Shallow neural networks for fluid flow reconstruction with limited sensors

N. Benjamin Erichson, Lionel Mathelin, Zhewei Yao, Steven Brunton, Michael Mahoney, J. Nathan Kutz

Shallow Neural Networks for Fluid Flow Reconstruction with Limited Sensors

N. Benjamin Erichson[1], Lionel Mathelin[2], Zhewei Yao[3], Steven L. Brunton[4], Michael W. Mahoney[1], and J. Nathan Kutz[5]

[1]ICSI and Department of Statistics, University of California, Berkeley
[2]Université Paris-Saclay, CNRS, LIMSI, 91400 Orsay, France
[3]Department of Mathematics, University of California, Berkeley
[4]Department of Mechanical Engineering, University of Washington, Seattle
[5]Department of Applied Mathematics, University of Washington, Seattle

Abstract

In many applications, it is important to reconstruct a fluid flow field, or some other high-dimensional state, from limited measurements and limited data. In this work, we propose a shallow neural network-based learning methodology for such fluid flow reconstruction. Our approach learns an end-to-end mapping between the sensor measurements and the high-dimensional fluid flow field, without any heavy preprocessing on the raw data. No prior knowledge is assumed to be available, and the estimation method is purely data-driven. We demonstrate the performance on three examples in fluid mechanics and oceanography, showing that this modern data-driven approach outperforms traditional modal approximation techniques which are commonly used for flow reconstruction. Not only does the proposed method show superior performance characteristics, it can also produce a comparable level of performance with traditional methods in the area, using significantly fewer sensors. Thus, the mathematical architecture is ideal for emerging global monitoring technologies where measurement data are often limited.

Article Info

Correspondence:
N. Benjamin Erichson
erichson@berkeley.edu

Disclosure:
Keywords:
Shallow learning, deep learning, neural networks, sensors, flow field estimation, fluid dynamics, machine learning

1 Introduction

The ability to reconstruct coherent flow features from limited observation can be critically enabling for applications across the physical and engineering sciences [1, 2, 3, 4, 5]. For example, efficient and accurate

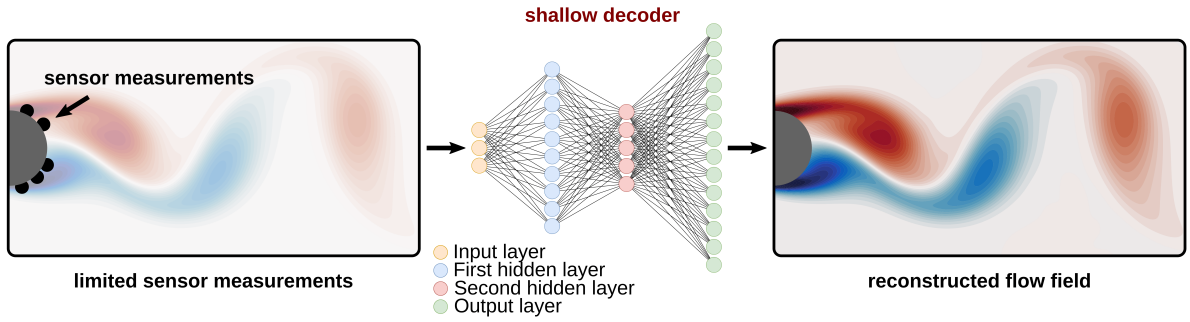


Figure 1: Illustration of our SHALLOW DECODER which maps a few sensor measurements $\boldsymbol{s} \in \mathbb{R}^5$ to the estimated field $\widehat{\boldsymbol{x}} \in \mathbb{R}^{78,406}$. In other words, this neural network based learning methodology provides an end-to-end mapping between the sensor measurements and the fluid flow field.

fluid flow estimation is critical for active flow control, and it may help to craft more fuel-efficient automobiles as well as high-efficiency turbines. The ability to reconstruct important fluid flow features from limited observation is also central in applications as diverse as cardiac bloodflow modeling and climate science [6]. All of these applications rely on estimating the structure of fluid flows based on limited sensor measurements.

More concretely, the objective is to estimate the flow field $\boldsymbol{x} \in \mathbb{R}^m$ from sensor measurements $\boldsymbol{s} \in \mathbb{R}^p$, that is, to learn the relationship $\boldsymbol{s} \mapsto \boldsymbol{x}$. The restriction of limited sensors gives $p \ll m$. The sensor measurements $\boldsymbol{s}$ are collected via a sampling process from the high-dimensional field $\boldsymbol{x}$. We can describe this process as

$$\boldsymbol{s} = \boldsymbol{H}(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{H} : \mathbb{R}^m \to \mathbb{R}^p$ denotes a measurement operator. Now, the task of flow reconstruction requires the construction of an inverse model that produces the field $\boldsymbol{x}$ in response to the observations $\boldsymbol{s}$, which we may describe as

$$\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{s}), \tag{2}$$

where $\boldsymbol{G} : \mathbb{R}^p \to \mathbb{R}^m$ denotes a non-linear forward operator. However, the measurement operator $\boldsymbol{H}$ may be unknown or highly-nonlinear in practice. Hence, the problem is often ill-posed, and we cannot directly invert the measurement operator $\boldsymbol{H}$ to obtain the forward operator $\boldsymbol{G}$.

Fortunately, given a set of training examples $\{\boldsymbol{x}_i, \boldsymbol{s}_i\}_i$, we may learn a function $\mathcal{F}$ to approximate the forward operator $\boldsymbol{G}$. Specifically, we aim to learn a function $\mathcal{F} : \boldsymbol{s} \mapsto \widehat{\boldsymbol{x}}$ which maps a limited number of measurements to the estimated state $\widehat{\boldsymbol{x}}$:

$$\widehat{\boldsymbol{x}} \,=\, \mathcal{F}\left(\boldsymbol{s}\right), \tag{3}$$

so that the misfit is small, *e.g.*, in an Euclidean sense over all sensor measurements

$$\|\mathcal{F}(\boldsymbol{s}) - \boldsymbol{G}(\boldsymbol{s})\|_2^2 < \epsilon,$$

where ϵ is a small positive number. Neural network based inversion is common practice in machine learning [7], dating back to the late 80's [8]. This powerful learning paradigm is also increasingly used for flow reconstruction, prediction, and simulations [9, 10, 11, 12, 13]. In particular, deep inverse transform learning is an emerging concept [14, 15, 16, 17], which has been shown to outperform traditional methods in applications such as denoising, deconvolution, and super-resolution.

Here, we explore shallow neural networks (SNNs) to learn the input-to-output mapping between the sensor measurements and the flow field. Figure 1 shows a design sketch for the proposed framework for fluid flow reconstruction. We can express the network architecture (henceforth called SHALLOW DECODER (SD)), more concisely as follows:

$$\boldsymbol{s} \mapsto \text{first hidden layer} \mapsto \text{second hidden layer} \mapsto \text{output layer} \mapsto \widehat{\boldsymbol{x}}.$$

SNNs are considered to be networks with very few hidden layers. We favor shallow over deep architectures, because the simplicity of SNNs allows faster training, less tuning, and easier interpretation (and also since it works, and thus there is no need to consider deeper architectures).

There are several advantages of this mathematical approach over traditional scientific computing methods for fluid flow reconstruction [18, 19, 20, 21, 4]. First, the SD considered here features a linear last layer and provides a supervised joint learning framework for the low-dimensional approximation space of the flow field and the map from the measurements to this low-dimensional space. This allows the approximation basis to be tailored not only to the state space but also to the associated measurements, preventing observability issues. In contrast, these two steps are disconnected in standard methods (discussed in more detail in Section 2). Second, the method allows for flexibility in the measurements, which do not necessarily have to be linearly related to the state, as in many standard methods. Finally, the shallow decoder network produces interpretable features of the dynamics, potentially improving on classical proper orthogonal decomposition (POD), also known as principal component analysis (PCA), low-rank features. For instance, Figure 2 shows that the basis learned via an SNN exhibits elements resembling physically consistent quantities, in contrast with alternative POD-based modal approximation methods that enforce orthogonality. The interpretation of the last (linear) layer is as follows: a given mode is constituted by the value of each spatially localized weights connecting the associated given node in the last hidden layer to nodes of the output layer.

Limitations of our approach are standard to data-driven methods, in that the training data should be as representative as possible of the system, in the sense that it should comprise samples drawn from the same statistical distribution as the testing data.

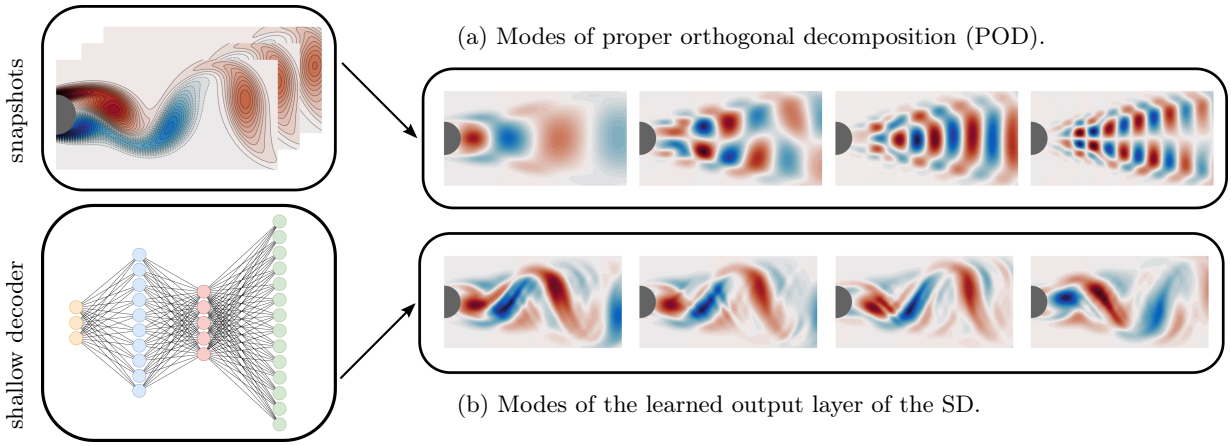
The paper is organized as follows. Sec. 2 discusses traditional modal approximations techniques. Then, in Sec. 3, the specific implementation and architecture of our SHALLOW DECODER is described. Results are presented in Sec. 4 for various applications of interest. We aim to reconstruct (a) the vorticity field of a flow behind a cylinder from a handful sensors on the cylinder surface, (b) the mean sea surface temperature from weekly sea surface temperatures for the last 26 years, and (c) the velocity field of a turbulent isotropic flow. We show that a very small number of sensor measurements is indeed sufficient for flow reconstruction in these applications. Further, we show that the SHALLOW DECODER can handle non-linear measurements and is robust to measurement noise. The results show significantly improved performance compared to traditional modal approximations techniques. The paper concludes in Sec. 5 with a discussion and outlook of the use of SNNs for more general flow field reconstructions.

2 Background on high-dimensional state estimation

The task of reconstructing from a limited number of measurements to the high-dimensional state-space is made possible by the fact that the dynamics for many complex systems, or datasets, exhibit some sort of low-dimensional structure. This fact has been exploited for state estimation using (i) a tailored basis, such as POD, or (ii) a general basis in which the signal is sparse, *e.g.*, typically a Fourier or wavelet basis will suffice. In the former, *gappy POD* methods [22] have been developed for principled reconstruction strategies [18, 19, 20, 21, 4]. In the latter, *compressive sensing* methods [23, 24, 25] serve as a principled technique for reconstruction. Both techniques exploit the fact that there exists a basis in which the high-dimensional state vector has a sparse, or compressible, representation. In [26], a basis is learned such that it leads to a sparse approximation of the high-dimensional state while enforcing observability from the sensors.

Next, we describe standard techniques for the estimation of a state $\boldsymbol{x}$ from observations $\boldsymbol{s}$, and we discuss observability issues. Established techniques for state reconstruction are based on the idea that a field $\boldsymbol{x}$ can be expressed in terms of a rank-k approximation

$$\boldsymbol{x} \approx \widehat{\boldsymbol{x}} = \sum_{j=1}^{k} \boldsymbol{\phi}_j \, \nu_j = \boldsymbol{\Phi} \, \boldsymbol{\nu} \,, \tag{4}$$



(a) Modes of proper orthogonal decomposition (POD).

(b) Modes of the learned output layer of the SD.

