

NEURAL FUNCTIONAL SINGULAR VALUE DECOMPOSITION FOR IRREGULARLY SAMPLED INFINITE DIMENSIONAL DATA

Anonymous authors

Paper under double-blind review

ABSTRACT

This work studies an extension of the singular value decomposition to infinite-dimensional spaces by considering neural networks as basis elements. In contrast to the classical finite-dimensional singular value decomposition, this approach is grid-less and can be used in cases where only irregularly sampled data are available and evaluation at arbitrary sample points is required. To the best of our knowledge, we are the first work to propose a neural rank reduction method that is capable of handling irregularly sampled data. Our approach is based on a regularized least-squares formulation that fits the neural network to the given data while enforcing normalization for each function and orthogonality between pairs of functions. Performing rank reduction for infinite-dimensional operators is particularly interesting for scientific machine learning with focus on predicting the solution of partial differential equations given some boundary condition. In this context, the learned neural basis functions form a linear and finite-dimensional approximation of the image of the solution operator. We demonstrate the efficacy of our algorithm by first learning this approximation based on given irregularly sampled data. In a second stage, we train an artificial neural network as a coefficient functional for the previously learned basis.

1 INTRODUCTION

Deep learning has seen remarkable success in a large variation of different tasks, ranging from computer vision (Krizhevsky et al., 2012), natural language processing (Brown et al., 2020), or playing games (Silver et al., 2016). Recently, neural networks have also found their way into the domain of scientific computing tackling challenging problems in fields such as Earth systems science (Reichstein et al., 2019) or protein folding (Jumper et al., 2021).

A central task in computational science is to solve partial differential equations (PDEs) where, historically, finite difference methods and finite element methods have been dominant (Brenner & Scott, 2008). However, in recent years, there has been significant focus on machine learning methods in order to augment the pool of available methods (Beck et al., 2023; Gonon et al., 2024). One notable approach in this area has been physics-informed neural networks (Raissi et al., 2019), which uses the residual of the PDE as part of the loss function together with some available measurements in order to train a neural network as Ansatz for the PDE. A second direction is to use a variational formulation such as in the DeepRitz method (E & Yu, 2018) to minimize an energy functional, which was further developed into methods for solving high-dimensional PDEs such as the electronic Schrödinger equation (Pfau et al., 2020; Hermann et al., 2020; Scherbela et al., 2022). In a similar vein, the Feynman-Kac formula relates the solution of a Kolmogorov PDE to the minimization of an expectation value, which was utilized by (Han et al., 2018; Beck et al., 2021) to solve very high-dimensional problems in mathematical finance such as option pricing by the Black-Scholes formula.

The most interesting topic for the present work is the field of operator learning. In this field, available simulation and measurement data are used to directly learn to predict the solution of PDEs by learning the infinite-dimensional solution operator (Li et al., 2021; Lu et al., 2021; Boullé et al., 2024; Nelsen & Stuart, 2024).

These methods all share the same underlying rationale: the data is modeled as elements in an infinite-dimensional vector space. This is sensible due to the fact, that much of the data we encounter are naturally modeled as continuous objects, functions over a domain, hence infinite-dimensional in principle. Since computers always have finite memory, practical implementations of algorithms for infinite-dimensional data require embedding the data in some finite-dimensional representation. This embedding into a latent space can be seen as using encoders and decoders (Bhattacharya et al., 2021; Lanthaler et al., 2022). Note that these finite representations of infinite-dimensional models are different from directly using a finite model, as we assume the model to have potentially infinite resolution and being sampling invariant.

Most state-of-the-art models make the assumption that the finite-dimensional training data are obtained by point sampling on regular sample points, that is, the same set of evaluation points for each function in the training data (Li et al., 2021; Lu et al., 2022). In practice, however, output data are often irregularly sampled: each training instance is observed at its own set of sensor locations due to heterogeneous meshes, moving probes, or measurement constraints.

We address this setting with a neural functional singular value decomposition (nfSVD) where we focus on finding a low-rank approximation of infinite-dimensional data with irregular point samples. We learn a continuous, orthonormal range basis directly from scattered output samples via a single regularized least-squares objective evaluated only at the observed points. To obtain a deployable surrogate, we then train a coefficient functional that maps inputs to basis coefficients. In our experiments this is instantiated with either a fully-connected feedforward network or a convolutional neural network.

Our contributions include:

- A neural singular value decomposition method for representing infinite-dimensional data. In contrast to existing methods, our method allows for multi-dimensional inputs and provides a grid-free representation.
- A novel training scheme for the neural network basis functions which guarantees orthogonality and normalization via soft penalty constraint.
- Numerical validation that our method is indeed capable of learning basis functions from both regularly and irregularly sampled data over one- and two-dimensional domains. The obtained basis functions furthermore perform well in the downstream task of operator learning.
- Simulation results on regularly sampled data show that our method achieves a performance that is comparable to the provably optimal basis of finite-dimensional proper orthogonal decomposition basis. In our simulations on irregularly sampled data, our method indeed uniformly outperforms existing state-of-the-art methods.

2 RELATED WORK

The main motivation for our work stems from the task of building operator surrogates. Foundational theoretical work in operator learning was done by (Chen & Chen, 1995), who proved that the universal approximation property of neural networks also holds for mappings between infinite-dimensional spaces. The goal here is to approximate some non-linear operator $\mathcal{G}: \mathcal{U} \rightarrow \mathcal{V}$ between Banach spaces $\mathcal{U}$ and $\mathcal{V}$. We write $v(y) = \mathcal{G}[u](y)$ to indicate that the operator $\mathcal{G}$ acts on the function $u \in \mathcal{U}$ and the resulting function $v \in \mathcal{V}$ is evaluated at the point $y \in \text{dom } \mathcal{V}$.

Building on this foundation, the deep operator network (DeepONet) (Lu et al., 2021) was the first widely used operator learning architecture, which extends the shallow architectural design proposed by Chen & Chen (1995) to deep networks. To approximate the operator $\mathcal{G}: \mathcal{U} \rightarrow \mathcal{V}$, the approach is to consider a sequence of sampling points $\{x_i\}_{i \in \mathbb{N}}$ and the architecture

$$\hat{v}(y) = \hat{\mathcal{G}}[u; \{x_i\}_{i \in \mathbb{N}}](y) = \sum_{n \in \mathbb{N}} \Phi_n(\{u(x_i)\}_{i \in \mathbb{N}}) \Psi_n(y), \quad (2.1)$$

where Ψ is a basis for the infinite-dimensional output space and Φ are the corresponding coefficient functionals. For practical applications, however, such a system of infinite size cannot be realized and the coefficient representations have to be truncated after finitely many elements.

