Electronic Transactions on Numerical Analysis.
Volume 64, pp. 23–44, 2025.

ISSN 1068–9613.
DOI: 10.1553/etna_vol64s23

ETNA
Kent State University and
Johann Radon Institute (RICAM)

SPECTRAL FUNCTION SPACE LEARNING AND NUMERICAL LINEAR ALGEBRA NETWORKS FOR SOLVING LINEAR INVERSE PROBLEMS*

ANDREA ASPRI†, LEON FRISCHAUF‡, AND OTMAR SCHERZER‡§¶

Abstract. We consider solving an ill-conditioned linear operator equation, where the operator is not modeled by physical laws but is specified via training pairs (consisting of images and data) of the input–output relation of the operator. The proposed method for computing the operator from training pairs consists of a Gram–Schmidt orthonormalization of images and a principal component analysis of data. Interestingly, this two-step algorithm provides us with a spectral decomposition of the linear operator, without explicit knowledge of it. Moreover, we indicate that both Gram–Schmidt and principal component analyses can be expressed as a deep neural network which delivers orthonormalized vectors from a set of vectors. This relates the algorithm to decoder and encoder networks. Therefore, we call the two-step algorithm a *linear algebra network*. Finally, we provide numerical simulations showing that the strategy is feasible for reconstructing spectral functions and for solving operator equations without explicitly making use of a physical model.

Key words. operator learning, singular value and function learning, data-driven reconstruction

AMS subject classifications. 65J20, 65J22, 15A18

1. Introduction.

We consider solving an ill-conditioned *linear operator equation*

$$F\mathbf{x} = \mathbf{y}, \tag{1.1}$$

where $\mathbf{x} \in \mathbb{R}^{\underline{m}}$ and $\mathbf{y} \in \mathbb{R}^{\overline{m}}$. We call $\mathbb{R}^{\underline{m}}$ the *image* and $\mathbb{R}^{\overline{m}}$ the *data space* following the terminology of [3, 4]. The main assumption of this work is that the operator F is *not* modeled mathematically by physical laws but indirectly via *training pairs*,

$$\mathcal{P} := \{(\mathbf{x}_i, \mathbf{y}_i) : i = 1, \dots, N\},$$

which satisfy

$$F\mathbf{x}_i = \mathbf{y}_i, \quad i = 1, \dots, N.$$

Here N is called the sampling size, and we denote the span of the training images and data by

$$X_N := \operatorname{span}\{\mathbf{x}_i : i = 1, \dots, N\} \subseteq \mathbb{R}^{\underline{m}} \equiv \mathbb{R}^{\underline{m}\times 1},$$
$$Y_N := \operatorname{span}\{\mathbf{y}_i : i = 1, \dots, N\} \subseteq \mathbb{R}^{\overline{m}} \equiv \mathbb{R}^{\overline{m}\times 1},$$

and

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_N \end{pmatrix} \in \mathbb{R}^{\underline{m}\times N}, \quad \mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 & \cdots & \mathbf{y}_N \end{pmatrix} \in \mathbb{R}^{\overline{m}\times N}.$$

Without further notice, we always assume that the training images $\mathbf{x}_i$ are linearly independent and that F has trivial nullspace, such that the training data $\mathbf{y}_i$ are also linearly independent.

*Received February 18, 2025. Accepted October 6, 2025. Published online on November 12, 2025. Recommended by Ronny Ramlau.

†Corresponding author. Università degli Studi di Milano Statale, Department of Mathematics "Federigo Enriques", Via Saldini 50, 20133 Milano, Italy (andrea.aspri@unimi.it).

‡Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, A-1090 Vienna, Austria.

§Johann Radon Institute for Computational and Applied Mathematics (RICAM), Altenbergerstraße 69, A-4040 Linz, Austria.

¶Christian Doppler Laboratory for Mathematical Modeling and Simulation of Next Generations of Ultrasound Devices (MaMSi), Oskar-Morgenstern-Platz 1, A-1090 Vienna, Austria.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Operator learning focuses on identifying or approximating the unknown transformation F. The problem involves finding a mapping between input functions $\mathbf{x}_i$ and corresponding outputs $\mathbf{y}_i$, for $i = 1, \dots, N$, that can also accurately predict outputs for unseen inputs. That is, we study *learning* the operator F and its inverse through an *encoding–decoding* strategy, where both steps are based solely on the available training data. After learning the operator F, we can solve (1.1) for arbitrary data $\mathbf{y} \in \mathbb{R}^{\overline{m}}$. This is different from the strategy proposed in [4, 19], where the operator equation is solved directly without learning the operator in a first step.

Operator learning is a very active field of research. There exist a variety of such methods, such as very established ones, like *black box* strategies (see, for instance, [32]) for linear operator learning. We remark that, in the literature, the term *operator learning* is most commonly associated with the approximation of infinite-dimensional operators, one of its main advantages being the independence from discretization [22]. However, this is only possible when the number of training samples $N \to \infty$. In other words the approach is discretization-independent but sampling-dependent.

Our setting here is more restricted: we directly approximate a linear discrete inverse problem in a finite-dimensional framework. Nevertheless, via regularization by projection, this finite-dimensional operator can be understood as the restriction of an infinite-dimensional operator to a finite-dimensional subspace (see [4, 19]). In this sense, our methodology should be regarded as a finite-dimensional learning strategy, conceptually closer to end-to-end methods in inverse problems [1]. Moreover, we note that the term *operator learning* is also frequently employed when referring to discretized realizations of infinite-dimensional operators [26], which is the context here and therefore justifies the used terminology.

For nonlinear operator learning, it is popular to use *deep neural networks* (see [21, 23, 24, 25]). This is typically realized using neural operators, which extend neural networks to handle function-based inputs and outputs. By discretizing functions at specific sensor locations, the model is parameterized and trained through an optimization process that minimizes the error between predicted and actual outputs. Key challenges include selecting a suitable model structure, managing computational costs, and ensuring strong generalization capabilities. For applications in inverse problems, see [1, 18].

Coding is a term used in *manifold learning*, which, in turn, is a basic tool in machine learning. The basic assumption there is that all potentially measured $\mathbf{y}$'s are elements of a *low-dimensional* manifold (see, for instance, [9, 13]). The setting of manifold learning (no operator connecting data) is represented in Figure 1.1.

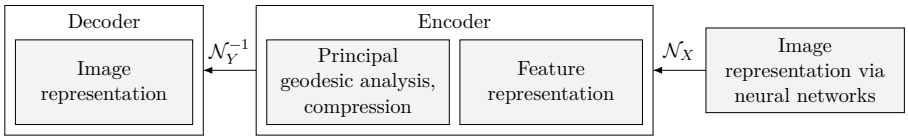


FIG. 1.1. *Variational encoding and decoding with neural networks. The image data are represented via a neural network. After representation they are transformed into a* feature *space (with the operator $\mathcal{N}_X$). The features are compressed by a principal geodesic analysis. The decoder $\mathcal{N}_Y^{-1}$ (we assume for the sake of simplicity that the operator is invertible) transforms features into images.*

Our approach for operator learning is conceptually akin to manifold learning, yet methodologically different (compare Figure 1.1 and Figure 1.2). We note that the relation between autoencoders and classical linear algebra methods has been studied in the literature before. In particular, it has been shown that linear autoencoders essentially perform principal component analysis (PCA). In [5] the authors demonstrated that a linear autoencoder trained to

ETNA
Kent State University and
Johann Radon Institute (RICAM)

reproduce its inputs learns to span the subspace of the leading principal components and that the associated optimization problem is free from spurious local minima. Independently, the authors of [8] established a connection between auto-associative multilayer perceptrons and the singular value decomposition (SVD), showing that hidden representations align with the principal components of the data. More recently, others [20] revisited autoencoder formulations in the context of inverse problems, proposing a paired autoencoder framework based on Bayes risk minimization. This approach illustrates how classical insights on the link between autoencoders, PCA, and SVD can be systematically extended to modern data-driven methodologies for inverse problems.