Figure 2: Dominant modes learned by the SHALLOW DECODER in contrast to the POD modes. These dominant features show that the SD constructs a reasonable characterization of the flow behind a cylinder. Indeed, by not constraining the modes to be linear and orthogonal, as is enforced with POD, a potentially more interpretable feature space can be extracted from data. Such modes can be exploited for reconstruction of the state space from limited measurements and limited data.

where $\{\boldsymbol{\phi}_j\}_j$ are the *modes* of the approximation and $\{\nu_j\}_j$ are the associated coefficients. The approximation space is derived from a given training set using unsupervised learning techniques. A typical approach to determine the approximation modes is POD [27, 18, 19, 4]. Randomized methods for linear algebra enable the fast computation of such approximation modes [28, 29, 30, 31, 32, 33]. Given the approximation modes $\boldsymbol{\Phi}$, estimating the state $\boldsymbol{x}$ reduces to determining the coefficients $\boldsymbol{\nu}$ from the sensor measurements $\boldsymbol{s}$ using supervised techniques. These typically aim to find the minimum-energy or minimum-norm solution that is consistent in a least-squares sense with the measured data.

2.1 Standard approach: Estimation via POD based methods

Two POD-based methods are discussed, which we will refer to as POD and POD PLUS in the following. Both approaches reconstruct the state with POD modes, by estimating the coefficients from sensor information. The POD modes $\boldsymbol{\Phi}$ are obtained via the singular value decomposition of the mean centered training set $\boldsymbol{X} = (\boldsymbol{x}_1 \dots \boldsymbol{x}_n)$, with typically $n \leq m$:

$$\boldsymbol{X} = \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{V}^{\mathsf{T}}\,, \tag{5}$$

where the columns of $\boldsymbol{U} \in \mathbb{R}^{m\times n}$ are the left singular vectors and the columns of $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ are the right singular vectors. The corresponding singular values are the diagonal elements of $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$. Now, we define the approximation modes as $\boldsymbol{\Phi} := \boldsymbol{U}_k$, by selecting k left singular vectors, with $k \leq p$. Typically, we select the dominant k singular vectors as approximation modes, however, there are exceptions to this rule as discussed below.

2.1.1 Standard POD-based method

Let a linear measurement operator $\boldsymbol{H} : \mathbb{R}^m \to \mathbb{R}^p$ describe the relationship between the field and the associated observations, $\boldsymbol{s} = \boldsymbol{H}\,\boldsymbol{x}$. The approximation of the field $\boldsymbol{x}$ with the approximation modes $\{\boldsymbol{\phi}_j\}_j$ is obtained by solving the following equation for $\boldsymbol{\nu} \in \mathbb{R}^n$:

$$\boldsymbol{s} = \boldsymbol{H}\,\boldsymbol{x} \approx \boldsymbol{H}\boldsymbol{\Phi}\,\boldsymbol{\nu}\,. \tag{6}$$

A standard approach is to simply solve the following least-squares problem

$$\boldsymbol{\nu} \in \arg\min_{\widetilde{\boldsymbol{\nu}}} \; \|\boldsymbol{s} - \boldsymbol{H}\,\boldsymbol{\Phi}\,\widetilde{\boldsymbol{\nu}}\|_2^2\,. \tag{7}$$

The solution with the minimum L^2-norm is given by:

$$\boldsymbol{\nu} = (\boldsymbol{H}\,\boldsymbol{\Phi})^{+}\;\boldsymbol{s}\,, \tag{8}$$

with the superscript $^+$ denoting the Moore-Penrose pseudo-inverse. In this situation, the high-dimensional state is then estimated as

$$\boldsymbol{x} \approx \widehat{\boldsymbol{x}} = \boldsymbol{\Phi}\,\boldsymbol{\nu}\,. \tag{9}$$

This approach is hereafter referred to as POD and has been used in previous efforts, *e.g.*, [34, 35].

With a nonlinear measurement operator $\boldsymbol{H}$, the problem formulates similarly as a nonlinear least squares problem:

$$\boldsymbol{\nu} \in \arg\min_{\widetilde{\boldsymbol{\nu}}} \; \|\boldsymbol{s} - \boldsymbol{H}\,(\boldsymbol{\Phi}\,\widetilde{\boldsymbol{\nu}})\|_2^2\,. \tag{10}$$

In this case, no closed form solution is available in general and a nonlinear optimization problem must be solved, whose computational burden limits the online (real-time) field reconstruction capability. Further, the solution of the, often ill-posed, problem is not necessarily unique and does not allow for a reliable estimate. In contrast, the shallow decoder is trained end-to-end and essentially learns to associate measurements to the right solution (see Section 3 for details) .

2.1.2 Improved POD-based method

The standard POD-based method has several shortcomings. First, the least-squares problem formulated in Eq. 7 can be underspecified. Thus, it is favorable to introduce some bias in order to reduce the variance by means of regularization. Ridge regularization is the most popular regularization technique for reducing the variance of the estimator:

$$\boldsymbol{\nu} \in \arg\min_{\widetilde{\boldsymbol{\nu}}} \ \|\boldsymbol{s} - \boldsymbol{H}\,\boldsymbol{\Phi}\,\widetilde{\boldsymbol{\nu}}\|_2^2 + \alpha\ \|\widetilde{\boldsymbol{\nu}}\|_2^2\ , \tag{11}$$

where $\alpha > 0$ is the penalization parameter. Typically, this parameter is determined by k-fold cross-validation. An alternative approach to reduce the variance is to select a subset of the POD modes, *i.e.*, only a few of the estimated coefficients are non-zero. The so-called Least Absolute Shrinkage and Selection Operator (LASSO) for least-squares [36, 37] can be formulated as:

$$\boldsymbol{\nu} \in \arg\min_{\widetilde{\boldsymbol{\nu}}} \ \|\boldsymbol{s} - \boldsymbol{H}\,\boldsymbol{\Phi}\,\widetilde{\boldsymbol{\nu}}\|_2^2 + \beta\ \|\widetilde{\boldsymbol{\nu}}\|_1\ , \tag{12}$$

where $\beta > 0$ controls the amount of sparsity. One can also combine both LASSO and ridge regularization, resulting in the so-called ElasticNet [38, 37] regularizer:

$$\boldsymbol{\nu} \in \arg\min_{\widetilde{\boldsymbol{\nu}}} \ \|\boldsymbol{s} - \boldsymbol{H}\,\boldsymbol{\Phi}\,\widetilde{\boldsymbol{\nu}}\|_2^2 + \alpha\ \|\widetilde{\boldsymbol{\nu}}\|_2^2 + \beta\ \|\widetilde{\boldsymbol{\nu}}\|_1\ . \tag{13}$$

This regularization scheme often shows an improved predictive performance in practice, however, it requires that the user fiddles around with two tuning parameters α and β.

Yet another approach is to use a shrinkage estimator that only retains the high variance POD modes, *i.e.*, an estimator that selects a subset of all the POD modes that is used for solving the least squares problem. More concretely, we formulate the following constrained problem:

$$\boldsymbol{\nu} \in \arg\min_{\widetilde{\boldsymbol{\nu}}} \ \|\boldsymbol{s} - \boldsymbol{H}\,\boldsymbol{\Phi}\,\widetilde{\boldsymbol{\nu}}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{\Phi}_{(n-k)}\widetilde{\boldsymbol{\nu}} = \boldsymbol{0}\,, \tag{14}$$

where $\boldsymbol{\Phi}_{(n-k)} = \{\boldsymbol{\phi}_{k+1}, \ldots, \boldsymbol{\phi}_n\}$. Here, $k \leq n$ refers to the number of selected POD modes, reordered with indices $\{1, 2, \ldots, k\}$. This hard threshold regularizer constraints the solution to the column space of the selected POD modes and is also known as Principal Component Regression (PCR) [37]. In contrast to the smooth shrinkage effect of ridge regularization, the hard threshold regularizer has a discrete shrinkage effect that nullifies the contributions of some of the low variance modes completely. However, based on our experiments, both ridge regression and the hard threshold shrinkage estimator perform on par for the task of flow field reconstruction. This said, the ElasticNet regularizer might lead to a better predictive accuracy, since it can select the POD modes that are most useful for prediction, rather than only selecting the high-variance POD modes. It is known that the POD modes with low variances may also be important for predictive tasks [39, 40] and could help to further improve the performance of the POD-based methods.

Another shortcoming of the POD-based approach is that it requires explicit knowledge of the observation operator $\boldsymbol{H}$ and is subjected to ill-conditioning of the least-squares problem. These limitations render this "vanilla flavored" approach often impractical in many situations, and they motivate an alternative formulation. The idea is to learn the map between coefficients and observations without explicitly referring to $\boldsymbol{H}$. It can be implicitly described by a, possibly nonlinear, operator $\boldsymbol{P} : \mathbb{R}^k \to \mathbb{R}^p$ typically determined offline by minimizing the Bayes risk, defined as the misfit in the L^2-sense:

$$\boldsymbol{P} \in \arg\min_{\widetilde{\boldsymbol{P}}} \ \mathbb{E}_{\mu_{\boldsymbol{s},\boldsymbol{\nu}}} \left[\left\| \boldsymbol{s} - \widetilde{\boldsymbol{P}}\,\boldsymbol{\nu} \right\|_2^2 \right], \tag{15}$$

where $\mu_{\boldsymbol{s},\boldsymbol{\nu}}$ is the joint probability measure of the observations $\boldsymbol{s}$ and the coefficients $\boldsymbol{\nu}$ obtained by projecting the field onto the (orthonormal) POD modes, $\boldsymbol{\nu} = \boldsymbol{\Phi}^\mathsf{T}\boldsymbol{x}$. This step only relies on information from the training set and is thus performed offline.

We assume the training set is representative of the underlying system, in the sense that it should contain independent samples drawn from the stationary distribution of the physical system at hand. The Bayes risk is then approximated by an empirical estimate, and the operator $\boldsymbol{P}$ is determined as

$$\boldsymbol{P} \in \arg\min_{\widetilde{\boldsymbol{P}}} \ \sum_{i=1}^{n} \left\| \boldsymbol{s}_i - \widetilde{\boldsymbol{P}}\,\boldsymbol{\nu}_i \right\|_2^2. \tag{16}$$

When the measurement operator $\boldsymbol{H}$ is linear, $\boldsymbol{P}$ is then an empirical estimate of $\boldsymbol{H}\,\boldsymbol{\Phi}$, the contribution of the basis modes $\{\boldsymbol{\phi}_j\}_j$ to the measurements $\boldsymbol{s}$. This formulation was already considered in our previous work, *e.g.*, [26], and brings flexibility in the properties of the map $\boldsymbol{P}$ compared to the closed-form solution in Eq. (8). For instance, regularization by sparsity can be enforced in $\boldsymbol{P}$, via L^0- or L^1-penalization. Expressing Eq. (16) in matrix form yields:

$$\boldsymbol{P} \in \underset{\widetilde{\boldsymbol{P}}\in\mathbb{R}^{p\times k}}{\arg\min} \left\| \boldsymbol{S} - \widetilde{\boldsymbol{P}}\,\boldsymbol{N} \right\|_F^2 , \tag{17}$$

where $\boldsymbol{S} \in \mathbb{R}^{p\times n}$ and $\boldsymbol{N} \in \mathbb{R}^{k\times n}$ respectively refers to the training data measurements $\{\boldsymbol{s}_i\}_i$ and coefficients $\{\boldsymbol{\nu}_i\}_i$. It immediatly follows

$$\boldsymbol{P} \;=\; \boldsymbol{S}\,\boldsymbol{N}^+ \;=\; \boldsymbol{S}\,\left(\boldsymbol{\Phi}^+\,\boldsymbol{X}\right)^+ \;=\; \boldsymbol{S}\boldsymbol{V}\boldsymbol{\Sigma}^+ , \tag{18}$$

and the online approximation obtained by POD PLUS is finally given by the solution to the following least-squares problem

$$\boldsymbol{\nu} \in \underset{\widetilde{\boldsymbol{\nu}}}{\arg\min} \; \|\boldsymbol{s} - \boldsymbol{P}\,\widetilde{\boldsymbol{\nu}}\|_2^2 \, . \tag{19}$$

However, $\boldsymbol{\nu} \in \mathbb{R}^k$ is typically higher-dimensional than $\boldsymbol{s} \in \mathbb{R}^p$, and thus the problem is ill-posed. We then make use of the popular Tikhonov regularization, selecting the solution with the minimum L^2-norm. This results in a ridge regression problem formulated as:

$$\boldsymbol{\nu} \in \underset{\widetilde{\boldsymbol{\nu}}}{\arg\min} \; \|\boldsymbol{s} - \boldsymbol{P}\,\widetilde{\boldsymbol{\nu}}\|_2^2 + \lambda \, \|\widetilde{\boldsymbol{\nu}}\|_2^2 , \tag{20}$$

with $\lambda > 0$. As will be seen in the examples below, penalization of the magnitude of the coefficients can significantly improve the performance of the POD approach.