In the case of regularly sampled data Lu et al. (2022) proposed the POD-DeepONet, which first approximates the covariance matrix of the training data and then uses the first N -dominant eigenvectors of that matrix as a low-rank approximation of the range space of the operator, that is, as the basis from 2.1. Notice that this approach, while computationally efficient, comes with the drawback of being valid only on the fixed regular sample points of the dataset.

An extension of low-rank approximation to the infinite-dimensional setting is not straightforward. Constructing the full covariance function, such as it is done in the functional principal component analysis Ramsay & Silverman (2005), is expensive and requires a lot of data (see (Tan et al., 2025)). Lee & Shin (2024) approximate the image of the operator via a non-orthogonal neural network basis. Subsequently, they perform a finite-dimensional QR-decomposition on the functions sampled on regular sample points. However, this being done as post-processing after the training bears the risk of raising generalization error as now the basis does not necessarily form a good approximation of the distribution of functional data. Note that the method as proposed in (Lee & Shin, 2024) relies on regularly sampled data.

Tan et al. (2025) developed an approach that uses elements from a reproducing kernel Hilbert space (RKHS) to perform a functional singular value decomposition for infinite-dimensional data from heterogeneous sources. For an RKHS $\mathcal{H}_K$, sampling points $\{T_{ij}|j \in [J_i]\}$, and hidden functions $X_1, \dots, X_N \in \mathcal{H}_K$ with noisy observations $Y_{ij} = X_i(T_{ij}) + \varepsilon_{ij}$, their algorithm iteratively finds the functional singular value decomposition up to rank R by finding the current leading singular component via

$$\arg \min_{\mathbf{a}_1 \in \mathbb{R}^N, \phi_1 \in \mathcal{H}_K} \sum_{i=1}^N \frac{1}{J_i} \sum_{j=1}^{J_i} (Y_{ij} - a_{i1} \phi_1(T_{ij}))^2 + \nu \|\mathbf{a}\|^2 \cdot \|\mathcal{P} \phi_1\|_{\mathcal{H}}. \quad (2.2)$$

Here, $\nu > 0$ is a tuning parameter, and $\mathcal{P}$ is a projection operator from $\mathcal{H}_K$ onto its subspace. The subspace corresponding to ϕ_1 is then removed from the data, and they continue sequentially for the next singular component. This approach operates in a kernel regime and is limited to one-dimensional input data.

3 NEURAL FUNCTIONAL SINGULAR VALUE DECOMPOSITION

In this section, we present a nfSVD tailored to operator learning with irregularly sampled outputs. Unlike RKHS-based variants, we work with a neural hypothesis class $\mathbb{F}_{d,N} \subset L^2(\mathbb{D})$, a grid-free least-squares objective, and enforce $L^2(\mathbb{D})$ orthonormality via Monte-Carlo inner products. The result is a finite-rank, mesh-independent range basis suitable for multi-dimensional domains and heterogeneous sensors. In the following we now formalize the objective and its discretization.

In the conventional finite-dimensional setting, the singular value decomposition (SVD) is a method for factorizing a matrix $\mathbf{X} \in \mathbb{C}^{J \times M}$ into $\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$, where $\mathbf{U} \in \mathbb{C}^{J \times J}$, and $\mathbf{V} \in \mathbb{C}^{M \times M}$ are unitary matrices and $\mathbf{\Sigma} \in \mathbb{R}^{J \times M}$ is a rectangular matrix with decreasing non-negative entries on the diagonal. By restricting this factorization only to the first $N < J$ columns of $\mathbf{U}$ and $\mathbf{V}$ it is known to be the best possible low-rank approximation of $\mathbf{X}$ by the Eckart-Young-Theorem (see (Golub & Van Loan, 2013, Theorem 2.4.8)).

In this work, we restrict to real-valued spaces and consider an extension of this to the infinite-dimensional setting. That is, instead of a matrix, we consider a linear operator $X \in \mathcal{L}(\mathbb{R}^M, \mathcal{H})$ from the M -dimensional Euclidean space to some separable Hilbert space $\mathcal{H}$. In this setting, we can then uniquely identify X with an M -tuple $(x_1, \dots, x_M) \subset \mathcal{H}$ such that $X : \mathbb{R}^M \rightarrow \mathcal{H}$ is the map $\mathbf{v} \mapsto \sum_{m=1}^M v_m x_m$. As orthonormal basis in $\mathcal{H}$ for a rank N approximation of the operator X with $N \leq M$ we then consider the solution of the constrained least-squares problem

$$\begin{aligned} (r_1^*, \dots, r_N^*), \mathbf{A}^* \in & \arg \min_{\substack{(r_1, \dots, r_N) \in \mathcal{H}^N, \\ \mathbf{A} \in \mathbb{R}^{N \times M}}}, \sum_{m=1}^M \left\| x_m - \sum_{n=1}^N A_{n,m} r_n \right\|_{\mathcal{H}}^2 \\ \text{s.t. } & \langle r_{m_1}, r_{m_2} \rangle_{\mathcal{H}} = \mathbf{1}_{m_1=m_2}, \forall m_1, m_2 \in [M]. \end{aligned} \quad (3.1)$$

For a practical implementation of this problem, we consider the Hilbert space $L^2(\mathbb{D})$ with compact domain $\mathbb{D} \subset \mathbb{R}^d$ and make the following observations:

- We do not have access to the infinite-dimensional operator X but only to point samples of the data $\{x_1, \dots, x_M\}$ and need to reformulate the minimization.
- Optimization over the full space $L^2(\mathbb{D})$ (or even its unit sphere) is impossible and we need to restrict to some parametrized hypothesis class $\mathbb{F}$.
- For non-linearly parametrized hypothesis classes, it is infeasible to enforce the orthonormality as a hard constraint and we need to resort to regularization instead.