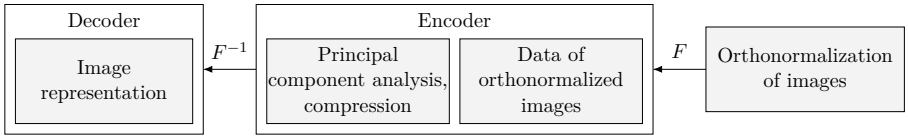


FIG. 1.2. *Encoding and decoding scheme for* linear *operators. First the image data are orthonormalized and the corresponding data are computed by applying F—this is done by explicit calculations without making use of any physical model describing the forward operator (see Section 2 for details). On the orthonormalized data, a principal component analysis (PCA) is applied, which allows the data space to be compressed. The decoder calculates the inverse of some given data in the compressed space.*

In this paper, we investigate a new coding scheme for solving *linear ill-conditioned problems*, as outlined in Figure 1.2. The standard linear algebra approach for linear operator learning consists of computing the linear least-squares problem for (see [35])

$$\operatorname{argmin}_{F\in\mathcal{L}(\mathbb{R}^m,\mathbb{R}^{\overline{m}})\equiv\mathbb{R}^{\overline{m}\times m}} \|\mathbf{Y}-F\mathbf{X}\|^2_{\mathbb{R}^{\overline{m}\times N}}.$$

For this linear optimization problem, the minimizing operator F is given by

$$F_{ls} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^\dagger,$$

where $(\mathbf{X}\mathbf{X}^T)^\dagger$ denotes the Moore–Penrose inverse (see [28]) of $\mathbf{X}\mathbf{X}^T$. In general the matrices $\mathbf{X}\mathbf{X}^T$ are huge, and more memory-efficient techniques can be implemented, which are based on orthonormalization; we use orthonormalization as a key tool for operator learning (see, for instance, [37]). This paper is based on the following two interesting observations and results:

1. The algorithm shown in Figure 1.2 allows one to determine the singular vectors and values of F (see Theorem 2.6); note that the operator is not physically modeled but computed through its input–output relations (training pairs). In other words, the choice of the training pairs only influences which spectral vectors and values can be computed. While the result follows from elementary linear algebra considerations, to the best of our knowledge it has not yet been observed in the context of operator learning. One can therefore say that the singular value decomposition is invariant with respect to the choice of training pairs.
2. Related to operator learning are *regularization issues*. In [3, 4], we showed that orthonormalization of training data or training images (in a separate manner) can be used to stably determine an operator if the orthonormalization produces a basis that is close to the singular vectors. There is an antisymmetry in the two approaches, which is overcome in this paper. In this context, regularization and stability analysis of the implementation should not be confused. For instance, as used here, Gram–Schmidt orthonormalization is unstable with an increased number of expert images and data, respectively. However, regularization results apply if pre-processing orthonormalization methods (like Gram–Schmidt) have been implemented in a stable manner—this

ETNA
Kent State University and
Johann Radon Institute (RICAM)

is what we discuss here. In view of this, the numerical results need to be evaluated (see Section 4.2). The inaccuracies are due to numerical instabilities of orthonormalization algorithms and not because of regularization by truncated SVD. We need to further analyze whether the combined strategy of Gram–Schmidt orthonormalization and PCA offers practical benefits versus orthonormalization of either training images or data.

3. Moreover, we show that each building block of the proposed algorithm, represented in Figure 1.2 (in particular, orthonormalization) can be expressed via a customized *linear algebra network* (see Figure 2.1 in Section 2.1). The term *customized* refers to the fact that the parameters in the neural network are given by the algorithm (specifically, Gram–Schmidt orthonormalization and PCA) and do not need to be optimized from the training data.
4. Finally, we present some numerical experiments on learning the Radon operator (the source code is referenced and available in [2]). For this operator, the singular values are explicitly known (see [12, 29]). Therefore, we can compare the computed singular values from training data with the analytical ones (see Section 4). Although it is theoretically possible to recover singular vectors, practical handicaps are due to instabilities of the orthonormalization algorithms, as discussed already above.

We summarize the main objectives of this paper: We show that finite-dimensional linear operators can be learned with the proposed hybrid orthonormalization strategy. In contrast, several methods for operator learning based on decode and encoder networks have been developed recently. We show that our hybrid linear algebra technology can indeed be written as a decoder and encoder network. In view of this, our paper provides a systematic link between these two areas of research. Moreover, our analysis provides an insight into the efficiency and stability of neural network coders.

2. Encoding of linear operators.

The encoder from Figure 1.2 consists of two steps:

1. Calculating orthonormalized images.
2. Computing a principal component of data of orthonormalized images

$$\overline{\mathbf{y}}_j := F(\overline{\mathbf{x}}_j), \quad j = 1, \dots, N.$$

We recall that, according to our general assumptions on training images and data, we have $N \leq \{\underline{m}, \overline{m}\}$. The assumption can be neglected at several places, but we leave it for the sake of simplicity.

2.1. Pre-processing—orthonormalization.

We review the Gram–Schmidt orthonormalization method (see, for instance, [17]) and show that it can be expressed as a deep neural network (NN). In a second step, we show that also the QR decomposition (see [10]), realizing PCA, is representable as a deep NN. In other words, they are *customized* because the parameters of the network do not need to be optimized. This motivates our terminology *linear algebra networks*.

2.1.1. Gram–Schmidt as a deep network.

We start with the training images $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^{\underline{m}}$ and orthonormalize them iteratively. Let

$$\begin{aligned} \boldsymbol{\sigma} : \mathbb{R}^{\underline{m}} \setminus \{0\} &\to \mathbb{R}^{\underline{m}}, \\ \mathbf{x} &\mapsto \frac{\mathbf{x}}{\|\mathbf{x}\|}, \end{aligned}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ denote the Euclidean norm and inner product in $\mathbb{R}^{\underline{m}}$ and $\mathbb{R}^{\overline{m}}$, respectively. Then Gram–Schmidt orthonormalization can be written as follows:

$$\overline{\mathbf{x}}_j := \boldsymbol{\sigma}\Bigg(\underbrace{\mathbf{x}_j - \sum_{i=1}^{j-1}\langle \mathbf{x}_j, \overline{\mathbf{x}}_i\rangle \overline{\mathbf{x}}_i}_{=:\,\boldsymbol{\rho}(\mathbf{x}_j)}\Bigg) \quad \text{for all } j = 1,\dots,N. \tag{2.1}$$

For each vector $j = 1,\dots,N$, we treat $\mathbf{x}_j$ as the input to the jth layer of a neural network. Here $\boldsymbol{\rho}(\mathbf{x}_j)$ is a linear, vector-valued function, and therefore every one of its components can be represented by a standard affine linear neural network. Since the vectors $\overline{\mathbf{x}}_i$ are provided as inputs from the previous layers, they do not interfere with the computations in the jth layer. This is also represented in the sparse graph representing the network in Figure 2.1.