2.2 Observability issue

The above techniques are standard in the scientific computing literature for flow reconstruction, but they bear a severe limitation. Indeed, since it is derived in an unsupervised fashion from the set of instances $\{\boldsymbol{x}_i\}_i$, the approximation basis $\{\boldsymbol{\phi}_j\}_j$ is *agnostic* to the measurements $\boldsymbol{s}$. In other words, the approximation basis is determined with no supervision by the measurements. To illustrate the impact of this situation, let $\boldsymbol{\nu}^\star = \boldsymbol{\Phi}^+\,\boldsymbol{x}$ be the least-squares estimate of the approximation coefficients for a given field $\boldsymbol{x}$. The difference between the least-square estimate coefficients $\boldsymbol{\nu}^\star$ and the coefficients $\boldsymbol{\nu}$ obtained from the linear sensor measurements $\boldsymbol{s}$ writes

$$\boldsymbol{\nu}^\star - \boldsymbol{\nu} = \left(\boldsymbol{\Phi}^+ - \left(\boldsymbol{H}\,\boldsymbol{\Phi}\right)^+ \boldsymbol{H}\right) \boldsymbol{x} , \tag{21}$$

and the error in the reconstructed field is obtained immediately:

$$\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\| = \left\| \left(\boldsymbol{I} - \boldsymbol{\Phi}\,\left(\boldsymbol{H}\,\boldsymbol{\Phi}\right)^+ \boldsymbol{H}\right) \boldsymbol{x} \right\| , \tag{22}$$

where $\boldsymbol{I}$ is the identity matrix of suitable dimension.

The error in the reconstructed field is seen to depend on both the approximation basis $\boldsymbol{\Phi}$ and the measurement operator $\boldsymbol{H}$. The measurement operator is entirely defined by the sensor locations, and it does not depend on the basis considered to approximate the field. Hence, to reduce (the expectation of) the reconstruction error, the approximation basis must be informed *both* by the dataset $\{\boldsymbol{x}_i\}_i$ *and* the sensors available, through $\boldsymbol{H}$. For example, poorly located sensors will lead to a large set of $\boldsymbol{x}_i$ to lie in the nullspace of $\boldsymbol{H}$, preventing their estimation, while the coefficients of certain approximation modes may be affected by the observation $\boldsymbol{H}\,\boldsymbol{x}_i$ of certain realizations $\boldsymbol{x}_i$ being severely amplified by $(\boldsymbol{H}\,\boldsymbol{\Phi})^+$ if the approximation basis is not carefully chosen.

This remark can be interpreted in terms of the control theory concept of *observability* of the basis modes by the sensors. Most papers in the literature focus their attention on deriving an approximation basis leading to a good representation [20, 21, 4], *i.e.*, such that the training set is well approximated in the k-dimensional basis $\{\boldsymbol{\phi}_j\}_j$, $\boldsymbol{x} \approx \boldsymbol{\Phi}\,\boldsymbol{\nu}$. But *how well* the associated coefficients $\boldsymbol{\nu} = \boldsymbol{\nu}\,(\boldsymbol{s})$ are informed by the measurements is usually overlooked when deriving the basis. In practice, the decoupling between learning an

approximation basis and learning a map to the associated coefficients often leads to a performance bottleneck in the estimation procedure. Enforcing observability of the approximation basis by the sensors is key to a good recovery performance and can dramatically improve upon unsupervised methods, as shown in [26].

3 Shallow neural networks for flow reconstruction

Shallow learning techniques are widely used for flow reconstruction. For instance, the approximation based approach for flow reconstruction, outlined in Section 2, can be considered to have two levels of complexity. The first level is concerned with computing an approximation basis, while the second level performs a linear weighted combination of the basis elements to estimate the high-dimensional flow field. Such shallow learning techniques are easy to train and tune. In addition, the levels are often physically meaningful, and they may provide some interesting insights into the underlying mechanics of the system under consideration.

In the following, we propose a simple SSN as an alternative to traditional methods, which are typically *very* shallow, for flow reconstruction problems. Our proposed shallow decoder adds only one or two additional layers of complexity to the problem.

3.1 A shallow decoder for flow reconstruction

We can define a fully-connected neural network (NN) with K layers as a nested set of functions

$$\mathcal{F}(\boldsymbol{s};\boldsymbol{W}) := R(\boldsymbol{W}^K R(\boldsymbol{W}^{K-1}\cdots R(\boldsymbol{W}^1\boldsymbol{s}))), \tag{23}$$

where $R(\cdot):\mathbb{R}\to\mathbb{R}$ denotes a coordinate-wise scalar (non-linear) activation function and $\boldsymbol{W}$ denotes a set of $\{\boldsymbol{W}^k\}_k$ weight matrices, $k=1,...,K$, with appropriate dimensions. NN-based learning provides a flexible framework for estimating the relationship between quantities from a collection of samples. Here, we consider SNNs, which are considered to be networks with very few, often only one, or even no, hidden layers, *i.e.*, K is very small.

In the following, an estimate of a vector $\boldsymbol{y}$ is denoted as $\widehat{\boldsymbol{y}}$, while $\widetilde{\boldsymbol{y}}$ denotes dummy vectors upon which one optimizes. Relying on a training set $\{\boldsymbol{x}_i,\boldsymbol{s}_i\}_{i=1}^n$, with n examples $\boldsymbol{x}_i$ and corresponding sensor measurements $\boldsymbol{s}_i$, we aim to learn a function $\mathcal{F}:\boldsymbol{s}\mapsto\widehat{\boldsymbol{x}}$ belonging to a class of neural networks $\mathscr{F}$ which minimizes the misfit in an Euclidean sense, over all sensor measurements

$$\mathcal{F} \in \underset{\widetilde{\mathcal{F}}\in\mathscr{F}}{\arg\min} \sum_{i=1}^{n} \left\| \boldsymbol{x}_i - \widetilde{\mathcal{F}}(\boldsymbol{s}_i) \right\|_2^2 . \tag{24}$$

We assume that only a small number of training examples is available. Further, no prior information is assumed to be available, and the estimation method is purely data-driven. Importantly, we assume no knowledge about the underlying measurement operator which is used to collect the sensor measurements. Further, unlike traditional methods for flow reconstruction, this NN-based learning methodology allows the joint learning of *both* the modes and the coefficients.

3.2 Architecture

We now discuss some general principles guiding the design of a good network architecture for flow reconstruction. These considerations lead to the following nested nonlinear function

$$\mathcal{F}(\boldsymbol{s}) = \boldsymbol{\Omega}(\boldsymbol{\nu}(\boldsymbol{\psi}(\boldsymbol{s}))) . \tag{25}$$

The architecture design is guided by the paradigm of simplicity. Indeed, the architecture should enable fast training, little tuning, and offer an intuitive interpretation.

Recall that the interpretability of the flow field estimate is favored by representing it in a basis of moderate size, whose modes can be identified with spatial structures of the field. This means, the estimate can be represented as a linear combination of k modes $\{\boldsymbol{\phi}_j\}_j$, weighted by coefficients $\{\nu_j\}_j$, see Eq. (4). These modes are a function of the inputs. This naturally leads to consider a network in which the output $\widehat{\boldsymbol{x}}$ is given

by a linear, fully connected, last layer of k inputs, interpreted as $\boldsymbol{\nu}$. These coefficients are informed by the sensor measurements $\boldsymbol{s}$ in a nonlinear way.

The nonlinear map $\boldsymbol{s} \mapsto \boldsymbol{\nu}$ can be described by a hidden layer, whose outputs $\boldsymbol{\psi}$ are hereafter termed *measurement features*, in analogy with kernel-based methods, where raw measurements $\boldsymbol{s}$ are nonlinearly lifted as extended measurements to a higher-dimensional space. In this architecture, the measurement features $\boldsymbol{\psi}$ essentially describe nonlinear combinations of the input measurement $\boldsymbol{s}$. The nonlinear combinations are then mapped to the coefficients $\boldsymbol{\nu}$ associated with the modes $\boldsymbol{\phi}$. While the size of the output layer is that of the discrete field $\boldsymbol{x}$, the size of the last hidden layer ($\boldsymbol{\nu}$) is *chosen* and defines the size k of the dictionary $\boldsymbol{\Phi}$. This size can be estimated from the data $\{\boldsymbol{x}_i\}_i$ by dimensionality estimation techniques [41, 42]. Restricting the description of the training data to a low-dimensional space is of potential interest to practitioners who may interpret the elements of the resulting basis in a physically meaningful way. The additional structure allows one to express the field of interest in terms of modes that practitioners may interpret, *i.e.*, relate to some physics phenomena such as traveling waves, instability patterns (*e.g.*, Kelvin-Helmholtz), etc.

In contrast, the size of the first hidden layer describing $\boldsymbol{\psi}$ is essentially driven by the size of the input layer ($\boldsymbol{s}$) and the number of nonlinear combinations used to nonlinearly inform the coefficients $\boldsymbol{\nu}$. The general shape of the network then bears flexibility in the hidden layers. A popular architecture for decoders consists of non decreasing layer sizes, so as to increase continuously the size of the representation from the low-dimensional observations to the high-dimensional field. We can model $\mathcal{F}$ as a shallow neural network with two hidden layers $\boldsymbol{\psi}$ and $\boldsymbol{\nu}$, followed by a linear output layer $\boldsymbol{\Omega}$.

Two types of hidden layers, namely fully-connected (FC) and convolution layers can be considered. The power of convolution layers is key to the success of recent deep learning architectures in computer vision. However, in our problem, we favor fully-connected layers. The reason are as follows: (i) our sensor measurements have no spatial ordering; (ii) depending on the number of filters, convolution layers require a large number of examples for training, while we assume that only a small number of examples are available for training; (iii) potential dynamical systems that we consider evolve on a curved domain which is typically represented using an unstructured grid. Thus, the first and second hidden layers take the form

$$\boldsymbol{z}^{\boldsymbol{\psi}} = \boldsymbol{\psi}(\boldsymbol{s}) := R(\boldsymbol{W}^{\boldsymbol{\psi}}\boldsymbol{s} + \boldsymbol{b}^{\boldsymbol{\psi}})\,,$$

and

$$\boldsymbol{z}^{\boldsymbol{\nu}} = \boldsymbol{\nu}(\boldsymbol{z}^{\boldsymbol{\psi}}) := R(\boldsymbol{W}^{\boldsymbol{\nu}}\boldsymbol{z}^{\boldsymbol{\psi}} + \boldsymbol{b}^{\boldsymbol{\nu}})\,,$$

where $\boldsymbol{W}$ denotes a dense weight matrix and $\boldsymbol{b}$ is a bias term. The function $R(\cdot)$ denotes an activation function used to introduce nonlinearity into the model as discussed below. The final linear output layer simply takes the form of

$$\widehat{\boldsymbol{x}} = \boldsymbol{\Omega}(\boldsymbol{z}^{\boldsymbol{\nu}}) := \boldsymbol{\Phi}\boldsymbol{z}^{\boldsymbol{\nu}} + \boldsymbol{b}^{\boldsymbol{\Phi}}\,,$$

where we interpret the columns of the weight matrix $\boldsymbol{\Phi}$ as modes. In summary, the architecture of our shallow decoder can be outlined as

$$\boldsymbol{s} \mapsto \boldsymbol{\psi}(\boldsymbol{s}) \mapsto \boldsymbol{\nu}(\boldsymbol{z}^{\boldsymbol{\psi}}) \mapsto \boldsymbol{\Omega}(\boldsymbol{z}^{\boldsymbol{\nu}}) \equiv \widehat{\boldsymbol{x}}\,.$$

Depending on the dataset, we need to adjust the size of each layer. Here, we use narrow rather than wide layers. Prescribing the size of the output layer restricts the dimension of the space in which the estimation lies, and it effectively regularizes the problem, *e.g.*, filtering-out most of the noise which is not living in a low-dimensional space.

The rectified linear unit (ReLU) activation function is among the most popular choices in computer vision applications, owing to its favorable properties [43]. The ReLU activation, illustrated in Figure 3a, is defined as the positive part of a signal $\boldsymbol{z}$:

$$R(\boldsymbol{z}) := \max(\boldsymbol{z}, \boldsymbol{0})\,. \tag{26}$$

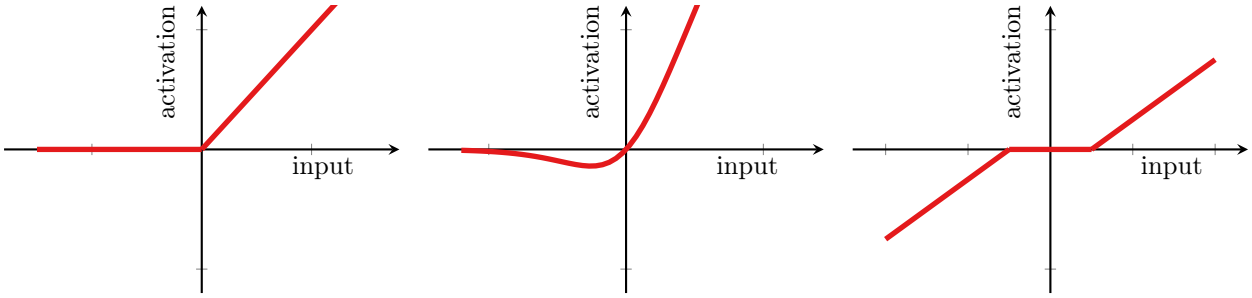
The transformed input signal is also called activation. While the ReLU activation function performs best on average in our experiments, there are other choices. For instance, we have considered the Swish [44] and SoftShrinkage activation function, also illustrated in Figure 3. These two activation functions can be fine-tuned via an additional hyper-parameter and there are potential situations in which these activation functions outperform ReLU. Interestingly, different activation functions considerably affect the modes (*i.e.*, columns of the weight matrix $\boldsymbol{\Phi}$), as shown in Figure 4.