3.1 DISCRETIZED FUNCTION DATA

We consider the setting, where for each function x_m we have access to a certain number $T_m \in \mathbb{N}$ of sample points

$$\mathbf{s}^{(m)} := (s_1^{(m)}, \dots, s_{T_m}^{(m)}) \in \mathbb{D}^{T_m} \quad (3.2)$$

and the corresponding function values

$$\mathbf{x}^{(m)} := (x_m(s_1^{(m)}), \dots, x_m(s_{T_m}^{(m)})) \in \mathbb{R}^{T_m}. \quad (3.3)$$

Based on the sample points, we then also define for the basis functions $r_1, \dots, r_N \in L^2(\mathbb{D})$ the representation

$$\mathbf{R}^{(m)} := (r_n(s_t^{(m)}))_{t \in [T_m], n \in [N]} \in \mathbb{R}^{T_m \times N}. \quad (3.4)$$

We consider two major cases: regularly sampled data and irregularly sampled data. Here, regular sampling means that the sample points stay constant over different data functions, that is, $T_{m_1} = T_{m_2}$ and $\mathbf{s}^{(m_1)} = \mathbf{s}^{(m_2)}$ for all $m_1, m_2 \in [M]$. Note, however, that this does not necessarily mean that the sample points lie on a regular grid. Contrary to that, irregular sampling means, that the sample points vary for different data functions.

Based on the data discretization, the loss function will also be discretized to

$$\left\| x_m - \sum_{n=1}^N A_{n,m} r_n \right\|_{L^2(\mathbb{D})}^2 \approx \frac{|\mathbb{D}|}{T_m} \left\| \mathbf{x}^{(m)} - \mathbf{R}^{(m)} \mathbf{A}_{:,m} \right\|_2^2, \quad (3.5)$$

which is a statistically unbiased estimator, given that the sample points are i.i.d. uniform in $\mathbb{D}$ or on a regular grid.

Similarly to that, we also approximate the inner product. To do so, we build the union over all available sample points and the corresponding matrix representation of the basis functions

$$\bar{\mathbf{s}} := (\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(M)}) \in \mathbb{D}^T, \quad \text{and} \quad \bar{\mathbf{R}} := (r_n(\bar{s}_t))_{t \in [T], n \in [N]}, \quad (3.6)$$

respectively, with $T = \sum_{m=1}^M T_m$. The resulting approximation is then

$$\langle r_{n_1}, r_{n_2} \rangle_{L^2(\mathbb{D})} \approx \frac{|\mathbb{D}|}{T} \bar{\mathbf{R}}_{:,n_2}^T \bar{\mathbf{R}}_{:,n_1}, \quad (3.7)$$

which is again unbiased if the sample points are i.i.d. uniform in $\mathbb{D}$ or on a regular grid.

3.2 HYPOTHESIS CLASS AND ORTHONORMAL REGULARIZATION

As hypothesis class for the basis functions, we consider artificial neural networks (ANNs) with d input neurons and N output neurons and a fixed hidden architecture, which we denote by $\mathbb{F}_{d,N}$. The hidden architecture is not important for the current discussion and will be specified later in the experiments in Section 4. Note here, that $f \in \mathbb{F}_{d,N}$ is not necessarily a tuple of N functions with d -dimensional input but rather a functions $f : \mathbb{R}^d \rightarrow \mathbb{R}^N$. However, it is clear that $\mathbb{F}_{d,N} \subset L^2(\mathbb{D})^N$.

Unlike kernel methods, which are naturally formulated in a Hilbert space, ANNs do not have a canonic way of forcing the orthogonality of individual functions r_n in the parameter space. Enforcing the orthogonality a posteriori bears the risk of worsening performance due to the non-linear nature of neural network parametrization. Thus, directly enforcing the constraint $\langle r_{n_1}, r_{n_2} \rangle_{L^2(\mathbb{D})} =$

$\mathbf{1}_{n_1=n_2}$ is practically infeasible. Instead, we reformulate the constraint into regularization terms of the form $(\langle r_{n_1}, r_{n_2} \rangle_{L^2(\mathbb{D})} - \mathbf{1}_{n_1=n_2})^2$, which are weighted by regularization parameters τ_N for normalization (i.e., $n_1 = n_2$) and τ_O for orthogonalization (i.e., $n_1 \neq n_2$). Notice that we also enforce normality of the neural basis functions to avoid individual functions collapsing to zero or individual modes becoming too dominant. In this way, we can ensure that the basis has a more equitable weight across the spectrum. Here, directly enforcing the constraint would be possible by rescaling the last layer. However, since gradient methods for neural network training such as ADAM (Kingma & Ba, 2017) are based on moment estimation, this kind of additional projection step would worsen optimization results.

Overall, the final learning problem is given by

$$(r_1^*, \dots, r_N^*), \mathbf{A}^* \in \arg \min_{\substack{(r_1, \dots, r_N) \in \mathbb{F}_{d,N} \\ \mathbf{A} \in \mathbb{R}^{N \times M}}} \sum_{m=1}^M \frac{1}{T_m} \left\| \mathbf{x}^{(m)} - \mathbf{R}^{(m)} \mathbf{A}_{:,m} \right\|_2^2 \quad (3.8)$$

$$+ \tau_N \frac{|\mathbb{D}|}{T^2 N} \sum_{n=1}^N (\|\bar{R}_{:,n}\|_2^2 - 1)^2 + \tau_O \frac{|\mathbb{D}|}{T^2 N(N-1)} \sum_{\substack{n_1=1 \\ n_1 \neq n_2}}^N (\bar{R}_{:,n_2}^T \bar{R}_{:,n_1})^2.$$

4 EXPERIMENTS

For the experimental analysis of our algorithm we look at the downstream task of learning infinite-dimensional operators (see Section 2). That is, we assume that we are given the initial condition or some parameters (the input) of the PDE and aim to predict the solution (the output). In this setting, the training data are composed of point samples taken from M pairs of input- and output-functions $k: \mathbb{D}_I \rightarrow \mathbb{R}^{d_I}$ and $x: \mathbb{D}_O \rightarrow \mathbb{R}^{d_O}$, respectively. Here, $\mathbb{D}_I, \mathbb{D}_O, d_I, d_O$ represent the input and output domain and the dimensions of the input and output functions, respectively. In all our experiments we will assume that $d_I = d_O = 1$ and $\mathbb{D}_I = \mathbb{D}_O = [0, 1]^d =: \mathbb{D}$. For each input function k_m with $m \in [M]$ we are then given a set of $S_m \in \mathbb{N}$ sample points $\mathbf{y}^{(m)} \in \mathbb{D}^{S_m}$ with the corresponding function samples $\mathbf{k}^{(m)} \in \mathbb{R}^{S_m}$. Analogously, for the output functions x_m , we are given $T_m \in \mathbb{N}$ sample points $\mathbf{s}^{(m)} \in \mathbb{D}^{T_m}$ with the corresponding function samples $\mathbf{x}^{(m)} \in \mathbb{R}^{T_m}$.