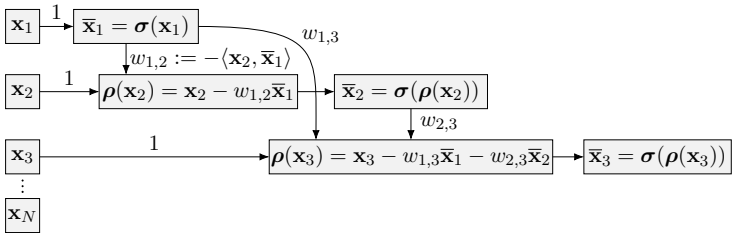


FIG. 2.1. *The neural network structure of the Gram–Schmidt orthonormalization. Here $w_{i,j}$ represent customized weights.*

REMARK 2.1. In practical applications the smooth approximation

$$\begin{aligned}\boldsymbol{\sigma}_\varepsilon : \mathbb{R}^{\underline{m}} &\to \mathbb{R}^{\underline{m}},\\ \mathbf{x} &\mapsto \frac{\mathbf{x}}{\sqrt{\|\mathbf{x}\|^2 + \varepsilon^2}},\end{aligned}$$

is used. With this modification, we can avoid the Gram–Schmidt orthonormalization breaking down with linearly dependent training images $\mathbf{x}_i$. In the machine learning context, the approximations $\boldsymbol{\sigma}_\varepsilon$ of the high-dimensional sign function $\boldsymbol{\sigma}$ are called *activation functions*. Note, however, that activation functions are typically scalar maps applied component-wise to vectors (see, for instance, [7]). In contrast, in our setting the activation acts on the entire vector and is therefore vector-valued; to make this distinction explicit, we denote it in boldface as $\boldsymbol{\sigma}_\varepsilon : \mathbb{R}^{\underline{m}} \to \mathbb{R}^{\underline{m}}$. Such a component-wise network cannot be used here because we must ensure that every $\boldsymbol{\rho}(\mathbf{x}_j)$, $j = 1,\dots,N$, is a linear combination of $\mathbf{x}_i$, $i = 1,\dots,j-1$. Replacing $\boldsymbol{\sigma}$ by $\boldsymbol{\sigma}_\varepsilon$ in Figure 2.1, we obtain an N-layer neural network, which we call a *Gram–Schmidt network*. There are efficient alternatives to Gram–Schmidt, which are, for instance, block-based (see, for instance, [11]), which can be reinterpreted again as deep neural networks.

2.2. Data from orthonormalized images. If the nullspace of F is trivial, then from (2.1) it follows immediately that (see Figure 2.1)

$$\overline{\mathbf{y}}_j := F\overline{\mathbf{x}}_j := F(\boldsymbol{\sigma}(\boldsymbol{\rho}(\mathbf{x}_j))) = \frac{1}{\|\boldsymbol{\rho}(\mathbf{x}_j)\|} F(\boldsymbol{\rho}(\mathbf{x}_j)) \quad \text{for all } j = 1,\dots,N. \tag{2.2}$$

This means that the data of orthonormalized images can be computed without explicit knowledge of F. Note that $\boldsymbol{\rho}(\mathbf{x}_j)$ is a linear combination of $\mathbf{x}_i$, $i = 1,\dots,j$, where the coefficients of the linear map are given from previous layers.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

With the knowledge of $\overline{\mathbf{y}}_j$, $j = 1, \dots, N$, we can compute data for every image $\mathbf{x} \in X_N$. Since

$$\mathbf{x} = \sum_{i=1}^{N} \langle \mathbf{x}, \overline{\mathbf{x}}_i \rangle \overline{\mathbf{x}}_i,$$

it follows that

$$\mathbf{y} = F\mathbf{x} = \sum_{i=1}^{N} \langle \mathbf{x}, \overline{\mathbf{x}}_i \rangle F\overline{\mathbf{x}}_i = \sum_{i=1}^{N} \langle \mathbf{x}, \overline{\mathbf{x}}_i \rangle \overline{\mathbf{y}}_i.$$

REMARK 2.2. If, instead of the exact Gram–Schmidt procedure, one considers the so-called Gram–Schmidt network obtained by replacing $\boldsymbol{\sigma}$ with $\boldsymbol{\sigma}_\varepsilon$ in Figure 2.1, then the approximate relation

$$\overline{\mathbf{y}}_j := F\overline{\mathbf{x}}_j \approx F(\boldsymbol{\sigma}_\varepsilon(\boldsymbol{\rho}(\mathbf{x}_j))) =: \frac{1}{\|\boldsymbol{\sigma}_\varepsilon(\boldsymbol{\rho}(\mathbf{x}_j))\|} F(\boldsymbol{\rho}(\mathbf{x}_j)), \quad j = 1, \dots, N, \tag{2.3}$$

is made use of. We note, however, that in our numerical simulations we always employ the classical Gram–Schmidt algorithm, and not the Gram–Schmidt network. The network formulation is introduced here only to highlight the connection with neural network structures and to provide a conceptual interpretation of orthonormalization in terms of customized networks.

2.2.1. Principal component analysis (PCA). In the following, we analyze the stability of PCA in the regularization context, as mentioned in the introduction. The starting point of this discussion is *Seidman's veto* [33], which states that regularization by projection is in general not a regularization method. This means that, when projecting onto X_S, the inversion of F restricted to $F(X_S)$ becomes unstable as $S \to \infty$. On the other hand, if F is inverted on Y_S, then it is in fact stable (see again [33]). However, in the context of machine learning, this requires the collection of training data of $F^*\mathbf{y}_i$, $i = 1, \dots, S$, which in general is not available. Note that F^* denotes the adjoint of F. As a conclusion from [33] we find that F can be stably inverted by projection if X_S is the space of the singular vectors corresponding to the largest S singular values (see, for instance, [14]). Now, we show how these singular values can be calculated with machine learning techniques.