3.3 Regularization

Overfitting is a common problem in machine learning and occurs if a function fits a limited set of data points too closely. In particular, this is a problem for deep neural networks which often have more neurons (trainable parameters) than can be justified by the limited amount of training examples which are available. There is increasing interest in characterizing and understanding generalization and overfitting in NNs [45, 46]. Hence, additional constraints are required to learn a function which generalizes to new observations that have not been used for training. Standard strategies to avoid overfitting include early stopping rules, and weight penalties (L^2 regularization) to regularize the complexity of the function (network). In addition to these two strategies, we use also batch normalization (BN) [47] and dropout layers (DL) [48] to improve the convergence and robustness of the shallow decoder. This yields the following architecture:

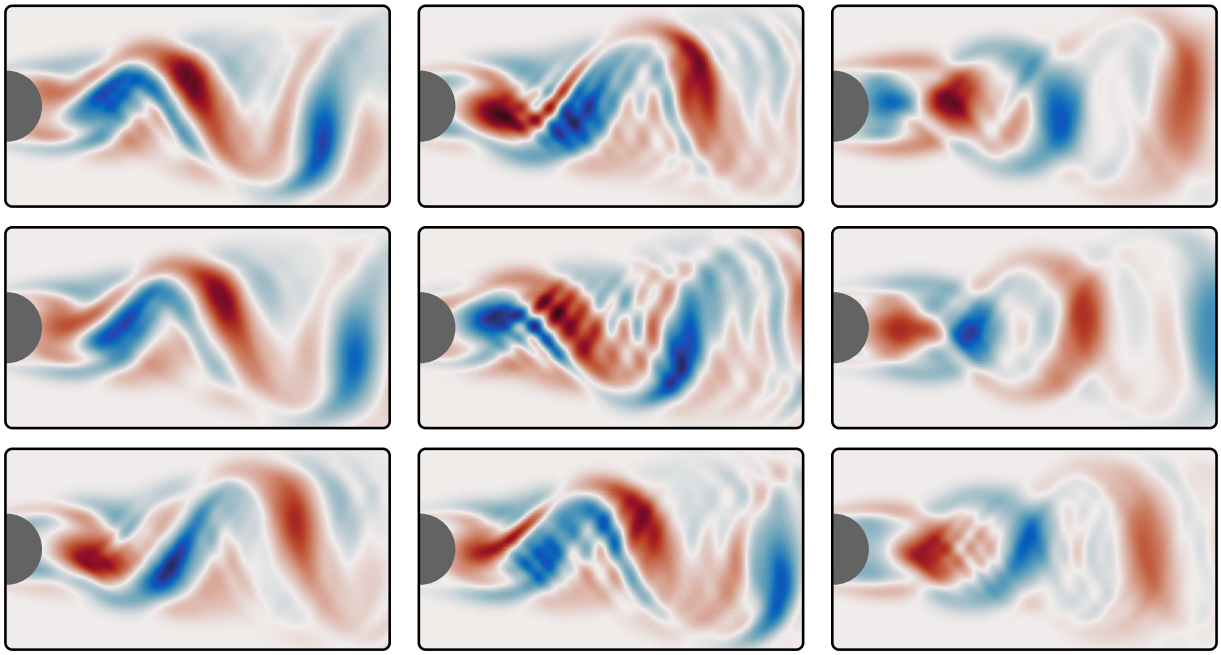
$$\boldsymbol{s} \mapsto \boldsymbol{\psi}(\boldsymbol{s}) \mapsto BN \mapsto DL \mapsto \boldsymbol{\nu}(\boldsymbol{z}^{\boldsymbol{\psi}}) \mapsto BN \mapsto \boldsymbol{\Omega}(\boldsymbol{z}^{\boldsymbol{\nu}}) \equiv \widehat{\boldsymbol{x}}\,.$$

Regularization, in its various forms, requires one to "fiddle" with a large number of knobs (*i.e.*, hyper-parameters). However, we have found that SNNs are less sensitive to the particular choice of parameters; hence, SNNs are easier to tune.



(a) ReLU (b) Swish (c) SoftShrinkage

Figure 3: Illustration of different activation functions.



(a) ReLU (b) Swish (c) SoftShrinkage

Figure 4: Dominant first three modes obtained by using different activation functions.

Batch normalization. BN is a technique to normalize (mean zero and unit standard deviation) the activation. From a statistical perspective, BN eases the effect of internal covariate shifts [47]. In other words, BN accounts for the change of distribution of the output signals (activation) across different mini batches during training. Each BN layer has two parameters which are learned during the training stage. This simple, yet effective, prepossessing step allows one to use higher learning rates for training the network. In addition it also reduces overfitting owing to its regularization effect.

Dropout layer. DL helps to improve the robustness of a NN. The idea is to switch off (drop) a small fraction of randomly chosen hidden units (neurons) during the training stage. This strategy can be seen as some form of regularization which also helps to reduce interdependent learning between the units of a fully connected layer. In our experiments the drop ratio is set to $p = 10\%$.

3.4 A note on overparameterized networks

The expressive power of NNs can be seen as a function of the depth (*i.e.*, number of hidden layers) and the width (*i.e.*, number of neurons per hidden layer) of the architecture [49]. Shallow networks typically tend to compensate for the reduced depth by increasing the width of the hidden layers. In turn, this can lead to shallow architectures that have more parameters than a comparable deep and narrow architecture for the same problem. However, such (potentially) overparameterized networks do not necessarily perform worse. On the contrary, recent theory suggests that it can be easier to train very overparameterized models with stochastic gradient descent (SGD) [50, 51].

This may be surprising, since conventional ML wisdom states that overparamerized models tend to overfit and show poor generalization performance. However, recent results show that overparamerized models trained to minimum norm solutions can indeed preserve the ability to generalize well [52, 53, 54, 55, 56].

3.5 Optimization

Given a training set with n targets $\{\boldsymbol{x}_i\}_i$ and corresponding sensor measurements $\{\boldsymbol{s}_i\}_i$, we minimize the misfit between the reconstructed quantity $\widehat{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{s})$ and the observed quantity $\boldsymbol{x}$, in terms of the squared L^2-norm

$$\mathcal{F} \in \arg\min_{\widetilde{\mathcal{F}}} \sum_{i=1}^{n} \left\| \boldsymbol{x}_i - \widetilde{\mathcal{F}}\left(\boldsymbol{s}_i\right) \right\|_2^2 + \lambda \|\boldsymbol{W}^i\|_2^2 .$$

The second term on the right hand side introduces L^2 regularization to the weight matrices, which is controlled via the parameter $\lambda > 0$. It is well-known that L^2-norm is sensitive to outliers; and the L^1-norm can be used as a more robust loss function. Alternatively, a popular option is the Huber norm (smooth L^1-loss), leading to the following optimization problem

$$\mathcal{F} \in \arg\min_{\widetilde{\mathcal{F}}} \sum_{i=1}^{n} \rho_{\mathrm{H}}(\boldsymbol{x}_i, \widetilde{\mathcal{F}}\left(\boldsymbol{s}_i\right); \kappa) ,$$

where

$$\rho_{\mathrm{H}}(\boldsymbol{x}, \widehat{\boldsymbol{x}}; \kappa) = \begin{cases} \kappa|\boldsymbol{x} - \widehat{\boldsymbol{x}}| - \kappa^2/2, & |\boldsymbol{x} - \widehat{\boldsymbol{x}}| > \kappa, \\ (\boldsymbol{x} - \widehat{\boldsymbol{x}})^2/2, & \text{otherwise} . \end{cases}$$

The tuning parameter κ controls the threshold. The Huber loss functions grow at a linear rate for residuals outside the thresholding parameter κ, rather than quadratically. This can reduce the influence of large deviations when learning the decoder. Further, it has been reported that this loss function prevents exploding gradients in some cases [57].

We use the ADAM optimization algorithm [58] to train the shallow decoder, with learning rate 10^{-2} and weight decay 10^{-4} (also known as L^2 regularization). The learning rate, also known as step size, controls how much we adjust the weights in each epoch. The weight decay parameter is important since it allows one to regularize the complexity of the network. In practice, we can improve the performance by changing the learning rate during training. We decay the learning rate by a factor of 0.9 after 100 epochs. Indeed, the reconstruction performance in our experiments is considerably improved by this dynamic scheme, compared

to a fixed parameter setting. In addition, we decrease the weight decay by a factor of 0.8. Further, we use a relatively large batch size, since we have only a limited amount of data available for training. In our experiments, ADAM shows a better performance than SGD with momentum [59] and averaged SGD [60]. The hyper-parameters can be fine tuned in practice, but our choice of parameters works reasonably well for several different examples. Note that we use the method described by [61] in order to initialize the weights. This initialization scheme is favorable, in particular because the output layer is high-dimensional.

4 Empirical evaluation

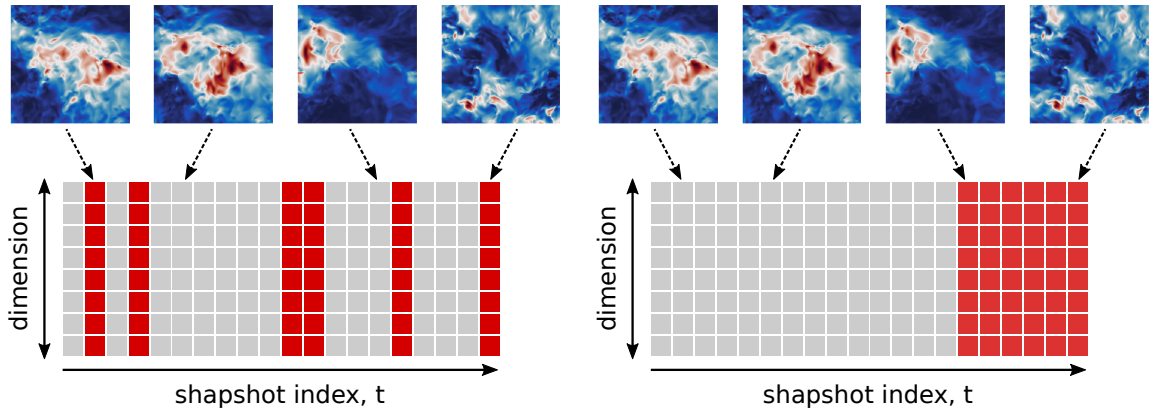
We evaluate our methods on three classes of data. First, we consider a periodic flow behind a circular cylinder, as a canonical example of fluid flow. Then, we consider the weekly mean sea surface temperature (SST), as a second and more challenging example. Finally, the third and most challenging example we consider is a forced isotropic turbulence flow.

As discussed in Section 1, the SHALLOW DECODER requires that the training data represent the system, in the sense that they should comprise samples drawn from the same statistical distribution as the testing data. Indeed, this limitation is standard to data-driven methods, both for flow reconstruction and also more generally. Hence, we are mainly concerned with exploring reconstruction performance and generalizability for *within sample prediction* rather than for *out of sample prediction* tasks. In our third example, however, we demonstrate the limitations of the SHALLOW DECODER, illustrating difficulties that arise when one tries to extrapolate, rather than interpolate, the flow field. Figure 5 illustrates the difference between the two types of tasks.

In the first two example classes of data, the sensor information is a subset of the high-dimensional flow field, *i.e.*, the measurement operator $\boldsymbol{H} \in \mathbb{R}^{p \times m}$ only has one non-zero entry in rows corresponding to the index of a sensor location. Letting $\mathcal{J} \in [1, m]^p \subset \mathbb{N}^p$ be the set of indices indexing the spatial location of the sensors, the measurement operator is such that

$$\boldsymbol{s} = \boldsymbol{H}\,\boldsymbol{x} = \boldsymbol{x}_{\mathcal{J}}\,, \tag{27}$$

that is, the observations are simply point-wise measurements of the field of interest. In the above equation, $\boldsymbol{x}_{\mathcal{J}}$ is the restriction of $\boldsymbol{x}$ to its entries indexed by $\mathcal{J}$. In this paper, no attempt is made to optimize the location of the sensors. In practical situations, they are often given or constrained by other considerations (wiring, intrusivity, manufacturing, etc.). We use random locations in our examples. The third example class of data demonstrates the SD using sub-gridscale measurements.