For the operator learning task, we then assume that the input data is regularly sampled, that is, $\mathbf{y}^{(m_1)} = \mathbf{y}^{(m_2)} =: \mathbf{y}$ with $S_{m_1} = S_{m_2} =: S$, while we consider both cases of regularly and irregularly sampling for the output functions.

In all our experiments, we consider a set of 1000 pairs of input output functions for training and a set of 200 pairs for testing. As parameters, we set the regularization parameters to $\tau_O = 10^{-3}$, $\tau_N = 10^{-5}$, an initial learning rate (for both steps) of 10^{-3} , and a weight decay regularization of magnitude 10^{-4} . The learning rate is reduced with a geometric decay starting after 20000 epochs for the basis functions and with an inverse time decay for the coefficient functionals, both at a rate of 10^{-4} . Differences to this standard setting will be mentioned when necessary.

Our main error metric will be the relative $L^2(\mathbb{D})$ -error which is averaged over five independent initializations of the neural networks. Note, that the random sampling of our data is kept constant over these individual runs such that the statistics of the $L^2(\mathbb{D})$ -error solely reflect the variations in the model and not the variations in the dataset.

4.1 OPERATOR LEARNING VIA NFSVD

In order to apply our algorithm to the operator learning task, we identify the collection of output functions $(x_1, \dots, x_M)$ with the operator $X: \mathbb{R}^M \rightarrow L^2(\mathbb{D})$. We then use our algorithm (3.8) in order to find a neural network Ψ^* which provides a low-rank approximation for the span of X . For this learned basis and the corresponding coefficient matrix $\mathbf{A}^*$, we train another network Φ , with the aim that each Φ_n is the coefficient functional for the n -th basis function. These functionals are trained to fit the data given by the samples of the input functions and the corresponding coefficients in the matrix $\mathbf{A}^*$ of (3.8) in a least-squares sense, thus conforming to the operator learning formalism of (2.1).

A similar two-stage training method to learn an orthonormal basis of neural networks was proposed in form of the QR-DeepONet (Lee & Shin, 2024), which aims to reduce the complexity of the training procedure for DeepONets. In this approach, the first step is to train a neural network $\Psi \in \mathbb{F}_{d,N}$ on the output data using a least-squares objective. In a second stage, the network is transformed into a matrix representation by evaluating on the sampling points. By means of a QR-decomposition of the matrix representation the approach obtains a reweighting matrix $\mathbf{R}$, whose inverse is applied to the neural network Ψ such that the basis functions are orthonormal. Note, that in the original description of Lee & Shin (2024), the approach is limited to regularly sampled data. We extend this formulation to irregularly sampled data, by considering the matrix representation $\bar{\mathbf{R}}$ on the union of all sample points as we have done in our approach in (3.6).

Furthermore, we compare to the predecessors of QR-DeepONet, namely the Vanilla DeepONet Lu et al. (2021) and the POD-DeepONet Lu et al. (2022) where the network Ψ is replaced by a matrix $\mathbf{P} \in \mathbb{R}^{T \times N}$. In order to ensure a fair comparison between the different models, we use the same hyperparameters for training whenever possible. For example, our approach and QR-DeepONet use the same optimizer, initial learning rate and learning rate scheduler for the training of the basis functions in Ψ . Furthermore, the output functions are rescaled by $\sqrt{N}$ according to the second order analysis of (Lu et al., 2022).

4.2 PROBLEM SETUP

For the experiments we resort to some of the commonly used test settings of the operator learning literature. Namely, the one dimensional viscous Burgers equation (Li et al., 2021; Lu et al., 2022), Darcy flow on a rectangular domain with continuous permeability field (Lu et al., 2022), and Darcy flow on a rectangular domain with piecewise constant permeability field (Li et al., 2021; Lu et al., 2022). In the following we proceed with a detailed description of our datasets and method of random sampling.¹

4.2.1 BURGERS DATA

We consider the one dimensional viscous Burgers equation

$$\partial_t u(x, t) + \partial_x(u^2(x, t)/2) = \nu \partial_{xx} u(x, t), \quad x \in (0, 1), t \in (0, 1] \quad (4.1)$$

$$u(x, 0) = u_0(x), \quad x \in (0, 1) \quad (4.2)$$

with periodic boundary conditions and viscosity $\nu = 0.1$. This dataset originates from (Li et al., 2021), where initial condition u_0 was sampled from $\mathcal{N}(0, 625(-\Delta + 25I)^{-2})$ and the data was generated with a spatial resolution of 2^{13} points. The task of operator learning is then to consider the input $k = u_0$ and predict the solution $x = u(\cdot, 1)$ at time $t = 1$.

For our simulations on regularly sampled data, we subsample all data to a spatial resolution of 128 points with uniform spacing. For irregularly sampled output data, we perform a random subsampling of the output data, which means, a subset of 128 unique sample points is picked uniformly from the original 2^{13} points. Examples of input-output pairs for both regular and irregular sampling are shown in Fig. 1.

In order to enforce the periodic boundary conditions, we adopt the approach of (Lu et al., 2022) to transform the spatial variable

$$s \mapsto (\cos(2\pi s), \sin(2\pi s), \cos(4\pi s), \sin(4\pi s))$$

at the input of Ψ . The architecture of our networks for this case is that both Ψ and Φ have 3 hidden layers with 128 neurons and tanh activation function, and we consider $N = 32$ basis functions. We train the basis functions for 50000 epochs and the coefficient functionals for 1000000 epochs.

