:259–294, 2007.

[7] T. Blickhan. A Registration Method for Reduced Basis Problems Using Linear Optimal Transport. *SIAM Journal on Scientific Computing*, 46(5):A3177–A3204, 2024.

[8] A. Buffa, Y. Maday, A. T. Patera, C. Prud'Homme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical modelling and numerical analysis*, 46(3):595–603, 2012.

[9] H. Choi and S. Choi. Robust kernel isomap. *Pattern recognition*, 40(3):853–862, 2007.

[10] S. Cucchiara, A. Iollo, T. Taddei, and H. Telib. Model order reduction by convex displacement interpolation. *Journal of Computational Physics*, 514:113230, 2024.
[11] S. Dutta, P. Rivera-Casillas, B. Styles, and M. W. Farthing. Reduced order modeling using advection-aware autoencoders. *Mathematical and Computational Applications*, 27(3):34, 2022.
[12] V. Ehrlacher, D. Lombardi, O. Mula, and F.-X. Vialard. Nonlinear model reduction on metric spaces. Application to one-dimensional conservative PDEs in Wasserstein spaces. *ESAIM: Mathematical Modelling and Numerical Analysis*, 54(6):2159–2197, 2020.
[13] A. Ferrero, T. Taddei, and L. Zhang. Registration-based model reduction of parameterized two-dimensional conservation laws. *Journal of Computational Physics*, 457:111068, 2022.
[14] M. S. Floater, C. Manni, E. Sande, and H. Speleers. Best low-rank approximations and Kolmogorov n-widths. *SIAM Journal on Matrix Analysis and Applications*, 42(1):330–350, 2021.
[15] N. R. Franco, D. Fraulin, A. Manzoni, and P. Zunino. On the latent dimension of deep autoencoders for reduced order modeling of PDEs parametrized by random fields. *Adv. Comput. Math.*, 50:96, 2024.
[16] S. Fresca, L. Dede', and A. Manzoni. A Comprehensive Deep Learning-Based Approach to Reduced Order Modeling of Nonlinear Time-Dependent Parametrized PDEs. *Journal of Scientific Computing*, 87(2):61, 2021.
[17] B. Ghojogh, M. Crowley, F. Karray, and A. Ghodsi. *Elements of dimensionality reduction and manifold learning*. Springer, 2023.
[18] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley. Locally linear embedding and its variants: Tutorial and survey. *arXiv preprint arXiv:2011.10925*, 2020.
[19] C. Greif and K. Urban. Decay of the Kolmogorov N-width for Wave Problems. *Applied Mathematics Letters*, 96:216–222, 2019.
[20] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics. Springer International Publishing AG, Cham, 1st ed. 2016 edition, 2015.
[21] S. R. Jena and G. S. Gebremedhin. Computational technique for heat and advection–diffusion equations. *Soft Computing*, 25(16):11139–11150, 2021.
[22] M. Kaya and H. Ş. Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.
[23] M. Khamlich, F. Pichi, M. Girfoglio, A. Quaini, and G. Rozza. Optimal transport-based displacement interpolation with data augmentation for reduced order modeling of nonlinear dynamical systems. *Journal of Computational Physics*, 531:113938, 2025.
[24] M. Khamlich, F. Pichi, and G. Rozza. Optimal Transport–Inspired Deep Learning Framework for Slow-Decaying Kolmogorov n-Width Problems: Exploiting Sinkhorn Loss and Wasserstein Kernel. *SIAM Journal on Scientific Computing*, pages C235–C264, 2025.
[25] M. Kvalheim and E. Sontag. Why should autoencoders work? *Transactions on Machine Learning Research*, 2024.
[26] K. Lee and K. T. Carlberg. Model Reduction of Dynamical Systems on Nonlinear Manifolds Using Deep Convolutional Autoencoders. *Journal of Computational Physics*, 404:108973, 2020.
[27] Y. Ma and Y. Fu. *Manifold learning theory and applications*, volume 434. CRC press Boca Raton, 2012.
[28] L. Magri and A. K. Doan. On interpretability and proper latent decomposition of autoencoders. *arXiv preprint arXiv:2211.08345*, 2022.
[29] R. Maulik, B. Lusch, and P. Balaprakash. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids*, 33(3), 2021.
[30] M. Meilă and H. Zhang. Manifold learning: What, how, and why. *Annual Review of Statistics and Its Application*, 11, 2024.
[31] O. M. Morrison, F. Pichi, and J. S. Hesthaven. GFN: A Graph Feedforward Network for Resolution-Invariant Reduced Operator Learning in Multifidelity Applications. *Computer Methods in Applied Mechanics and Engineering*, 432:117458, 2024.
[32] N. J. Nair and M. Balajewicz. Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks. *International Journal for Numerical Methods in Engineering*, 117(12):1234–1262, 2019.
[33] A. Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[34] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[35] M. Nonino, F. Ballarin, G. Rozza, and Y. Maday. A Reduced Basis Method by Means of Transport Maps for a Fluid–Structure Interaction Problem with Slowly Decaying Kolmogorov n-width. *Advances in Computational Science and Engineering*, 1(1):36–58, 2023.