THEOREM 2.3 (Spectral theory: see [17, Theorem 2.5.2]). *Let the operator $F : \mathbb{R}^{\underline{m}} \to \mathbb{R}^{\overline{m}}$ be linear. Then for every $\mathbf{x} \in \mathbb{R}^{\underline{m}}$*

$$F\mathbf{x} = \sum_{j=1}^{\min\{\underline{m},\overline{m}\}} \gamma^j \langle \mathbf{x}, \mathbf{u}^j \rangle \mathbf{v}^j \quad \textit{and} \quad F^T\mathbf{v}^j = \gamma^j \mathbf{u}^j, \quad F\mathbf{u}^j = \gamma^j \mathbf{v}^j, \tag{2.4}$$

where $\mathbf{u}^j \in \mathbb{R}^{\underline{m}}$, $j = 1, \dots, \underline{m}$, and $\mathbf{v}^j \in \mathbb{R}^{\overline{m}}$, $j = 1, \dots, \overline{m}$, are orthonormal, respectively, and

$$0 \le \gamma^1 \le \gamma^2 \le \cdots \le \gamma^{\min\{\underline{m},\overline{m}\}}.$$

In matrix form, this identity becomes more compact:

$$F = \mathbf{V}D\mathbf{U}^T \quad \textit{with} \quad \mathbf{U} \in \mathbb{R}^{\underline{m}\times\underline{m}}, \quad \mathbf{V} \in \mathbb{R}^{\overline{m}\times\overline{m}}, \tag{2.5}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal and

$$D = \operatorname{diag}(\gamma^1, \gamma^2, \dots, \gamma^{\min\{\underline{m},\overline{m}\}}, 0, \dots, 0) \in \mathbb{R}^{\overline{m}\times\underline{m}}.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

In the following, we group the spectral values:

DEFINITION 2.4. *Let F be linear with trivial nullspace with spectral decomposition as in (2.4). We denote by*

$$\Gamma := \{\gamma^j : j = 1, \dots, \min\{\underline{m}, \overline{m}\}\} = \{\hat{\gamma}^k : k = 1, \dots, \hat{m}\} \tag{2.6}$$

the set of distinct singular values. Because F is assumed to have trivial nullspace, $\gamma^j > 0$ for all $j = 1, \dots, \min\{\underline{m}, \overline{m}\}$. Moreover, we associate to each multiple singular value the associated singular vectors: For every $k = \{1, \dots, \hat{m}\}$ let

$$E^k := \operatorname{span}\{\mathbf{v}^j : \gamma^j = \hat{\gamma}^k, j = 1, \dots, \min\{\underline{m}, \overline{m}\}\}. \tag{2.7}$$

Now, we apply the spectral theory to show that the proposed decoding algorithm is stable. There, we adopt some general notation:

DEFINITION 2.5. *Let $\overline{\mathbf{x}}_i$, $i = 1, \dots, N$, be the orthonormalized training images and $\overline{\mathbf{y}}_i$, $i = 1, \dots, N$, be the corresponding images as defined in (2.1) and (2.2), respectively. We denote*

$$\begin{aligned} &\overline{\mathbf{X}} := (\overline{\mathbf{x}}_1, \dots, \overline{\mathbf{x}}_N) \in \mathbb{R}^{\underline{m} \times N}, \quad \overline{\mathbf{Y}} := (\overline{\mathbf{y}}_1, \dots, \overline{\mathbf{y}}_N) \in \mathbb{R}^{\overline{m} \times N}, \quad \text{and} \\ &\mathbf{A} := \overline{\mathbf{Y}}\,\overline{\mathbf{Y}}^T \in \mathbb{R}^{\overline{m} \times \overline{m}}. \end{aligned} \tag{2.8}$$

Note that, by our general assumptions, the rank of each of the three matrices is always $N \leq \overline{m}$.

With this result, we can state the main result of this paper:

THEOREM 2.6. *Let $\boldsymbol{\psi}_j \in \mathbb{R}^{\overline{m}}$, $j = 1, \dots, N$, be a non-zero eigenvector of $\mathbf{A}$. Then*

1. *there exists $k \in \{1, \dots, \hat{m}\}$ such that $\boldsymbol{\psi}_j \in E^k$, and thus*
2. *in particular, this means that, if a non-zero eigenvector of $\mathbf{A}$ has multiplicity one, then it equals a spectral vector of F.*

Proof. The PCA in image space calculates the eigenvalue decomposition of the covariance matrix

$$\mathbf{A} := \overline{\mathbf{Y}}\,\overline{\mathbf{Y}}^T = \overline{\mathbf{W}} \underbrace{\begin{bmatrix} \Lambda \in \mathbb{R}^{N \times N} & 0 \\ 0 & 0 \end{bmatrix}}_{=: \overline{\Lambda}} \overline{\mathbf{W}}^T \in \mathbb{R}^{\overline{m} \times \overline{m}}, \tag{2.9}$$

where $\overline{\mathbf{W}} \in \mathbb{R}^{\overline{m} \times \overline{m}}$ is an orthonormal matrix, $\overline{\mathbf{W}}$ describes the principal directions of the data $\overline{\mathbf{Y}}$, and the corresponding entries of Λ describe the elongation of the data in this direction. Now, since $\overline{\mathbf{Y}} = F\overline{\mathbf{X}}$, we get from (2.9)

$$F\overline{\mathbf{X}}\,\overline{\mathbf{X}}^T F^T = \overline{\mathbf{Y}}\,\overline{\mathbf{Y}}^T = \mathbf{A} = \overline{\mathbf{W}}\,\overline{\Lambda}\,\overline{\mathbf{W}}^T. \tag{2.10}$$

Since $\overline{\mathbf{X}}$ consists of orthonormal vectors and has rank N, we have

$$\overline{\mathbf{X}}\,\overline{\mathbf{X}}^T = \begin{bmatrix} \mathcal{I} \in \mathbb{R}^{N \times N} & 0 \\ 0 & 0 \end{bmatrix} =: \overline{\mathcal{I}} \in \mathbb{R}^{\underline{m} \times \underline{m}},$$

where $\mathcal{I}$ is the identity matrix. Let $\mathbf{U} \in \mathbb{R}^{\underline{m} \times \underline{m}}$ be the orthonormal matrix from (2.5). Then, after reordering of columns of $\overline{\mathbf{X}}$,

$$\mathbf{U}^T \overline{\mathbf{X}}^T \overline{\mathbf{X}} \mathbf{U} = \mathbf{U}^T \overline{\mathcal{I}} \mathbf{U} \in \mathbb{R}^{\underline{m} \times \underline{m}}.$$

The matrix $\overline{\mathcal{I}}\mathbf{U}$ projects onto the first basis vectors $\mathbf{u}_i$ (after potential reordering of the vectors). Then we get from (2.10) and (2.5) the identity

$$\overline{\mathbf{W}}\,\overline{\Lambda}\,\overline{\mathbf{W}}^T = \overline{\mathbf{Y}}\,\overline{\mathbf{Y}}^T = F\overline{\mathbf{X}}\,\overline{\mathbf{X}}^T F^T = \mathbf{V} D \overline{\mathcal{I}} D^T \mathbf{V}^T.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

This means that we have found two singular value decompositions of $\overline{\mathbf{Y}}\overline{\mathbf{Y}}^T$. We know that the matrices on the left and right have the same eigenspaces. This means, in particular, that if an eigenvalue has multiplicity one, then the corresponding eigenspaces of $\mathbf{V}$ and $\mathbf{W}$ match, and the representing eigenvectors of the eigenspace are identical up to the sign. This proves the second item. For eigenvalues of multiplicity higher than one, the corresponding columns of $\mathbf{V}$ are rotations and mirrors of $\overline{\mathbf{W}}$, which proves the first item. □

REMARK 2.7. Theorem 2.6 tells us the following:

1. Orthonormalization of the training images is the basis of stable decoding. The difficulty, however, is that orthonormalization may not provide us with an ordering of the singular vectors by absolute magnitude. For instance, there is no guarantee that singular vectors belonging to the absolutely largest singular values, which usually carry most of the information, can be recovered. See Table 4.1, which shows this issue for an example related to the Radon transform: The choice of the training data does not allow one to reconstruct some low-frequency components.
2. Orthonormalization is unstable. Thus, N must be chosen small to guarantee numerical stability. We emphasize again that we do not discuss the instability of orthonormalization algorithms. Instead we discuss regularization aspects related to the truncated singular value decomposition, which is determined via hybrid orthonormalization.