4.2.2 DARCY CONTINUOUS PERMEABILITY FIELD

An example for a two-dimensional problem for operator learning is given by the Darcy-Flow problem on a rectangular domain with a log-normal random field as permeability field, constant one Dirichlet boundary condition on the left, zero Dirichlet boundary condition on the right, and zero

¹A link to the source code will be made available upon publication

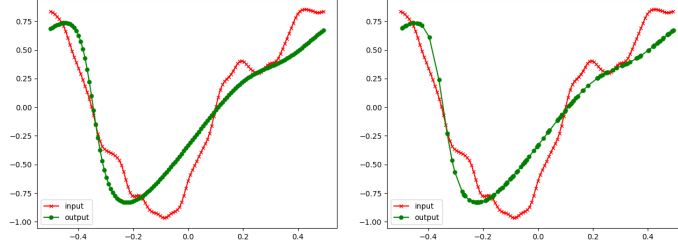


Figure 1: Examples of input-output pairs of the Burgers dataset for regularly sampled output (left) and irregularly sampled output (right).

Table 1: Results for Burgers equation with regular sampling and irregular sampling

	regular sampling		irregular sampling	
	rel. L^2	proj. rel. L^2	rel. L^2	proj. rel. L^2
nfSVD	1.39 ± 0.14 %	0.19 %	1.46 ± 0.09 %	0.18 %
POD (Lu et al., 2022)	1.39 ± 0.04 %	0.21 %	N/A	N/A
QR (Lee & Shin, 2024)	1.58 ± 0.23 %	0.17 %	1.62 ± 0.19 %	0.15 %
VAN (Lu et al., 2021)	1.70 ± 0.04 %	0.36 ± 0.01 %	1.80 ± 0.05 %	0.41 ± 0.01 %

Neumann boundary condition on the top and bottom boundary of the domain. That is, we consider the problem

$$-\nabla \cdot (k(s)\Delta x(s)) = 0, \quad s \in (0, 1)^2, \quad (4.3)$$

$$x(s) = 1 - s_1, \quad s_1 \in \{0, 1\}, s_2 \in (0, 1), \quad (4.4)$$

$$\partial_n x(s) = 0, \quad s_1 \in (0, 1), s_2 \in \{0, 1\} \quad (4.5)$$

with the permeability field $k(s) = \exp(f(s))$ where f is a two dimensional Gaussian process with covariance kernel $\exp(\|s\|_2^2/0.5^2)$. For this dataset we could not reuse the data of (Lu et al., 2022) as we needed a higher resolution available for random subsampling. Instead, we use the FEniCS finite element simulation tool (Alnæs et al., 2015) to simulate the PDE with a resolution on a grid with 401×401 points. For the regularly sampled case, we then subsample this to a resolution of 21×21 points, resulting in 441 points. For the irregularly sampled case, we then subsampled from the two-dimensional data by randomly selecting 441 unique sample points. An example is given in Fig. 2.

The architecture of our networks for this case is that Ψ has 3 hidden layers of width 40 with tanh activation function, Φ is a convolutional neural network with tanh activation function, and we consider $N = 10$ basis functions. We train the basis functions for 50000 epochs and the coefficient functionals for 100000 epochs.

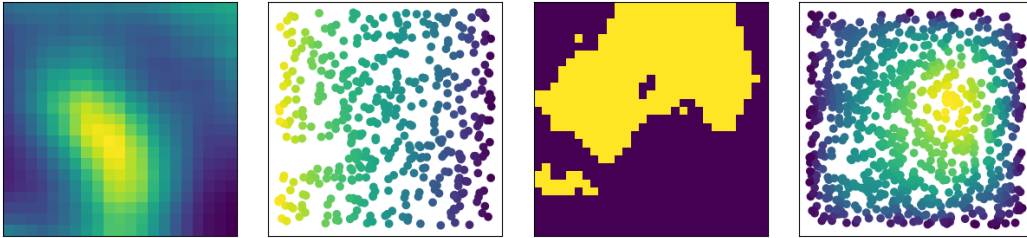


Figure 2: Example of input-output pair for Darcy flow on a rectangular domain with regularly sampled input and irregularly sampled output. The plots on the left show one example of for the continuous permeability field and the plots on the right show one example of a piecewise constant permeability field.

Table 2: Results for Darcy flow with continuous permeability field with regular sampling and irregular sampling

	regular sampling		irregular sampling	
	rel. L^2	proj. rel. L^2	rel. L^2	proj. rel. L^2
nfSVD	1.12 ± 0.02 %	0.79 %	1.48 ± 0.03 %	0.98 ± 0.03 %
POD (Lu et al., 2022)	1.13 ± 0.02 %	0.78 %	N/A	N/A
QR (Lee & Shin, 2024)	1.14 ± 0.04 %	0.78 %	1.51 ± 0.03 %	0.92 %
VAN (Lu et al., 2021)	1.15 ± 0.02 %	0.79 ± 0.01 %	1.56 ± 0.12 %	0.97 ± 0.04 %

Table 3: Results for Darcy flow with piecewise constant permeability field with regular sampling and irregular sampling

	regular sampling		irregular sampling	
	rel. L^2	proj. rel. L^2	rel. L^2	proj. rel. L^2
nfSVD	2.27 ± 0.01 %	1.06 %	2.79 ± 0.17 %	1.38 ± 0.21 %
POD (Lu et al., 2022)	2.26 ± 0.01 %	0.92 %	N/A	N/A
QR (Lee & Shin, 2024)	2.25 ± 0.01 %	0.93 %	15.12 ± 25.04 %	0.98 ± 0.06 %
VAN (Lu et al., 2021)	2.61 ± 0.02 %	1.15 ± 0.01 %	2.98 ± 0.03 %	1.05 %

4.2.3 DARCY PIECEWISE CONSTANT PERMEABILITY FIELD

We also tested our method for another instantiation of the Darcy flow problem in two dimensions. Here, we chose homogeneous Dirichlet boundary condition of zero and a piecewise constant permeability field given by

$$k(s) = \begin{cases} 12, & \text{if } T > 0 \\ 3, & \text{otherwise} \end{cases}$$

where T was sampled by a Gaussian random field with a kernel of $\mathcal{N}(0, (-\Delta + 9I)^{-2})$ and zero Neumann boundary condition. This problem poses an interesting challenge as the spectrum of the covariance operator is decaying very slowly. This then also accounts for the generally high number of basis functions needed for a good approximation. The original data has a resolution of 421×421 , which we subsampled to a grid 29×29 points in the regular case and 841 randomly selected but unique points in the irregular case. We used the raw data from the repository of (Lu et al., 2022).