(a) Within sample prediction. (b) Out of sample prediction.

Figure 5: Two different training and test set configurations, showing (a) a within sample prediction task and (b) an out of sample prediction task. Here, the gray columns indicate snapshots used for training, while the red columns indicate snapshots used for testing.

The error is quantified in terms of the normalized root-mean-square residual error

$$\text{NME} = \frac{\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}\|_2}\,, \tag{28}$$

denoted in the following as "NME." However, this measure can be misleading if the empirical mean is dominating. Hence, we consider also a more sensitive measure which quantifies the reconstruction accuracy of the deviations around the empirical mean. We define this measure as

$$\text{NFE} = \frac{\|\boldsymbol{x}' - \widehat{\boldsymbol{x}}'\|_2}{\|\boldsymbol{x}'\|_2}\,, \tag{29}$$

where $\boldsymbol{x}'$ and $\widehat{\boldsymbol{x}}'$ are the fluctuating parts around the empirical mean. In our experiments, we average the errors over 30 runs for different sensor distributions.

4.1 Fluid flow behind the cylinder

The first example we consider is the fluid flow behind a circular cylinder, at Reynolds number 100, based on cylinder diameter, a canonical example in fluid dynamics [62]. The flow is characterized by a periodically shedding wake structure and exhibits smooth, large scale, patterns. A direct numerical simulation of the two-dimensional Navier-Stokes equations is achieved via the immersed boundary projection method [63, 64]. In particular, we use the fast multidomain method [64], which simulates the flow on five nested grids of increasing size, with each grid consisting of 199×449 grid points, covering a domain of 4×9 cylinder diameters on the finest domain. We collect 151 snapshots in time, sampled uniformly in time and covering several periods of vortex shedding. For the following experiment, we use cropped snapshots of dimension 199×384 on the finest domain, as we omit the spatial domain upstream to the cylinder. Further, we split the dataset into a training and test set so that the training set comprises the first 100 snapshots, while the remaining 51 snapshots are used for validation. Note that different splittings (interpolation and extrapolation) yield nearly the same results since the flow is periodic.

4.1.1 Varying numbers of random structured point-wise sensor measurements

We investigate the performance of the SHALLOW DECODER using varying numbers of sensors. A realistic setting is considered in that the sensors can only be located on a solid surface. The retained configuration aims at reconstructing the entire vorticity flow field from information at the cylinder surface only. The results are averaged over different sensor distributions on the cylinder downstream-facing surface and are summarized in Table 1. Further, to contextualize the precision of the algorithms, we also state the standard deviation in parentheses.

The SHALLOW DECODER shows an excellent flow reconstruction performance compared to traditional methods. Indeed, the results show that very few sensors are already sufficient to get an accurate approximation. Further, we can see that the SHALLOW DECODER is insensitive to the sensor location, *i.e.*, the variability of the performance is low when different sensor distributions on the cylinder surface are used. In stark contrast, this simple setup poses a challenge for the POD method without regularization, which is seen to be highly sensitive to the sensor configuration. This is expected since poorly located sensors lead to a large probability that the vorticity field $\boldsymbol{x}_i$ lies in the nullspace of $\boldsymbol{H}$, preventing its estimation, as discussed in Section 2. While regularization can improve the robustness slightly, the POD-based methods still require about at least 15 sensors to provide accurate estimations for the high-dimensional flow field. (Here, we list results for the POD method with hard-threshold regularization and POD PLUS method with ridge regularization. The number of retained components (hard-threshold), that were used for flow reconstruction, is indicated by k^* and the strength of ridge regularization is denoted by the parameter α. See Appendix A for more details.) In contrast, the SHALLOW DECODER exhibits a good performance with as few as 5 sensors. Note that the traditional methods could benefit from optimal sensor placement [4]; however, this is beyond the scope of this paper.

Figure 6 provides visual results for two specific sensor configuration using 5 sensors. The second configuration is challenging for POD, which fails to provide an accurate reconstruction. POD PLUS provides a more accurate reconstruction of the flow field. The SHALLOW DECODER outperforms the traditional methods in both situations.

4.1.2 Non-linear sensor measurements

So far, the sensor information consisted of pointwise measurements of the local flow field so that the j-th measurement is given by $\boldsymbol{s}^{(j)} = \boldsymbol{H}_j\, \boldsymbol{x} = \delta_{\tau_j}\, [\boldsymbol{x}] = \boldsymbol{x}^{(j)}$, $j = 1, \ldots, p$, with δ_{τ_j} a Dirac distribution centered at the location of the j-th sensor and $\boldsymbol{s}^{(j)}$ and $\boldsymbol{x}^{(j)}$ the j-th component of $\boldsymbol{s}$ and $\boldsymbol{x}$ respectively. We now consider *nonlinear* measurements to demonstrate the flexibility of the SHALLOW DECODER. Here, we consider the simple setting of squared sensor measurements: $\boldsymbol{s}^{(j)} = (\boldsymbol{x} \odot \boldsymbol{x})^{(j)}$, where $\odot$ denotes the Hadamard product. Table 2 provides a summary of the results, using 10 sensors. The SHALLOW DECODER is agnostic to the functional form of the sensor measurements, and it achieves nearly the same performance as in the linear

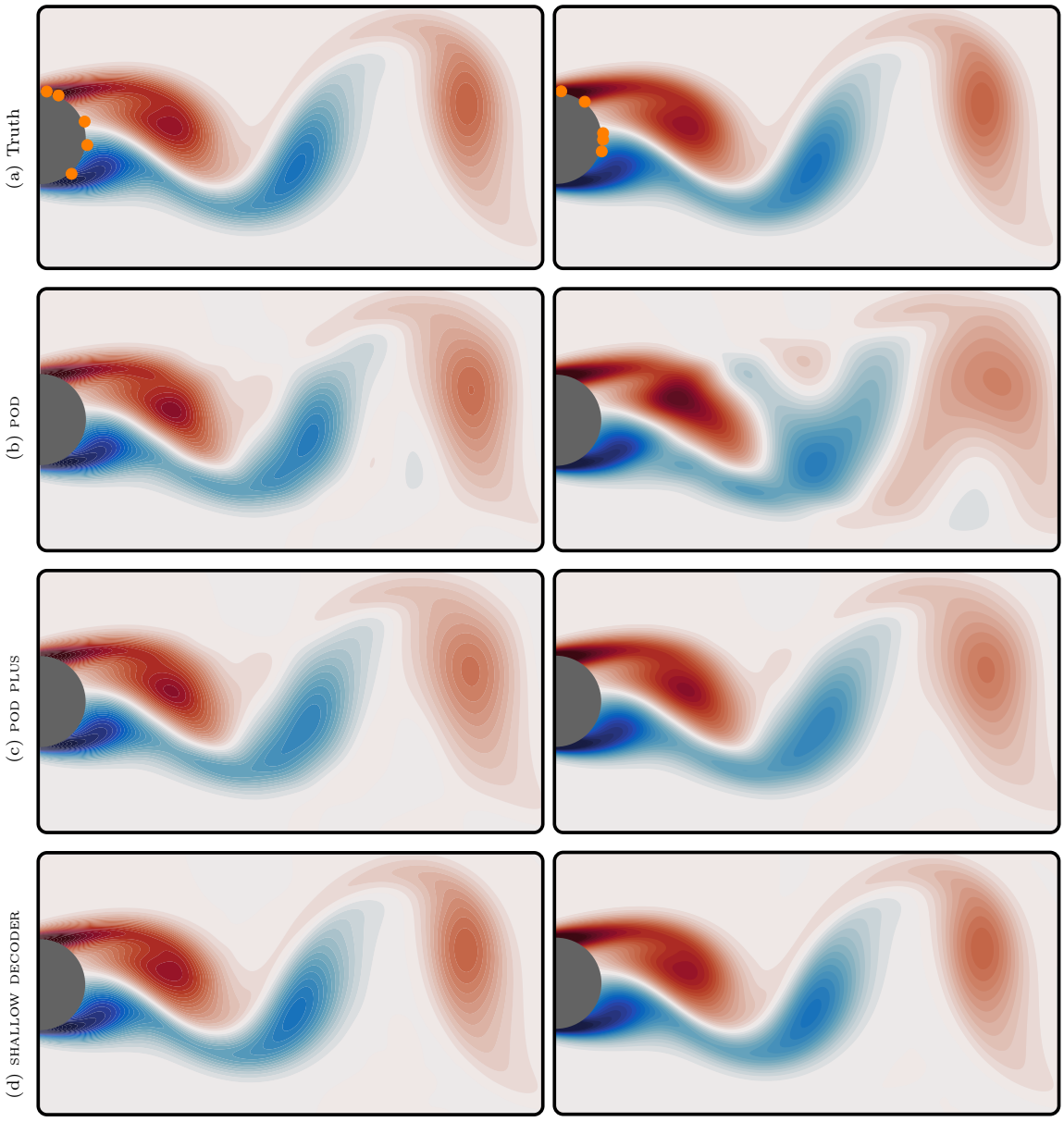


Figure 6: Visual results for the canonical flow for two different sensor distributions. In (a) the target snapshots and the specific sensor configurations (here using 5 sensors) are shown. Depending on the sensor distribution, the POD-based method is not able to accurately reconstruct the high-dimensional flow field, as shown in (b). The regularized POD PLUS method performs slightly better, as shown in (c). The SHALLOW DECODER yields an accurate flow reconstruction, as shown in (d).

case above. The average reconstruction accuracy for the test set increases only by about 1%.

	Sensors	Training Set		Test Set	
		NME	NFE	NME	NFE
POD	5	0.465 (0.39)	0.675 (0.57)	0.488 (0.41)	0.698 (0.59)
POD ($k^* = 4$)	5	0.217 (0.02)	0.325 (0.01)	0.227 (0.03)	0.324 (0.04)
POD PLUS ($\alpha =$ 1e-8)	5	0.198 (0.02)	0.288 (0.03)	0.203 (0.02)	0.291 (0.03)
SHALLOW DECODER	5	0.003 (0.00)	0.004 (0.00)	0.006 (0.00)	0.008 (0.00)
POD	10	0.346 (1.54)	0.502 (2.23)	0.379 (1.70)	0.542 (2.43)
POD ($k^* = 8$)	10	0.049 (0.00)	0.071 (0.01)	0.051 (0.01)	0.072 (0.01)
POD PLUS ($\alpha =$ 1e-13)	10	0.035 (0.01)	0.050 (0.02)	0.035 (0.01)	0.050 (0.02)
SHALLOW DECODER	10	0.002 (0.00)	0.003 (0.00)	0.005 (0.00)	0.007 (0.00)
POD	15	0.441 (1.81)	0.639 (2.63)	0.574 (2.44)	0.821 (3.49)
POD ($k^* = 12$)	15	0.015 (0.00)	0.023 (0.01)	0.016 (0.01)	0.023 (0.01)
POD PLUS ($\alpha =$ 1e-12)	15	0.016 (0.01)	0.023 (0.01)	0.016 (0.01)	0.022 (0.01)
SHALLOW DECODER	15	0.002 (0.00)	0.003 (0.00)	0.005 (0.00)	0.007 (0.00)

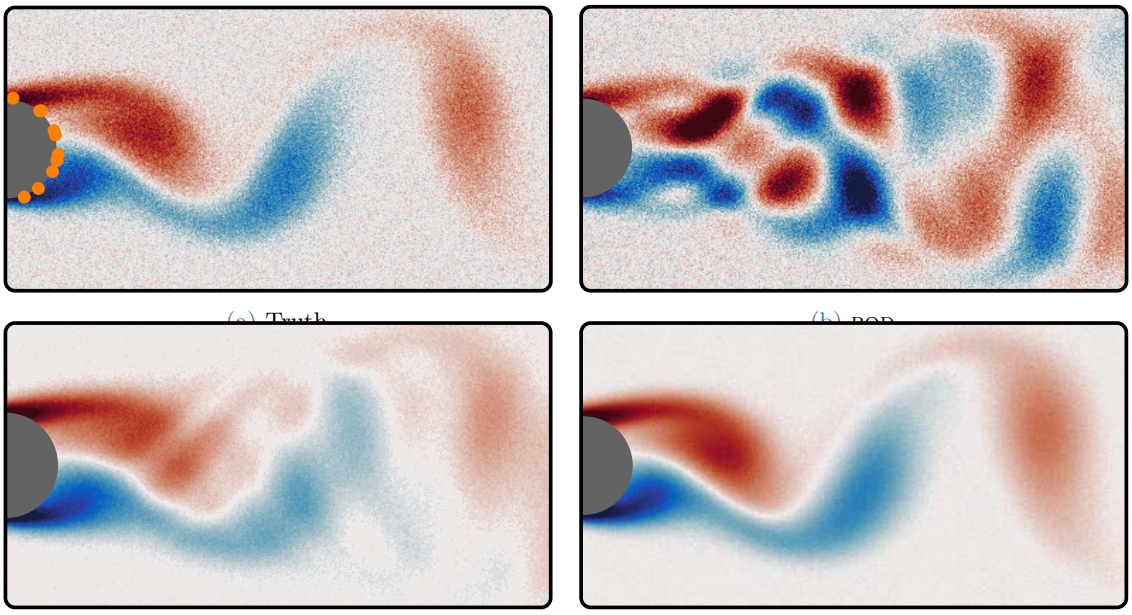
Table 1: Performance for the flow past cylinder for a varying number of sensors. Results are averaged over 30 runs with different sensor distributions, with standard deviations in parentheses. The parameter k^* indicates the number of modes that were used for flow reconstruction by the POD method, and α refers to the strength of ridge regularization applied to POD PLUS.