Last, we also verify that PCA, which is implemented via QR decomposition, can be approximately rewritten as a network, analogously to the Gram–Schmidt algorithm in Figure 2.1.

2.3. PCA expressed as a deep network. This section shows that spectral value decomposition can be expressed as a customized deep network. Thereby, we make use of the fact that *singular value decomposition* (SVD) applied to the covariance matrix $\mathbf{A} = \overline{\mathbf{Y}}\overline{\mathbf{Y}}^T \in \mathbb{R}^{\overline{m}\times\overline{m}}$ from (2.4) gives the principal components. The SVD can be implemented by iteratively applying the QR algorithm, which is the famous *Francis algorithm* [16] (see also [17, 39]). Now, we show that the QR algorithm can be approximated by a deep linear algebra network. Therefore, approximating the QR decomposition in the Francis algorithm with linear algebra networks gives a "*deep–deep*" linear algebra network realizing the PCA. The terminology deep–deep refers to the implementation of the Francis algorithm, which is iterative, and in each iteration step of the Francis algorithm a QR decomposition is implemented, which is represented as a deep neural network. When the single iterates of the Francis algorithm are represented with a neural network graph structure, we have one level of deepness more (for the iterates of the Francis algorithm).

Let us denote the column vectors of the matrix $\mathbf{A}$, defined in (2.8), by $\mathbf{a}_i$, $i = 1, \dots, \overline{m}$, and denote the orthonormalized vectors by $\overline{\mathbf{a}}_i$, $i = 1, \dots, \overline{m}$. Moreover, let

$$\boldsymbol{\rho}(\mathbf{a}_j) := \mathbf{a}_j - \sum_{i=1}^{j-1} \langle \mathbf{a}_j, \overline{\mathbf{a}}_i \rangle \overline{\mathbf{a}}_i.$$

Then it follows (see [34], which is the textbook where the formulas are written precisely in the same way as they are used here) that:

$$\mathbf{a}_j = \|\boldsymbol{\rho}(\mathbf{a}_j)\| \overline{\mathbf{a}}_j + \sum_{i=1}^{j-1} \langle \mathbf{a}_j, \overline{\mathbf{a}}_i \rangle \overline{\mathbf{a}}_i, \quad j = 1, \dots, \overline{m}. \tag{2.11}$$

Writing this in matrix notation reads as follows:

$$\mathbf{A} = QR \quad \text{with } Q = (\overline{\mathbf{a}}_1, \dots, \overline{\mathbf{a}}_n) \text{ and } R_{i,j} = \begin{cases} \langle \mathbf{a}_j, \overline{\mathbf{a}}_i \rangle & \text{for } i = 1, \dots, j-1, \\ \|\boldsymbol{\rho}(\mathbf{a}_j)\| & \text{for } i = j, \\ 0 & \text{for } i = j+1, \dots, \overline{m}. \end{cases}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

In fact, it follows from (2.11) that

$$\overline{\mathbf{a}}_j = \frac{1}{\|\boldsymbol{\rho}(\mathbf{a}_j)\|}\left(\mathbf{a}_j - \sum_{i=1}^{j-1}\langle \mathbf{a}_j, \overline{\mathbf{a}}_i\rangle \overline{\mathbf{a}}_i\right) = \boldsymbol{\sigma}(\boldsymbol{\rho}(\mathbf{a}_j)), \quad j = 1, \ldots, \overline{m},$$

Figure 2.1 with $\mathbf{x}$ replaced by $\mathbf{a}$ is the net that determines Q. This shows that the matrix Q from the QR algorithm can be determined with a *linear algebra network*. Moreover, it has been recently observed that matrix multiplication can be implemented very efficiently via reinforcement learning or, in other words, via neural networks (see [15]). We remark here that [38, Section 35] have already mentioned that the QR algorithm based on Gram–Schmidt orthonormalization may lead to unstable implementations. Thus, the exposition above is only used for theoretical purposes.

REMARK 2.8. In recent years PCA networks have become popular in the machine learning community (see [6, 31, 36]). The goal there is somewhat different and consists of minimizing the parameters of a network. While such networks would be worthwhile to investigate in this context, a detailed analysis falls outside the scope of this work.

3. Image representation and decoder. We consider the decoder as represented in Figure 1.2. We calculate the least-squares approximation of $\mathbf{y}$ with respect to the basis $\{\boldsymbol{\psi}_l : l = 1, \ldots, N\}$ (see Theorem 2.6) of the eigenvectors of $\mathbf{A}$. This is given by

$$\mathbf{y}_{ls} = \sum_{l=1}^{N} \langle \mathbf{y}, \boldsymbol{\psi}_l\rangle \boldsymbol{\psi}_l.$$

The goal is to find an explicit representation of the *decoding function* $\mathbf{x}_{ls}$, that is, a function with minimum norm, which satisfies

$$F\mathbf{x}_{ls} = \mathbf{y}_{ls} \tag{3.1}$$

and which can be represented by the training pairs alone.

From Theorem 2.6 it follows that, for every $\boldsymbol{\psi}_l$, there exists $k(l) \in \mathbb{N}$ such that

$$\boldsymbol{\psi}_l \in E^{k(l)}.$$

We denote the dimension of $E^{k(l)}$ by $m(l)$. In other words, the multiplicity of γ_l is $m(l)$. From (2.7) we know that

$$\boldsymbol{\psi}_l \in \operatorname{span}\{\mathbf{v}^{l(j)} : j = 1, \ldots, m(l)\} = E^{k(l)},$$

and thus

$$\boldsymbol{\psi}_l = \sum_{j=1}^{m(l)} \nu_j \mathbf{v}^{l(j)},$$

where $\mathbf{v}^{l(j)}$ denotes the $l(j)$th row of the matrix $\mathbf{V}$ of the singular value decomposition of F (see (2.5)). Applying (2.4) and (2.7) then implies that

$$\boldsymbol{\psi}_l = \sum_{j=1}^{m(l)} \nu_j \mathbf{v}^{l(j)} = \frac{1}{\hat{\gamma}^{k(l)}} F\left(\sum_{j=1}^{m(l)} \nu_j \mathbf{u}^{l(j)}\right).$$

We summarize the calculations now in the following lemma:

ETNA
Kent State University and
Johann Radon Institute (RICAM)

LEMMA 3.1 (Decoder). *The minimum-norm solution of* (3.1) *is given by the* decoder

$$\begin{aligned}\mathbf{x}_{ls} &= \sum_{l=1}^{N} \langle \mathbf{y}, \boldsymbol{\psi}_l \rangle \left(\frac{1}{\hat{\gamma}^{k(l)}} \sum_{j=1}^{m(l)} \nu_j \mathbf{u}^{l(j)} \right) = \sum_{l=1}^{N} \frac{\langle \mathbf{y}, \boldsymbol{\psi}_l \rangle}{(\hat{\gamma}^{k(l)})^2} F^* \left(\sum_{j=1}^{m(l)} \nu_j \mathbf{v}^{l(j)} \right) \\ &= \sum_{l=1}^{N} \frac{\langle \mathbf{y}, \boldsymbol{\psi}_l \rangle}{(\hat{\gamma}^{k(l)})^2} F^* \boldsymbol{\psi}_l \, .\end{aligned} \tag{3.2}$$

In summary, the necessary implementation steps to compute the least-squares solution $\mathbf{x}_{ls}$ are: First, QR decomposition is implemented to get the eigenfunctions $\{\boldsymbol{\psi}_l : l = 1, \dots, N\}$. Second, the decoder is implemented via (3.2).