[36] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1):A168–A192, 2014.

[37] F. Pichi, B. Moya, and J. S. Hesthaven. A Graph Convolutional Autoencoder Approach to Model Order Reduction for Parametrized PDEs. *Journal of Computational Physics*, 501:112762, 2024.

[38] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. La Matematica per Il 3+2, 92. Springer International Publishing, Cham, 1st ed. 2016. edition, 2016.

[39] J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40(3):A1322–A1344, 2018.

[40] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot. Higher order contractive auto-encoder. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II 22*, pages 645–660. Springer, 2011.

[41] D. Rim and K. T. Mandli. Model reduction of a parametrized scalar hyperbolic conservation law using displacement interpolation, 2018. `https://arxiv.org/abs/1805.05938`.

[42] D. Rim, B. Peherstorfer, and K. T. Mandli. Manifold Approximations via Transported Subspaces: Model Reduction for Transport-Dominated Problems. *SIAM Journal on Scientific Computing*, 45(1):A170–A199, 2023.

[43] G. Rozza, F. Ballarin, L. Scandurra, and F. Pichi. *Real Time Reduced Order Computational Mechanics: Parametric PDEs Worked Out Problems*, volume 5 of *SISSA Springer Series*. Springer Nature Switzerland, Cham, 2024.

[44] N. Saeed, H. Nam, M. I. U. Haq, and D. B. Muhammad Saqib. A survey on multidimensional scaling. *ACM Computing Surveys (CSUR)*, 51(3):1–25, 2018.

[45] L. K. Saul and S. T. Roweis. An introduction to locally linear embedding. *http://www.cs.toronto.edu/∼roweis/lle/publications.html*, 2000.

[46] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[47] I. Steinwart and C. Scovel. Mercer's theorem on general domains: On the interaction between measures, kernels, and RKHSs. *Constructive Approximation*, 35:363–417, 2012.

[48] T. Taddei. A registration method for model order reduction: data compression and geometry reduction. *SIAM Journal on Scientific Computing*, 42(2):A997–A1027, 2020.

[49] T. Taddei. An optimization-based registration approach to geometry reduction. *arXiv preprint arXiv:2211.10275*, 2022.

[50] T. Taddei, S. Perotto, and A. Quarteroni. Reduced Basis Techniques for Nonlinear Conservation Laws. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(3):787–814, 2015.

[51] T. Taddei and L. Zhang. Space-time registration-based model reduction of parameterized one-dimensional hyperbolic pdes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 55(1):99–130, 2021.

[52] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[53] M. Thongmoon and R. McKibbin. A comparison of some numerical methods for the advection-diffusion equation. *Res. Lett. Inf. Math. Sci.*, 10:49–62, 01 2006.

[54] D. Torlo. Model Reduction for Advection Dominated Hyperbolic Problems in an ALE Framework: Offline, Online Phases and Error Estimator. In A. Sequeira, A. Silvestre, S. S. Valtchev, and J. Janela, editors, *Numerical Mathematics and Advanced Applications ENUMATH 2023, Volume 2*, pages 409–419, Cham, 2025. Springer Nature Switzerland.

[55] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

[56] H. Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.

[57] Z. Zhang and J. Wang. Mlle: Modified locally linear embedding using multiple weights. *Advances in neural information processing systems*, 19, 2006.

[58] R. Zimmermann, B. Peherstorfer, and K. Willcox. Geometric subspace updates with applications to online adaptive nonlinear model reduction. *SIAM Journal on Matrix Analysis and Applications*, 39(1):234–261, 2018.