	Sensors	Training Set		Test Set	
		NME	NFE	NME	NFE
POD	10	-	-	-	-
POD PLUS ($\alpha =$ 5e-4)	10	0.676 (0.00)	0.981 (0.00)	0.682 (0.09)	0.974 (0.00)
SHALLOW DECODER	10	0.002 (0.00)	0.003 (0.00)	0.006 (0.00)	0.009 (0.01)

Table 2: Performance for estimating the flow behind a cylinder using nonlinear sensor measurements. The standard POD-based method fails for this task. POD PLUS is able to reconstruct the flow field, yet the estimation quality is poor. In contrast, the SD method performs well.

	SNR	Training Set		Test Set	
		NME	NFE	NME	NFE
POD	10	9.171 (14.7)	12.69 (20.4)	8.746 (12.9)	11.93 (17.6)
POD ($k^* = 2$)	10	0.461 (0.02)	0.638 (0.03)	0.468 (0.02)	0.639 (0.02)
POD PLUS ($\alpha =$ 5e-5)	10	0.468 (0.02)	0.648 (0.02)	0.472 (0.02)	0.644 (0.2)
SHALLOW DECODER	10	0.138 (0.02)	0.201 (0.02)	0.278 (0.04)	0.397 (0.05)
POD	50	4.837 (3.08)	6.946 (4.42)	4.520 (2.75)	6.390 (3.89)
POD ($k^* = 2$)	50	0.342 (0.01)	0.492 (0.01)	0.349 (0.01)	0.493 (0.01)
POD PLUS ($\alpha =$ 1e-5)	50	0.370 (0.03)	0.539 (0.04)	0.371 (0.02)	0.524 (0.03)
SHALLOW DECODER	50	0.134 (0.02)	0.198 (0.02)	0.173 (0.02)	0.247 (0.03)

Table 3: Performance for estimating the flow behind a cylinder in presence of white noise, using 10 sensors. POD fails for this task, while POD PLUS shows a better performance. The SD shows to be robust to noisy sensor measurements and outperforms the traditional techniques. The parameter k^* indicates the number of modes that were used for flow reconstruction by the POD method, and the parameter α refers to the strength of ridge regularization applied to the the POD PLUS method.



(c) POD PLUS (d) Shallow Decoder

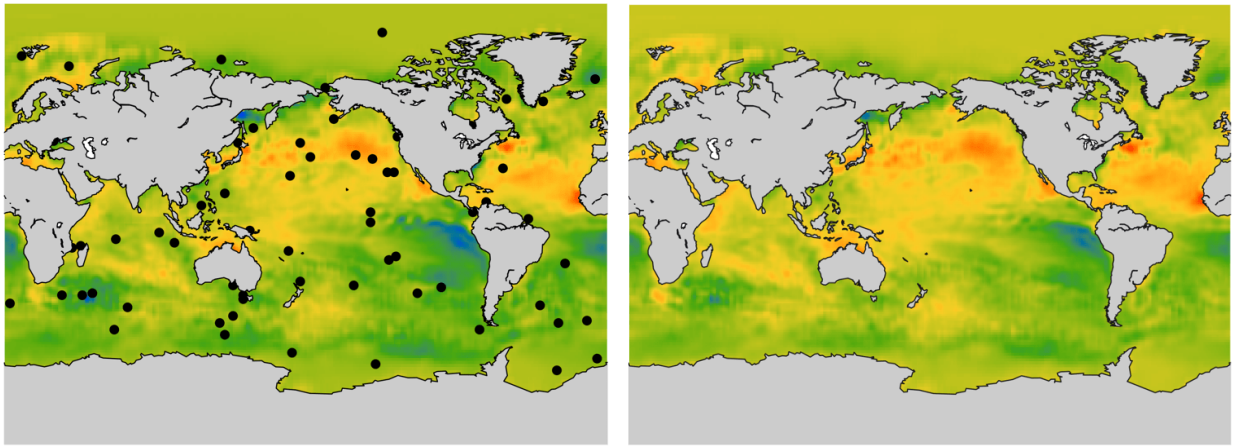
Figure 7: Visual results for the noisy flow behind the cylinder. Here the signal-to-noise ratio is 10. In (a) the target snapshot and the corresponding sensor configuration (using 10 sensors) is shown. Both, POD and POD PLUS are not able to reconstruct the flow field, as shown in (b) and (d). The SD is able to reconstruct the coherent structure of the flow field, as shown in (d).

4.1.3 Noisy sensor measurements

To investigate further the robustness and flexibility of the SHALLOW DECODER, we consider flow reconstruction in the presence of additive white noise. While this is not of concern when dealing with flow simulations, it is a realistic setting when dealing with flows obtained in experimental studies. Table 3 lists the results for both a high and low noise situation with linear measurements. By inspection, the performance of the SHALLOW DECODER outperforms classical techniques. In the high noise case, with a signal-to-noise ratio (SNR) of 10, the average relative reconstruction error for the test set is about 27% for the SHALLOW DECODER. For a SNR of 50, the relative error is as low as 17%. Note that we here use an additional dropout layer (placed after the first fully-connected layer) to improve the robustness of the SHALLOW DECODER. In contrast, standard POD fails in both situations. Again, the POD PLUS method shows improved results over the standard POD. However, the visual results in Figure 7 show that the reconstruction quality of the SHALLOW DECODER is favorable. The SHALLOW DECODER shows a clear advantage and a denoising effect. Indeed the reconstructed snapshots allow for a meaningful interpretation of the underlying structure.

4.1.4 Summary of empirical results for the flow behind the cylinder

The empirical results show that the advantage of the SHALLOW DECODER compared to the traditional POD based techniques is pronounced, even for a simple problem such as the flow behind the cylinder. It can be seen, that the performance of the traditional techniques is patchy, *i.e.*, the reconstruction quality is highly sensitive to the sensor location. While regularization can mitigate a poor sensor placement design, a relatively larger number (> 15) of sensors is required in order to achieve an accurate reconstruction performance. More challenging situations such as nonlinear measurements and sensor noise pose a challenge for the traditional techniques, while the SHALLOW DECODER shows to be able to reconstruct dominant flow features in such situations. The computational demands required to train the SHALLOW DECODER are minimal, *e.g.*, the time for training on a modern GPU remains below two minutes for this example.



(a) Truth (b) Shallow Decoder

Figure 8: Visual results for the SST dataset. In (a), the high-dimensional target and the sensor configurations (using 64 sensors) are shown; and in (b), the results of the Shallow Decoder are shown. Note that we show here the mean centered snapshot. The SHALLOW DECODER shows an excellent reconstruction quality for the fluctuations around the mean with an error as low as 12%.

	Sensors	Training Set		Test Set	
		NME	NFE	NME	NFE
POD	32	0.637 (0.59)	5.915 (5.56)	0.649 (0.62)	6.04 (5.77)
POD ($k^* = 5$)	32	0.036 (0.00)	0.342 (0.01)	0.037 (0.00)	0.344 (0.01)
POD PLUS (α = 1e-5)	32	0.036 (0.00)	0.341 (0.01)	0.037 (0.00)	0.343 (0.01)
SHALLOW DECODER	32	0.009 (0.00)	0.088 (0.00)	0.014 (0.00)	0.128 (0.00)
POD	64	0.986 (1.34)	9.183 (12.5)	1.007 (1.36)	9.344 (12.7)
POD ($k^* = 14$)	64	0.032 (0.00)	0.298 (0.01)	0.032 (0.00)	0.301 (0.01)
POD PLUS (α = 5e-5)	64	0.032 (0.00)	0.301 (0.00)	0.032 (0.00)	0.301 (0.00)
SHALLOW DECODER	64	0.009 (0.00)	0.085 (0.00)	0.012 (0.00)	0.118 (0.00)

Table 4: Performance for estimating the SST dataset for varying numbers of sensors. The SD outperforms the traditional techniques and shows to be highly invariant to the sensor location. The parameter k^* indicates the number of modes that were used for flow reconstruction by the POD method, and α refers to the strength of ridge regularization applied to POD PLUS.

4.2 Sea surface temperature using random point-wise measurements

The second example we consider is the more challenging sea surface temperature (SST) dataset. Complex ocean dynamics lead to rich flow phenomena, featuring interesting seasonal fluctuations. While the mean SST flow field is characterized by a periodic structure, the flow is non-stationary. The dataset consists of the weekly sea surface temperatures for the last 26 years, publicly available from the National Oceanic & Atmospheric Administration (NOAA).The data comprise 1483 snapshots in time with spatial resolution of 180×360. For the following experiments, we only consider $44,219$ measurements, by excluding measurements corresponding to the land masses. Further, we create a training set by selecting 1100 snapshots at random, while the remaining snapshots are used for validation.

We consider the performance of the SHALLOW DECODER using varying numbers of random sensors scattered across the spatial domain. The results are summarized in Table 4. We observe a large discrepancy between the NME and NFE error. This is because the long-term annual mean field accounts for the majority of the spatial structure of the field. Hence, the NME error is uninformative with respect to the performance of reconstruction methods. In terms of the NFE error the POD based reconstruction techniques is shown to fail

to reconstruct the high-dimensional flow field using limited sensor measurements. In contrast, the SHALLOW DECODER demonstrates an excellent reconstruction performance both using 32 and 64 measurements. Figure 8 shows visual results to support these quantitative findings.

4.3 Turbulent flow using sub-gridscale measurements

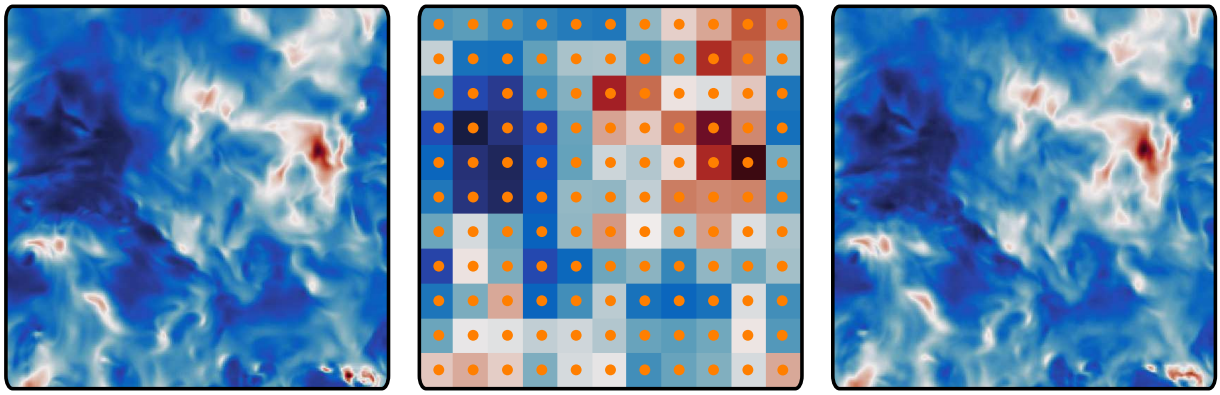
The final example we consider is the velocity field of a turbulent isotropic flow. Unlike the previous examples, the isotropic turbulent flow is non-periodic in time and highly non-stationary. Thus, this dataset poses a challenging task. Here, we consider data from a forced isotropic turbulence flow generated with a direct numerical simulation using $1,024^3$ points in a triply periodic $[0, 2\pi]^3$ domain. For the following experiments, we are using 800 snapshots for training and 200 snapshots for validation. The data spread across about one large-eddy turnover time. The data is provided as part of the Johns Hopkins Turbulence Database [65].

If the sensor measurements $\boldsymbol{s}$ are acquired on a coarse but regular grid, then the reconstruction task may be considered as a *super-resolution* problem [66, 67, 3]. There are a number of direct applications of super-resolution in fluid mechanics centered around sub-gridscale modeling. Because many fluid flows are inherently multiscale, it may be prohibitively expensive to collect data that captures all spatial scales, especially for iterative optimization and real-time control [1]. Inferring small-scale flow structures below the spatial resolution available is an important task in large eddy simulation (LES), climate modeling, and particle image velocimetry (PIV), to name a few applications. Deep learning has recently been employed for super-resolution in fluid mechanics applications with promising results [12]. Note that our setting differs from the super-resolution problem. Here, we obtain first a low-resolution image by applying a mean filter to the high-dimensional snapshot. Then, we use a single sensor measurement per grid cell to form the inputs (illustrated in Figure 9b). In contrast, super-resolution uses the low-resolution image as input.