The architecture of our networks for this case is that Ψ has 4 hidden layers with 256 neurons and tanh activation function, Φ is a convolutional neural network with tanh activation function, and we consider $N = 115$ basis functions. The hyperparameters in this experiments deviate from the standard setting in the sense that $\tau_O = 10^{-2}$, $\tau_N = 10^{-4}$ and the learning rate decay for the training of the basis functions has a rate of $5 \cdot 10^{-5}$. We train the basis functions for 50000 epochs and the coefficient functionals for 100000 epochs.

4.3 DISCUSSION

The resulting relative $L^2(\mathbb{D})$ error for both sampling methods is presented in Tables 1 to 3 alongside the resulting relative $L^2(\mathbb{D})$ error of the projection of the output functions onto the learned basis Ψ for nfSVD, QR-DeepONet, and DeepONet and P for the POD-DeepONet. We observe that the relative $L^2(\mathbb{D})$ error of our model consistently shows the best results in the irregularly sampled output case, while achieving the performance of the POD-DeepONet for the regularly sampled case. It is important to note that while the proper orthogonal decomposition used as basis in this architecture is provably optimal for rank reduction, both the QR-DeepONet and nfSVD network achieve lower projection error in our experiments on the Burgers data. The reason for that lies that we take the relative $L^2(\mathbb{D})$ projection error and both networks are trained on normalized function samples, whereas the POD-DeepONet architecture is build upon unnormalized data.

An interesting question arises from the projection error of the modified QR-DeepONet being generally lower than from the architecture based on nfSVD. This fact is however not surprising as the

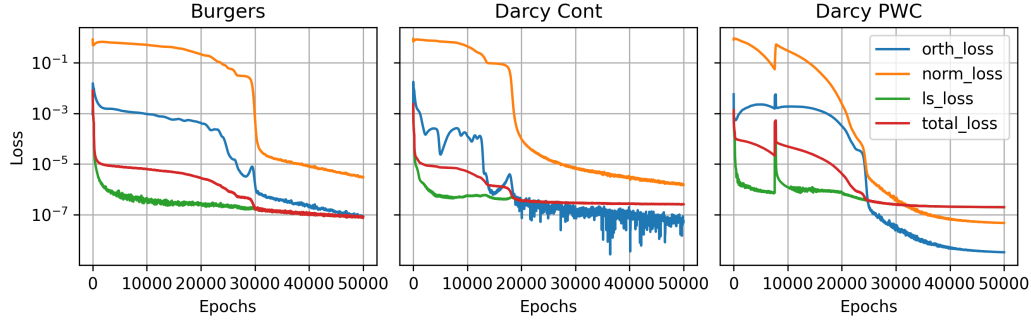


Figure 3: Convergence of nfSVD for irregularly sampled data on Burgers data (left), Darcy flow with continuous permeability field (middle) and Darcy flow with piecewise constant permeability field (right).

QR-DeepONet does not possess regularization terms, leaving data fit to be the sole objective. The generalization loss, on the contrary, shows better results for the regularized nfSVD approach, highlighting the possible advantage of such constraints. This is also visible in the last problem (see Table 3), where the unregularized approach failed to converge in some of the runs resulting in very high error.

Using a soft penalty constraint for orthonormality opens up the question to which degree we can actually achieve this in our neural network basis. As can be seen from Fig. 3, the normalization loss experiences a step-wise behavior which culminates in a one-time sharp drop at which also the orthogonality loss shows a significant drop. This behavior is explained by the fact that throughout our simulations we set the orthogonality parameter τ_O to be two orders of magnitude larger than the normality parameter τ_N . With this setting, the network Ψ is initially close to the zero function and whenever one of its outputs opens up a new linearly independent direction in $L^2(\mathbb{D})$, this output converges quickly to unit norm, resulting in the step of the normalization loss. The single significant drop then indicates that the last output converged to unit norm. As a consequence of our choice of τ_O and τ_N the normalization loss is generally higher than the orthogonality loss, however, we see that we indeed achieve the initial goal of learning orthonormal functions up to numerical inaccuracy (see Appendix B).

5 CONCLUSION

In this work, we introduced the neural functional singular value decomposition and applied it to learning operator surrogates. We have shown empirically that this method produces stable, accurate orthonormal approximations of the range space of different relevant PDE solution operators. The nfSVD consistently shows superior results on irregularly sampled data, while achieving results identical to the provably optimal row-rank approximation on shared grids.

There are several avenues of research going from here. A straightforward extension could be to utilize the nfSVD for encodings of the input space, either on its own or in combination with an output space encoding. Sequential training of neural basis functions could bring the benefit of finding the dominant modes first and lead to greater similarity of the learn basis with a proper orthogonal decomposition in the discrete space. Solving the least-squares problem for the basis coefficients could lead to a potentially even faster convergence at the price of a much more costly backpropagation. Additionally, exploring natural gradient descent or subspace methods for the basis coefficients could prove viable. An extension in a new direction outside the scope of this paper is given by non-linear low rank approximations. Instead of approximating the range space with a linear approximation given by an N -dimensional vector space, one might use non-linear elements parameterizing an N -dimensional manifold potentially leading to better or more efficient approximation of the structure of, for example, a non-linear operator.