4. Numerical simulations. We consider two simulation scenarios: First, we show that the singular value decomposition can indeed be computed from training pairs without knowing the forward operator explicitly. Second, we study a reconstruction test with a learned operator. In both cases, we use the Radon transform for two-dimensional images as the physical model for the forward operator, leveraging its analytically known singular value decomposition (see [12, 29]).

4.1. SVD from training pairs. We verify numerically that orthonormalization of data ($\mathbf{x}_i \to \overline{\mathbf{x}}_i$, $i = 1, \dots, N$) and a principal component analysis of the covariance matrix $\mathbf{A}$, defined in (2.8) obtained from the images of the orthonormalized data,

$$\{\overline{\mathbf{y}}_i = F\overline{\mathbf{x}}_i : i = 1, \dots, N\},$$

see (2.2) and (2.3), provides us with the singular value decomposition of the discrete operator F (see Figure 1.2).

We take as a prototype example for F the Radon transform in two dimensions (see Definition 4.1 below). The singular value decomposition of the Radon transform has been computed in [12] (see also [29]) for the general case of images of n variables. In our example $n = 2$, and we consider the Radon transform as an operator from $L^2(\mathcal{B}(0,1))$ into $L^2(Z, w^{-1})$, where $\mathcal{B}(0,1)$ is the unit disk in $\mathbb{R}^2$ centered at the origin, where

$$Z = \mathbb{S}^1 \times [-1, 1] \quad \text{and} \quad w(s) = (1 - s^2)^{1/2}$$

is a weight function. This means that $L^2(Z, w^{-1})$ is the weighted L^2-space with norm

$$\|g\|^2_{L^2(Z,w^{-1})} = \int_{\mathbb{S}^1} \int_{-1}^{1} |g(\omega, s)|^2 w(s)^{-1} \, ds \, d\omega$$

for $g \in L^2(Z, w^{-1})$. Before going into the numerical details, we provide the necessary notation and summarize well-known results on the Radon transform to facilitate the analysis.

4.1.1. Convention. Given $\vec{\omega} = (\omega_1, \omega_2)^T \in \mathbb{S}^1$ we have

$$\omega_1 = \cos(\phi) \quad \text{and} \quad \omega_2 = \sin(\phi), \quad \phi \in [0, 2\pi).$$

Moreover, the orthogonal vector related to $\vec{\omega}$ is given by $\vec{\omega}^{\perp} = (-\sin(\phi), \cos(\phi))^T$.

For the reader's convenience, we recall the definition of the Radon transform for functions on the unit disk in $\mathbb{R}^2$:

ETNA
Kent State University and
Johann Radon Institute (RICAM)

DEFINITION 4.1 (Radon transform on the unit disk). *Let $Z := \mathbb{S}^1 \times [-1,1]$. We define the Radon transform as*

$$R : L^2(\mathcal{B}(0,1)) \to L^2(Z, w^{-1}),$$
$$f \mapsto R[f](\vec{\omega}, s) = \int_{-\sqrt{1-s^2}}^{\sqrt{1-s^2}} f(s\vec{\omega} + t\vec{\omega}^{\perp})\, dt.$$

One can show that this operator is continuous and satisfies (see [12]):

$$\|R[f]\|_{L^2(Z,w^{-1})} \leq \sqrt{4\pi}\, \|f\|_{L^2(\mathcal{B}(0,1))}.$$

In the following, we recall the expression of the adjoint operator of the Radon transform R defined on the unit disk. This is the operator $R^* : L^2(Z, w^{-1}) \to L^2(\mathcal{B}(0,1))$, which satisfies

$$\langle R[f], g\rangle_{L^2(Z,w^{-1})} = \langle f, R^*[g]\rangle_{L^2(\mathcal{B}(0,1))}.$$

DEFINITION 4.2 (Adjoint). *For every $g \in L^2(Z, w^{-1})$ and almost all $\vec{x} = (x_1, x_2)^T \in \mathbb{R}^2$, the adjoint of the Radon transform R^* is given by*

$$R^*[g](\vec{x}) = \int_{\mathbb{S}^1} \frac{g(\vec{\omega}, \vec{x}\cdot\vec{\omega})}{w(\vec{x}\cdot\vec{\omega})}\, d\vec{\omega}.$$

In the following we give a survey on the singular value decomposition of the Radon transform for functions on the unit disk. The results are essentially taken from [29], with the main difference being that we consider the adjoint R^* restricted to the range of R (in contrast to Definition 4.2).

Before recalling the spectral decomposition of the Radon transform, we review the general definition of a spectral decomposition:

DEFINITION 4.3 (Spectral decomposition, [14]). *Let $K : X \to Y$ be a compact linear operator. A* singular system *$(u_k, v_k; \gamma_k)$ is defined as follows:*

1. *γ_k^2, $k \in \mathbb{N}_0$, are the non-zero eigenvalues of the self-adjoint operator K^*K (and also KK^*) written in decreasing order (we always take $\gamma_k > 0$);*
2. *u_k, $k \in \mathbb{N}_0$, are a complete orthonormal system of eigenvectors of K^*K (on the space $\overline{\mathcal{R}(K^*)} = \overline{\mathcal{R}(K^*K)}$);*
3. *$v_k := (1/\|Ku_k\|)Ku_k$, $k \in \mathbb{N}_0$.*

The set $\{v_k : k \in \mathbb{N}_0\}$ is a complete orthonormal system of eigenvectors of KK^* which spans $\overline{\mathcal{R}(K)} = \overline{\mathcal{R}(KK^*)}$. Moreover, the following formulas hold:

$$Ku_k = \gamma_k v_k, \quad K^*u_k = \gamma_k v_k,$$
$$Kx = \sum_{k=0}^{\infty} \gamma_k \langle x, u_k\rangle v_k \quad \text{for all } x \in X,$$
$$K^*y = \sum_{k=0}^{\infty} \gamma_k \langle y, v_k\rangle u_k \quad \text{for all } y \in Y.$$

Note that here x, y, u_k, and v_k are functions, while in our discrete setting we consider vectors $\mathbf{x}$, $\mathbf{y}$, $\mathbf{u}_k$, and $\mathbf{v}_k$.