First, we consider the within sample prediction task. In this case, we yield excellent results for the estimated high-dimensional flow fields, despite the challenging problem. Table 5 quantifies the performance for varying numbers of sub-gridscale measurements. In addition, Figure 9 provides some visual evidence for the good performance for this problem.

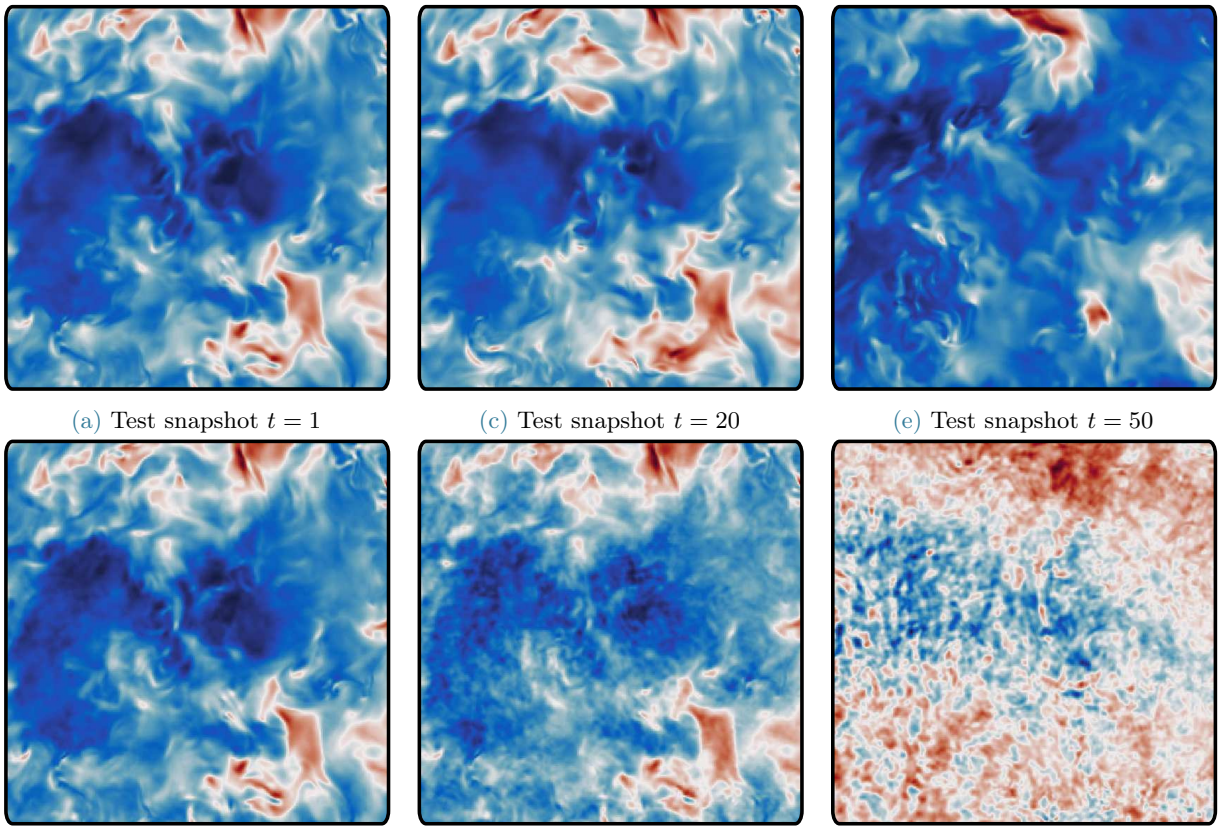
	Grids	Training Set		Test Set	
		NME	NFE	NME	NFE
SHALLOW DECODER	36	0.029 (0.00)	0.041 (0.00)	0.071 (0.00)	0.101 (0.01)
SHALLOW DECODER	64	0.027 (0.00)	0.039 (0.00)	0.067 (0.00)	0.096 (0.00)
SHALLOW DECODER	121	0.026 (0.00)	0.038 (0.00)	0.066 (0.00)	0.093 (0.00)

Table 5: Flow reconstruction performance for estimating the isotropic flow.



(a) Snapshot (b) Low resolution (c) Shallow Decoder

Figure 9: Visual results for the turbulent isotropic flow using 121 subgrid-cell measurements. The interpolation error of the SHALLOW DECODER error is about 9.3%.



(a) Test snapshot $t = 1$ (c) Test snapshot $t = 20$ (e) Test snapshot $t = 50$

(b) Reconstruction (d) Reconstruction (f) Reconstruction

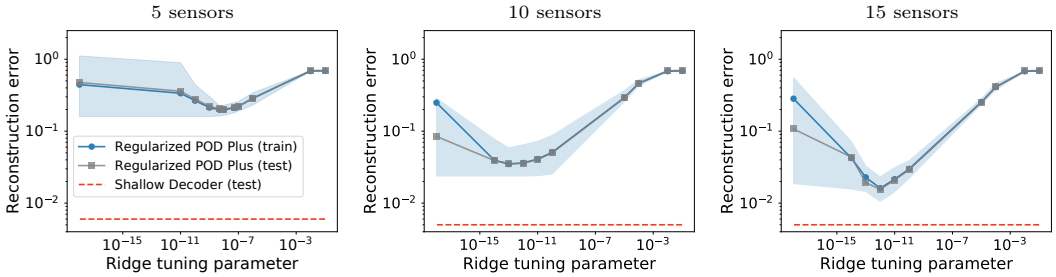
Figure 10: Visual results illustrating the limitation of the SHALLOW DECODER for extrapolation tasks. Flow fields sampled from or close to the statistical distribution describing the training examples can be reconstructed with high accuracy, as shown in (a) and (b). Extrapolation fails for fields which belong to a different statistical distribution, as shown in (e) and (f).

Next, we illustrate the limitation of the SHALLOW DECODER. Indeed, it is important to stress that the SD cannot be used for "out of sample prediction tasks" if the fluid flow is highly non-stationary. To illustrate this issue, Figure 10 shows three flow fields at different temporal locations. First, Figure 10a shows a test example, which is close in time to the training set. In this case, the SD is able to reconstruct the flow field with high accuracy. The reconstruction quality drops for snapshots which are further away in time, as shown in Figure 10c. Finally, Figure 10e shows that reconstruction fails if the test example is far away from the training set in time, *i.e.*, the flow field is not drawn from the same statistical distribution as the training examples are.

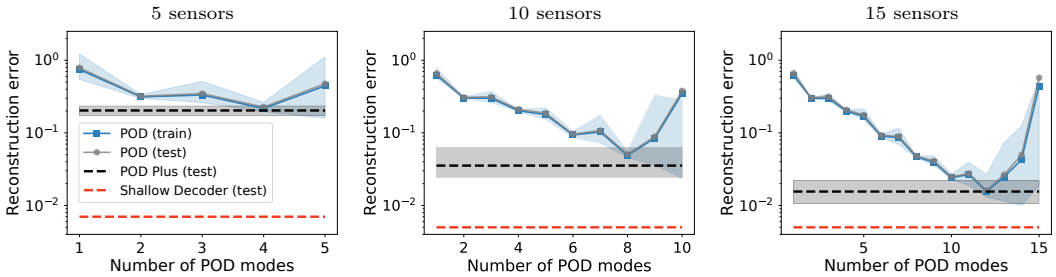
5 Discussion

The emergence of sensor networks for global monitoring (*e.g.*, ocean and atmospheric monitoring) requires new mathematical techniques that are capable of maximally exploiting sensors for state estimation and forecasting. Emerging algorithms from the machine learning community can be integrated with many traditional scientific computing approaches to enhance sensor network capabilities. For many global monitoring applications, the placement of sensors can be prohibitively expensive, thus requiring learning techniques such as the one proposed here, which can exploit a reduction in the number of sensors while maintaining required performance characteristics.

To partially address this challenge, we proposed a SHALLOW DECODER with two hidden layers for the problem of flow reconstruction. The mathematical formulation presented is significantly different from what



(a) POD Plus with ridge regularization.



(b) POD with hard-threshold regularization.

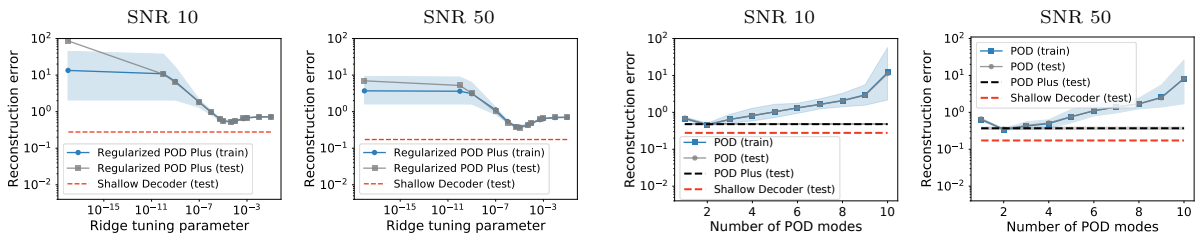
Figure 11: Results of the hyper-parameter search for the flow past the cylinder. The results for the POD-PLUS method, using ridge regularization, are shown in (a), and the results for the POD method, using hard-threshold, are shown in (b). The shallow decoder outperforms the POD-based methods in all situations, while the performance gap closes for an increased number of sensors.

is commonly used in flow reconstruction problems, *e.g.*, gappy interpolation with dominant POD modes. Indeed, our experiments demonstrate the improved the enhanced robustness and accuracy of fluid flow field reconstruction by using our SHALLOW DECODER.

Future work aims to leverage the underlying laws of physics in flow problems to further improve the efficiency. In the context of flow reconstruction or, more generally, observation of a high-dimensional physical system, insights from the physics at play can be exploited [68]. In particular, the dynamics of many systems do indeed remain low-dimensional and the trajectory of their state vector lies close to a manifold whose dimension is significantly lower than the ambient dimension. Moreover, the features exploited from the shallow decoder network can also be integrated in reduced order models (ROMs) for forecasting predictions [69]. In many high-dimensional systems where ROMs are used, the ability to generate low-fidelity models that can be rapidly simulated has revolutionized our ability to model such complex systems, especially in application of complex flow fields. The ability to rapidly generate low-rank feature spaces alternative to POD generates new possibilities for ROMs using limited sampling and limited data. This aspect of the SHALLOW DECODER will be explored further in future work.

Acknowledgments

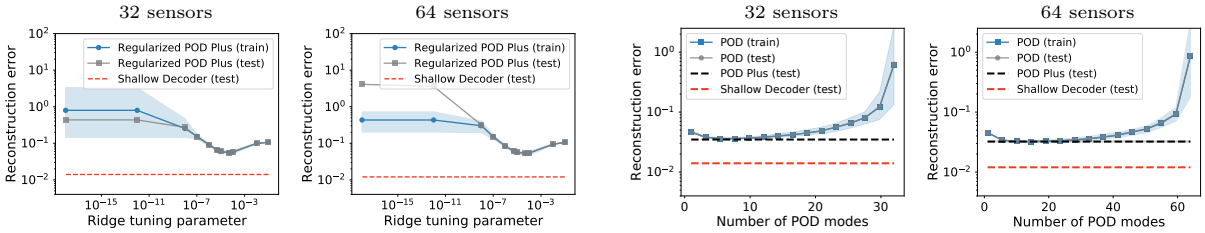
LM gratefully acknowledges the support of the French Agence Nationale pour la Recherche (ANR) and Direction Générale de l'Armement (DGA) via the *FlowCon* project (ANR-17-ASTR-0022). SLB acknowledges support from the Army Research Office (ARO W911NF-17-1-0422). JNK acknowledges support form the Air Force Office of Scientific Research (FA9550-19-1-0011). LM and JNK also acknowledge support from the Air Force Office of Scientific Research (FA9550-17-1-0329). MWM would like to acknowledge ARO, DARPA, NSF, and ONR for providing partial support for this work. We would also like to thank Kevin Carlberg for valuable discussions about flow reconstruction techniques.



(a) POD Plus with ridge regularization.

(b) POD with hard-threshold regularization.

Figure 12: Results of the hyper-parameter search for the noisy flow past the cylinder. Here we consider the signal-to-noise (SNR) ratios 10 and 50. Here we consider a setting with 10 sensors.



(a) POD Plus with ridge regularization.

(b) POD with hard-threshold regularization.

Figure 13: Results of the hyper-parameter search for the SST data. Here we consider a setting with 32 and 64 sensors. The shallow decoder outperforms the POD-based methods in all situations.