REFERENCES

- Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E Rognes, and Garth N Wells. The FEniCS Project Version 1.5. *Archive of Numerical Software*, Vol 3, 2015. doi: 10.11588/ANS.2015.100.20553. URL <http://journals.ub.uni-heidelberg.de/index.php/ans/article/view/20553>. Publisher: University Library Heidelberg Version Number: 1.0.0.
- Christian Beck, Sebastian Becker, Philipp Grohs, Nor Jaafari, and Arnulf Jentzen. Solving the Kolmogorov PDE by Means of Deep Learning. *Journal of Scientific Computing*, 88(3):73, September 2021. ISSN 0885-7474, 1573-7691. doi: 10.1007/s10915-021-01590-0. URL <https://link.springer.com/10.1007/s10915-021-01590-0>.
- Christian Beck, Martin Hutzenthaler, Arnulf Jentzen, and Benno Kuckuck. An overview on deep learning-based approximation methods for partial differential equations. *Discrete and Continuous Dynamical Systems - B*, 28(6):3697–3746, 2023. ISSN 1531-3492, 1553-524X. doi: 10.3934/dcdsb.2022238. URL <http://arxiv.org/abs/2012.12348>. arXiv:2012.12348 [math].
- Kaushik Bhattacharya, Bamdad Hosseini, Nikola Borislavov Kovachki, and Andrew M. Stuart. Model Reduction and Neural Networks for Parametric PDEs. *The SMAI journal of computational mathematics*, 7:121–157, July 2021. ISSN 2426-8399. doi: 10.5802/smai-jcm.74. URL <https://smai-jcm.centre-mersenne.org/articles/10.5802/smai-jcm.74/>.
- Nicolas Boullé, Diana Halikias, Samuel E. Otto, and Alex Townsend. Operator learning without the adjoint, November 2024. URL <http://arxiv.org/abs/2401.17739>. arXiv:2401.17739 [math].
- Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer New York, New York, NY, 2008. ISBN 978-0-387-75933-3 978-0-387-75934-0. doi: 10.1007/978-0-387-75934-0. URL <http://link.springer.com/10.1007/978-0-387-75934-0>.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*, pp. 1877–1901, Red Hook, NY, USA, December 2020. Curran Associates Inc. ISBN 978-1-7138-2954-6.
- Tianping Chen and Hong Chen. Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, July 1995. ISSN 10459227. doi: 10.1109/72.392253.
- Weinan E and Bing Yu. The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems. *Communications in Mathematics and Statistics*, 6(1):1–12, March 2018. ISSN 2194-6701, 2194-671X. doi: 10.1007/s40304-018-0127-z. URL <http://link.springer.com/10.1007/s40304-018-0127-z>.
- Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns hopkins studies in the mathematical sciences. Johns Hopkins University Press, 2013. ISBN 978-1-4214-0794-4.
- Lukas Gonon, Arnulf Jentzen, Benno Kuckuck, Siyu Liang, Adrian Riekert, and Philippe von Wurstemberger. An Overview on Machine Learning Methods for Partial Differential Equations: from Physics Informed Neural Networks to Deep Operator Learning, August 2024. URL <http://arxiv.org/abs/2408.13222>. arXiv:2408.13222 [math].
- Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, August 2018. doi: 10.1073/pnas.1718942115. URL <https://www.pnas.org/doi/10.1073/pnas.1718942115>. Publisher: Proceedings of the National Academy of Sciences.

- Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic Schrödinger equation. *Nature Chemistry*, 12(10):891–897, October 2020. ISSN 1755-4330, 1755-4349. doi: 10.1038/s41557-020-0544-y. URL <https://www.nature.com/articles/s41557-020-0544-y>.
- John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2.
- Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. URL <http://arxiv.org/abs/1412.6980>. arXiv:1412.6980 [cs].
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, volume 1 of *NIPS’12*, pp. 1097–1105, Red Hook, NY, USA, December 2012. Curran Associates Inc.
- Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error Estimates for DeepONets: A Deep Learning Framework in Infinite Dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, March 2022. ISSN 2398-4945. doi: 10.1093/imatrm/tnac001.
- Sanghyun Lee and Yeonjong Shin. On the Training and Generalization of Deep Operator Networks. *SIAM Journal on Scientific Computing*, 46(4):C273–C296, August 2024. ISSN 1064-8275, 1095-7197. doi: 10.1137/23M1598751. URL <https://epubs.siam.org/doi/10.1137/23M1598751>.
- Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhat-tacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, 2021.
- Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning Nonlinear Operators Via DeepONet Based on the Universal Approximation Theorem of Operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5.
- Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A Comprehensive and Fair Comparison of Two Neural Operators (with Practical Extensions) Based on FAIR Data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, April 2022. ISSN 00457825. doi: 10.1016/j.cma.2022.114778.
- Nicholas H. Nelsen and Andrew M. Stuart. Operator Learning Using Random Features: A Tool for Scientific Computing. *SIAM Review*, 66(3):535–571, May 2024. ISSN 0036-1445, 1095-7200. doi: 10.1137/24M1648703.
- David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. *Ab initio* solution of the many-electron Schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, September 2020. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.2.033429. URL <https://link.aps.org/doi/10.1103/PhysRevResearch.2.033429>.
- M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. ISSN 00219991. doi: 10.1016/j.jcp.2018.10.045. URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125>.

- J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer New York, New York, NY, 2005. ISBN 978-0-387-40080-8 978-0-387-22751-1. doi: 10.1007/b98888. URL <http://link.springer.com/10.1007/b98888>.
- Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat. Deep learning and process understanding for data-driven Earth system science. *Nature*, 566(7743):195–204, February 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-0912-1. URL <https://www.nature.com/articles/s41586-019-0912-1>. Publisher: Nature Publishing Group.
- Michael Scherbela, Rafael Reisenhofer, Leon Gerard, Philipp Marquetand, and Philipp Grohs. Solving the Electronic Schrödinger Equation for Multiple Nuclear Geometries with Weight-Sharing Deep Neural Networks. *Nature Computational Science*, 2(5):331–341, May 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00228-x. URL <https://www.nature.com/articles/s43588-022-00228-x>.
- David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. ISSN 1476-4687. doi: 10.1038/nature16961. URL <https://www.nature.com/articles/nature16961>. Publisher: Nature Publishing Group.
- Jianbin Tan, Pixu Shi, and Anru R. Zhang. Functional Singular Value Decomposition, February 2025. URL <http://arxiv.org/abs/2410.03619>. arXiv:2410.03619 [stat].