THEOREM 4.4 ([29, p. 99]). *The spectral decomposition of the Radon transform is given by $\{(\mathbf{u}_{k,l}, \mathbf{v}_{k,l}; \gamma_{k,l}) : (k,l) \in \mathcal{I}\}$ where the following hold:*

1. *$\mathcal{I} = \{(k,l) : k \in \mathbb{N}_0,\ l \in \{0, 1, \ldots, k\},$ satisfying $l + k$ is even$\}$.*
2. *$\gamma_k^2 = \gamma_{k,l}^2 = 4\pi/(k+1) > 0$ is independent of l.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

3. *Let Y_s be the spherical harmonics in $\mathbb{R}^2$ (which in $n = 2$ are just* sin *and* cos *functions, with specified frequencies) and let*

$$s \mapsto C_k(s) = \frac{\sin((k+1)\arccos(s))}{\sin(\arccos(s))}$$

denote the Chebyshev polynomials of the second kind. Moreover, let

$$\frac{1}{c_k} = \|wC_kY_{k-2l}\|_{L^2(Z,w^{-1})} = \|\sqrt{w}C_k\|_{L^2(-1,1)}.$$

Then the normalized eigenfunctions of RR^* on the orthogonal complement of the nullspace of R^*, $\mathcal{N}^\perp$, are given by

$$(s,\omega) \to v_{k,l}(s,\omega) := c_k w(s) C_k(s) Y_{k-2l}(\omega) \quad \textit{and} \quad u_{k,l} = \frac{1}{\gamma_k} R^*[v_{k,l}]\,. \tag{4.1}$$

The singular value decomposition of the Radon transform has been computed in several works (see, for instance, [12, 27, 29] and [30, Theorem 6.4] to mention but a few).

REMARK 4.5.

1. The existence of a singular value decomposition with $\gamma_k \to 0$ in particular shows that the Radon transform is compact.
2. From Theorem 4.4, item 1 and item 2, we see that, in general, γ_k has multiplicity higher than one. The sets E^k associated to a spectral value γ_k are spanned by the spectral functions $v_{k,l}$ with l such that $(k,l) \in \mathcal{I}$. For example, taking $k = 0,\dots,7$, and $l = 0,\dots,k$, with $k+l$ even (meaning that $(k,l) \in \mathcal{I}$), we have

$$\begin{aligned} E^1 &= \operatorname{span}\{v_{0,0}\}, & E^2 &= \operatorname{span}\{v_{1,1}\},\\ E^3 &= \operatorname{span}\{v_{2,0}, v_{2,2}\}, & E^4 &= \operatorname{span}\{v_{3,1}, v_{3,3}\},\\ E^5 &= \operatorname{span}\{v_{4,0}, v_{4,2}, v_{4,4}\}, & E^6 &= \operatorname{span}\{v_{5,1}, v_{5,3}, v_{5,5}\},\\ E^7 &= \operatorname{span}\{v_{6,0}, v_{6,2}, v_{6,4}, v_{6,6}\}, & E^8 &= \operatorname{span}\{v_{7,1}, v_{7,3}, v_{7,5}, v_{7,7}\}. \end{aligned}$$

It is an interesting fact stated in [29] that the adjoint of the Radon transform R^* has a nullspace. In fact, the nullspace of R^* is given by

$$\mathcal{N} = \{v_{k,l} : k \in \mathbb{N}_0,\ l \in \{0,1,\dots,k\},\ \text{satisfying } l+k \text{ is odd}\}.$$

Moreover, RR^* has the same nullspace as R^*, which is shown for instance in [29, p. 99].

4.2. Numerical simulations. In this section, we present two numerical experiments:

1. The first tests concern computing the SVD of the Radon transform in two test scenarios:
   (a) *Learning spectral functions from analytical spectral functions.* We use as training data $(\mathbf{x}_i, \mathbf{y}_i)$, $i = 1,\dots,49$, the first 49 analytical, given, spectral functions $(u_{k,l}, v_{k,l})$, defined in (4.1). We compare the outcome with our orthonormalization approach. We recall that $(\mathbf{x}_i, \mathbf{y}_i)$ are vectors, which are considered discretized functions/images, and then compared with functions $(u_{k,l}, v_{k,l})$.
   (b) *Learning spectral functions from arbitrary test functions.* Here we use annotations of images, which are orthonormalized. This is of course a much harder problem.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

2. We are concerned with decoding by making use of (3.2). Here we use both singular value decompositions obtained in the first step.

All codes are publicly available at the following link:
https://gitlab.com/csc1/learning-op.

4.2.1. Singular value decomposition. We visualize the first 20 benchmark images, which are the singular functions $u_{k,l}$ and $v_{k,l}$ with $(l,k) \in \mathcal{I}$ of the Radon transform outlined in Theorem 4.4. These plots are used to visualize (in Figures 4.1 and 4.2) the difference between some of the benchmark data and the learned singular functions. We emphasize that in the learned approaches we reconstruct vectors $\mathbf{u}_{k,l} \in \mathbb{R}^{\underline{m}}$ and $\mathbf{v}_{k,l} \in \mathbb{R}^{\overline{m}}$, with $\underline{m} = \overline{m} = 2500$.

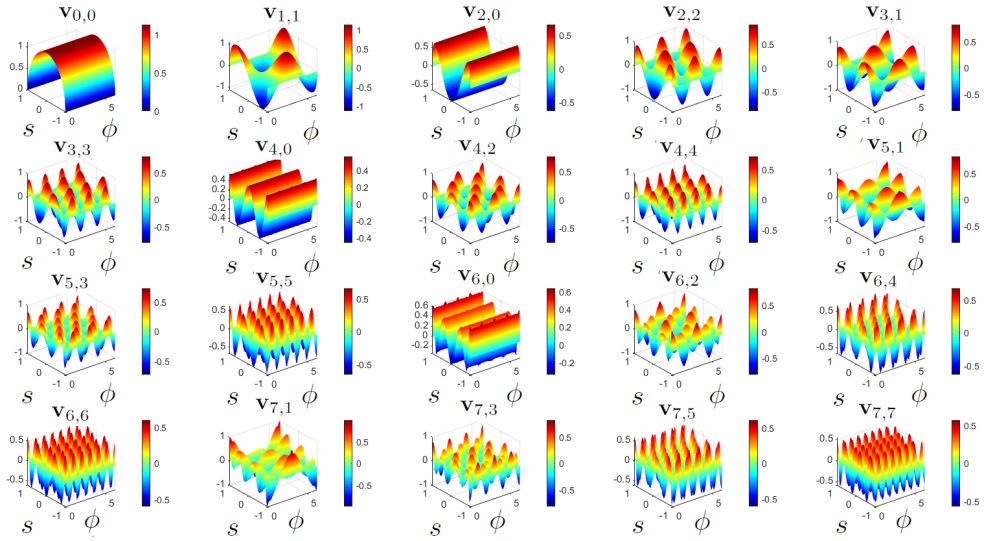


FIG. 4.1. *This plot visualizes the first 20 normalized singular functions* $\mathbf{v}_{k,l}$*, defined in* (4.1)*, for* $k = 0, 1, \ldots, 7$*,* $l = 0, 1, \ldots, k$*, and* $l + k$ *even in the discretized setting.*