A Hyper-parameter search for the POD based methods

In the following, we provide results of our hyper-parameter search for determining the optimal tuning parameters for flow reconstruction. We proceed by evaluating the reconstruction error of the POD and POD-PLUS method for a plausible range of values. Here, we consider hard-threshold regularization for the POD method and ridge regularization for the POD PLUS method. We run 30 trails of the experiment, where we use a unique sensor location configuration at each trial.

Figure 11 shows the results for the fluid flow past the cylinder. First, we show the results for the POD PLUS method in (a). Regularizing the solution improves the reconstruction accuracy and the effect of regularization on the reconstruction error is pronounced for an increasing number of sensors. (Note, that at the same time, while the reconstruction error is decreasing with an increasing numbers of sensors, finding the optimal tuning parameter becomes more difficult.) Next, we show the results for POD with hard-threshold regularization in (b). It can be seen, that the performance is on par with ridge regularization (plotted as black dashed line), where hard-threshold regularization shows to have a lower variance compared to ridge regularization. In contrast, the shallow decoder outperforms both the POD and the POD PLUS method, represented by a dashed read line. However, the performance gap between the POD-based methods and the shallow decoder is closing for an increased number of sensors. This is not surprising, since the flow past the cylinder represents a relatively simple problem where the POD method is known to provide good reconstruction results, given a sufficient large number of sensors.

Figure 12 and 13 show the results for the noisy flow past the cylinder and for the sea surface temperature data. Again, it can be seen that both ridge regularization and hard-threshold regularization performs on par, while the shallow decoder outperforms the POD-based methods.

B Setup for our empirical evaluation

Here, we provide details about the concrete network architectures of the shallow decoder, which are used for the different examples. The networks are implemented in Python using PyTorch; and research code for flow behind the cylinder is available via `https://github.com/erichson/ShallowDecoder`. Tables 6– 8 show the details. For each example we use a similar architecture design. The difference is that we use a slightly wider

design (more neurons per layer) for the SST dataset and the isotropic flow. That is because we are using a larger number of sensors for these two problems, and thus we need to increase the capacity of the network. In each situation, the learning rate is set to 1e-2 with a scheduled decay rate of 0.3. Further, we use a small amount of weight decay $\lambda =$ 1e-7 to regularize the network.

Layer	Weight size	Input Shape	Output Shape	Activation	Batch Norm.	Dropout
FC	sensors $\times$ 35	sensors	35	ReLU	True	-
FC	35 $\times$ 40	25	40	ReLU	True	-
FC	40 $\times$ 76,416	40	76,416	Linear	-	-

Table 6: Architecture of the SD for the flow behind the cylinder. The batch size is set to 32. Here, we set the dropout rate to 0.1 for the noisy situation. We use a small amount of weight decay $\lambda =$ 1e-7.

Layer	Weight size	Input Shape	Output Shape	Activation	Batch Norm.	Dropout
FC	sensors $\times$ 350	sensors	350	ReLU	True	0.1
FC	350 $\times$ 400	350	400	ReLU	True	-
FC	400 $\times$ 44,219	400	44,219	Linear	-	-

Table 7: Architecture of the SD for the SST dataset. Here, the batch size is set to 200.

Layer	Weight size	Input Shape	Output Shape	Activation	Batch Norm.	Dropout
FC	sensors $\times$ 350	sensors	350	ReLU	True	0.1
FC	350 $\times$ 400	350	400	ReLU	True	-
FC	400 $\times$ 122,500	400	122,500	Linear	-	-

Table 8: Architecture of the SD for isotropic flow. Here, the batch size is set to 200.

References

[1] Brunton SL, Noack BR. 2015 Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews* **67**, 050801–1–050801–48.

[2] Rowley CW, Dawson ST. 2017 Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics* **49**, 387–417.

[3] Callaham J, Maeda K, Brunton SL. 2018 Robust reconstruction of flow fields from limited measurements. *arXiv preprint arXiv:1810.06723*.

[4] Manohar K, Brunton BW, Kutz JN, Brunton SL. 2018 Data-Driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems* **38**, 63–86.

[5] Yu J, Hesthaven JS. 2018 Flowfield reconstruction method using artificial neural network. *AIAA Journal* pp. 1–17.

[6] Bolton T, Zanna L. 2019 Applications of deep learning to ocean data inference and subgrid parameterization. *Journal of Advances in Modeling Earth Systems* **11**, 376–399.

[7] McCann MT, Jin KH, Unser M. 2017 A review of convolutional neural networks for inverse problems in imaging. *arXiv preprint arXiv:1710.04011*.

[8] Zhou YT, Chellappa R, Vaid A, Jenkins BK. 1988 Image restoration using a neural network. *IEEE Trans. Acous., Speech, & Sig. Proc.* **36**, 1141–1151.

[9] Ling J, Kurzawski A, Templeton J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics* **807**, 155–166.

[10] Kim B, Azevedo VC, Thuerey N, Kim T, Gross M, Solenthaler B. 2018 Deep fluids: A Generative Network for Parameterized Fluid Simulations. *arXiv preprint arXiv:1806.02071*.

[11] Carlberg KT, Jameson A, Kochenderfer MJ, Morton J, Peng L, Witherden FD. 2018 Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning. *arXiv preprint arXiv:1812.01177*.

[12] Fukami K, Fukagata K, Taira K. 2018 Super-resolution reconstruction of turbulent flows with machine learning. *arXiv preprint arXiv:1811.11328*.

[13] Vlachas PR, Byeon W, Wan ZY, Sapsis TP, Koumoutsakos P. 2018 Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **474**, 20170844.

[14] Mousavi A, Baraniuk RG. 2017 Learning to invert: Signal recovery via deep convolutional networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* pp. 2272–2276.

[15] Jin KH, McCann MT, Froustey E, Unser M. 2017 Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing* **26**, 4509–4522.

[16] Adler J, Öktem O. 2017 Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems* **33**, 124007.

[17] Ye JC, Han Y, Cha E. 2018 Deep convolutional framelets: A general deep learning framework for inverse problems. *SIAM J. Imag. Sci.* **11**, 991–1048.

[18] Chaturantabut S, Sorensen DC. 2010 Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing* **32**, 2737–2764.

[19] Drmac Z, Gugercin S. 2016 A new selection operator for the discrete empirical interpolation method—Improved a priori error bound and extensions. *SIAM Journal on Scientific Computing* **38**, A631–A648.

[20] Bui-Thanh T, Damodaran M, Willcox KE. 2004 Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA journal* **42**, 1505–1516.

[21] Willcox K. 2006 Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & Fluids* **35**, 208–226.

[22] Everson R, Sirovich L. 1995 Karhunen–Loeve procedure for gappy data. *JOSA A* **12**, 1657–1664.

[23] Candès EJ, Romberg J, Tao T. 2006 Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* **52**, 489–509.

[24] Donoho DL. 2006 Compressed Sensing. *IEEE Transactions on Information Theory* **52**, 1289–1306.

[25] Baraniuk RG. 2007 Compressive Sensing. *IEEE Signal Processing Magazine* **24**, 118–121.

[26] Mathelin L, Kasper K, Abou-Kandil H. 2017 Observable dictionary learning for high-dimensional statistical inference. *Archives Comput. Meth. Eng.* **25**, 103–120. ArXiv 1702.05289.

[27] Barrault M, Maday Y, Nguyen NC, Patera AT. 2004 An "empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* **339**, 667–672.

[28] Mahoney MW. 2011 Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning* **3**, 123–224.

[29] Drineas P, Mahoney MW. 2016 RandNLA: Randomized Numerical Linear Algebra. *Communications of the ACM* **59**, 80–90.

[30] Halko N, Martinsson PG, Tropp JA. 2011 Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* **53**, 217–288.

[31] Erichson NB, Voronin S, Brunton SL, Kutz JN. 2016 Randomized matrix decompositions using R. *arXiv preprint arXiv:1608.02148*.

[32] Erichson NB, Zeng P, Manohar K, Brunton SL, Kutz JN, Aravkin AY. 2018 Sparse principal component analysis via variable projection. *arXiv preprint arXiv:1804.00341*.

[33] Erichson NB, Mathelin L, Kutz JN, Brunton SL. 2019 Randomized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems* **18**, 1867–1891.

[34] Murray NE, Ukeiley L. 2007 An application of Gappy POD. For subsonic cavity flow PIV data. *Experiments in Fluids* **42**.

[35] Podvin B, Fraigneau Y, Lusseyran F, Gougat P. 2005 A reconstruction method for the flow past an open cavity. *Journal of Fluids Engineering* **128**.

[36] Tibshirani R. 1996 Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological)* **58**, 267–288.

[37] Hastie T, Tibshirani R, Friedman J. 2009 *The elements of statistical learning: Data mining, inference, and prediction*. Springer Science.

[38] Zou H, Hastie T. 2005 Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society:Series B (Statistical Methodology)* **67**, 301–320.

[39] Frank LE, Friedman JH. 1993 A statistical view of some Chemometrics regression tools. *Technometrics* **35**, 109–135.

[40] Jolliffe IT. 1982 A note on the use of principal components in regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **31**, 300–303.

[41] Granata D, Carnevale V. 2016 Accurate estimation of the intrinsic dimension using graph distances: unraveling the geometric complexity of datasets. *Scientific Reports* **6**.

[42] Facco E, d'Errico M, Rodriguez A, Laio A. 2017 Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports* **7**.

[43] Glorot X, Bordes A, Bengio Y. 2011 Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* pp. 315–323.

[44] Agostinelli F, Hoffman M, Sadowski P, Baldi P. 2014 Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*.

[45] Poggio T, Kawaguchi K, Liao Q, Miranda B, Rosasco L, Boix X, Hidary J, Mhaskar H. 2017 Theory of deep learning III: Explaining the non-overfitting puzzle. *arXiv preprint arXiv:1801.00173*.

[46] Bartlett PL, Foster DJ, Telgarsky MJ. 2017 Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems* pp. 6240–6249.

[47] Ioffe S, Szegedy C. 2015 Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[48] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. 2014 Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15**, 1929–1958.

[49] Lu Z, Pu H, Wang F, Hu Z, Wang L. 2017 The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems* pp. 6231–6239.

[50] Keskar NS, Mudigere D, Nocedal J, Smelyanskiy M, Tang PTP. 2016 On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

[51] Allen-Zhu Z, Li Y, Liang Y. 2019 Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems* pp. 6155–6166.

[52] Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. 2016 Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

[53] Hardt M, Recht B, Singer Y. 2016 Train faster, generalize better: Stability of stochastic gradient descent. In Balcan MF, Weinberger KQ, editors, *Proceedings of The 33rd International Conference on Machine Learning* vol. 48*Proceedings of Machine Learning Research* pp. 1225–1234 New York, New York, USA. PMLR.

[54] Belkin M, Hsu D, Ma S, Mandal S. 2019 Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences* **116**, 15849–15854.

[55] Radhakrishnan A, Yang K, Belkin M, Uhler C. 2018 Memorization in overparameterized autoencoders. *arXiv preprint arXiv:1810.10333*.

[56] Dereziński M, Liang F, Mahoney MW. 2019 Exact expressions for double descent and implicit regularization via surrogate random design. *arXiv preprint arXiv:1912.04533*.

[57] Girshick R. 2015 Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision* pp. 1440–1448.

[58] Kingma DP, Ba J. 2014 Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[59] Sutskever I, Martens J, Dahl G, Hinton G. 2013 On the importance of initialization and momentum in deep learning. In Dasgupta S, McAllester D, editors, *Proceedings of the 30th International Conference on Machine Learning* vol. 28*Proceedings of Machine Learning Research* pp. 1139–1147.

[60] Polyak BT, Juditsky AB. 1992 Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.* **30**, 838–855.

[61] He K, Zhang X, Ren S, Sun J. 2015 Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* pp. 1026–1034.

[62] Noack BR, Afanasiev K, Morzynski M, Tadmor G, Thiele F. 2003 A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics* **497**, 335–363.

[63] Taira K, Colonius T. 2007 The immersed boundary method: A projection approach.. *Journal of Computational Physics* **225**, 2118–2137.

[64] Colonius T, Taira K. 2008 A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering* **197**, 2131–2146.

[65] Li Y, Perlman E, Wan M, Yang Y, Meneveau C, Burns R, Chen S, Szalay A, Eyink G. 2008 A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. *Journal of Turbulence* p. N31.

[66] Yang J, Wright J, Huang TS, Ma Y. 2010 Image super-resolution via sparse representation. *IEEE Transactions on Image Processing* **19**, 2861–2873.

[67] Freeman WT, Jones TR, Pasztor EC. 2002 Example-based super-resolution. *IEEE Computer Graphics and Applications* **22**, 56–65.

[68] Raissi M, Karniadakis GE. 2018 Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics* **357**, 125–141.

[69] Benner P, Gugercin S, Willcox K. 2015 A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review* **57**, 483–531.