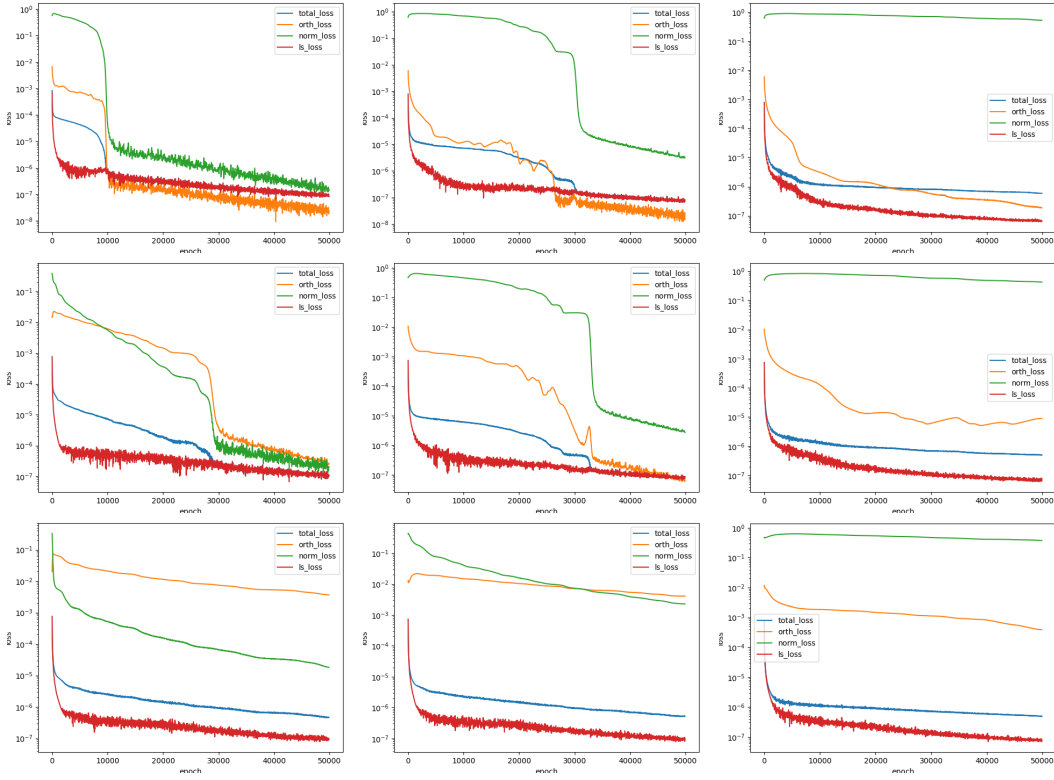


Figure 4: Loss curves for different choices of the regularization parameters τ_O and τ_N . From top to bottom we vary $\tau_O \in \{10^{-2}, 10^{-3}, 10^{-4}\}$ and from left to right we vary $\tau_N \in \{10^{-4}, 10^{-5}, 10^{-6}\}$.

A BURGERS REGULARIZATION PARAMETERS

In this section we want to provide a short ablation study on the influence of differing regularization parameters for the problem of solving the Burgers equation 4.2.1. We tested the parameters τ_O and τ_N at different orders of magnitude, among which we show all combinations for the orthogonality parameter $\tau_O \in \{10^{-2}, 10^{-3}, 10^{-4}\}$ and the normality parameter $\tau_N \in \{10^{-4}, 10^{-5}, 10^{-6}\}$ in Fig. 4. It is clearly visible that the smallest choice $\tau_O = 10^{-4}$ results in non-convergence of the orthogonalization loss within 50000 epochs, while the smallest choice $\tau_N = 10^{-6}$ results in non-convergence of the normalization loss. We observe that higher regularization parameters result in faster convergence, however, the combination $\tau_O = 10^{-3}$ and $\tau_N = 10^{-5}$ results in the lowest possible least squares loss. This justifies our specific choice of parameters in our experiments.

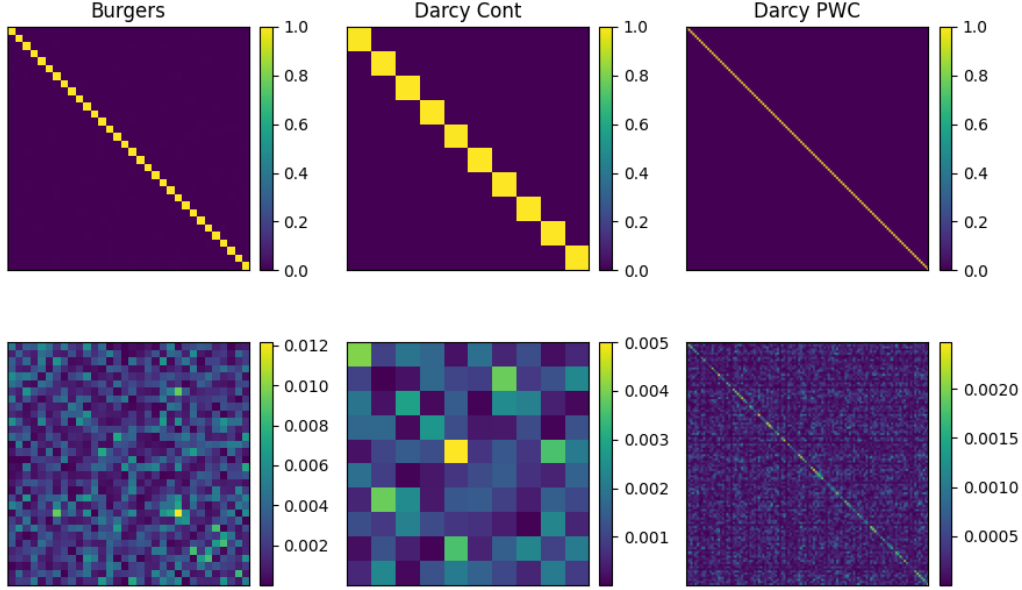


Figure 5: Correlation matrices (upper row) for basis functions and the corresponding error matrices (bottom row). The matrices are obtained from irregularly sampled data on the three datasets considered in our experiments.

B ORTHOGONALITY VISUALIZATION

In Fig. 5 we visualize the orthogonality and error in orthogonality in order to demonstrate the quality of our solutions. We do this by computing the correlation matrix $\frac{|\mathbb{D}|}{T} \bar{R}^T \bar{R}$ with $|\mathbb{D}| = 1$ and the absolute deviation of this matrix from the identity matrix. Orthogonality is an important property of a low-rank approximation, as it guarantees efficiency in the number of modes used. As functions are orthogonal exactly then, when their inner product is zero, a pure penalty of correlation tends to give outputs that are very small in norm. It is well known in machine learning, that optimization tends to work better on normalized functions, as such, building operator surrogates with a coefficient functional would prove much more unstable and error-prone.