We note that the first 49 non-zero eigenvalues of RR^* belong to the following 12 distinct singular values:

$$\Gamma = \{\gamma_{0,0}, \gamma_{1,1}, \gamma_{2,0}, \gamma_{3,1}, \gamma_{4,0}, \gamma_{5,1}, \gamma_{6,0}, \gamma_{7,1}, \gamma_{8,0}, \gamma_{9,1}, \gamma_{10,0}, \gamma_{11,1}, \gamma_{12,0}\}.$$

EXAMPLE 4.6 (Learning spectral functions from analytical spectral functions). We start with a basic test by learning the spectral functions of the Radon transform from the column vectors $\overline{\mathbf{y}}_{kl}$, which are discretizations of the spectral functions $v_{k,l}$ of the Radon transform as written down in (4.1) (some of them are shown in Figure 4.1). To be specific, we want to verify Theorem 2.6 numerically. For this purpose, we construct the matrix $\overline{\mathbf{Y}}$ consisting of the first 49 singular functions (column vectors) $\overline{\mathbf{y}}_{kl}$ and the corresponding matrices $\mathbf{A} = \overline{\mathbf{Y}}\,\overline{\mathbf{Y}}^T \in \mathbb{R}^{\overline{m}\times\overline{m}}$ and $\overline{\mathbf{Y}} \in \mathbb{R}^{\overline{m}\times 49}$, with $\overline{m} = 2500$.

In Figure 4.3, we show the first 20 singular values of $\mathbf{A}$, graphically representing the eigenvectors of the matrix $\overline{\mathbf{W}} = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{49})$ as defined in (2.10) in the proof of Theorem 2.6.

EXAMPLE 4.7. Comparing the results from analytical and learned singular value decomposition reveals what has been stated in Theorem 2.6 that, in the case of eigenvalues of RR^*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

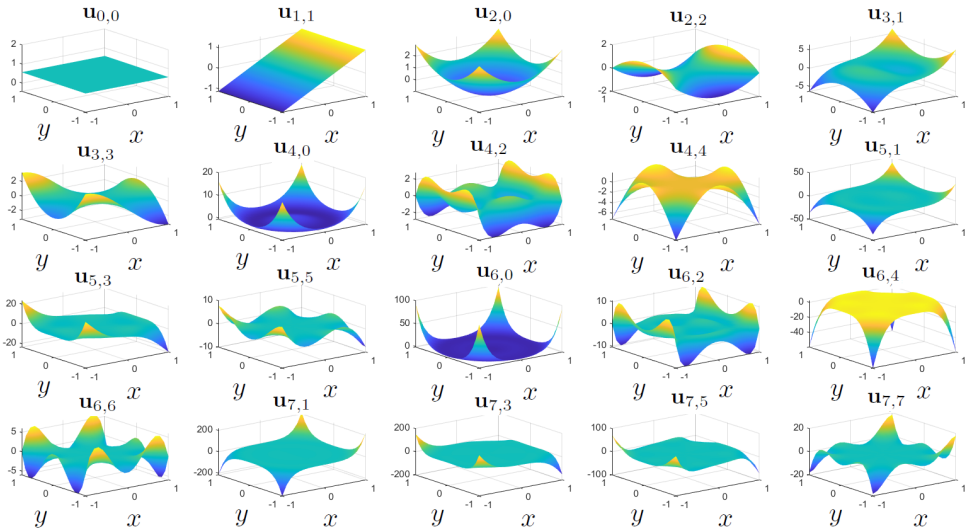


FIG. 4.2. *This plot visualizes the first 20 singular functions* $\mathbf{u}_{k,l}$*, defined in* (4.1)*, for* $k = 0, 1, \ldots, 7$*,* $l = 0, 1, \ldots, k$*, and* $l + k$ *even.*

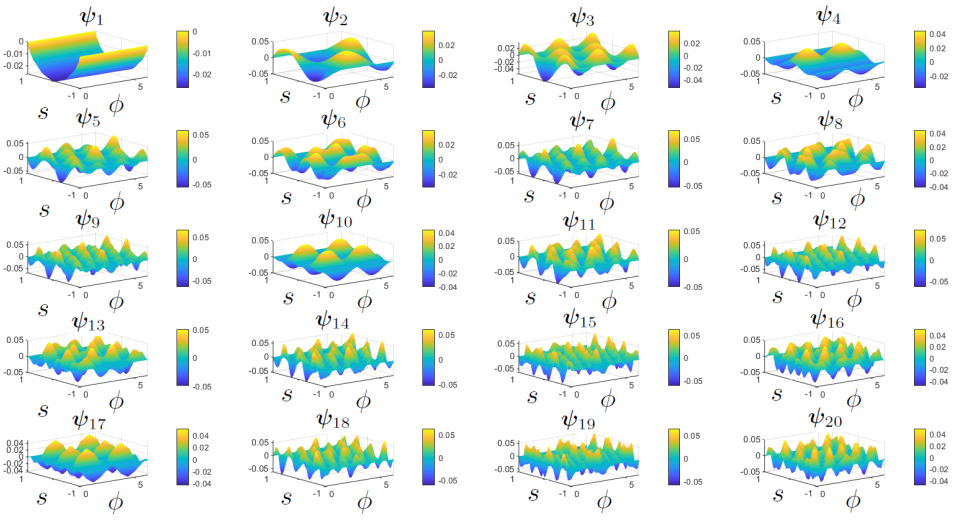


FIG. 4.3. *This plot shows the first 20 functions* $\boldsymbol{\psi}_h$ *obtained from the singular value decomposition of the matrix* $\mathbf{A}$*, defined in* (2.8) *with column vectors* $\overline{\mathbf{y}}_{kl} := \mathbf{v}_{k,l}$*. These results should be compared with the singular functions of the Radon transform* v_{kl} *as plotted in Figure* 4.1*. It is obvious that, if a singular value has multiplicity* 1*, then* $\mathbf{v}_{0,0}$ *is also an eigenfunction, so* $\boldsymbol{\psi}_1$ *perfectly represents the claim. For higher indices, linear combinations are reconstructed approximately, which are elements of the set* E^k*.*

with multiplicity higher than one, only the corresponding eigenspace can be reconstructed. In other words, every $\boldsymbol{\psi}_h$ should be able to be written as a linear combination of the corresponding functions $v_{k,l}$ with $(k, l) \in \mathcal{I}$ with some index l, which needs to be identified in addition. Recall again that $v_{k,l}$ are functions and $\boldsymbol{\psi}_h$ are vectors, which have to be interpreted as discretizations.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Using the eigenspaces E^k, for $k = 1, \ldots, 13$, of the Radon operator R according to the spectral value γ_k, as defined in Theorem 2.6, we identify the sets to which the numerically calculated functions ψ_h, for $h = 1, \ldots, 49$, most likely belong. This is a two-step process as discussed below:

1. Identifying the set E^k, to which ψ_h most likely belongs: We obtained the classification shown in Table 4.1 using our MATLAB code.

TABLE 4.1
Set E^k to which ψ_h most likely belongs.

Set	Singular values
E^1	ψ_1
E^2	ψ_2
E^3	ψ_3, ψ_4
E^4	ψ_5, ψ_6
E^5	ψ_7
E^6	ψ_9
E^7	ψ_8, ψ_{10}
E^8	ψ_{11}, ψ_{13}
E^9	$\psi_{12}, \psi_{14}, \psi_{16}, \psi_{17}$
E^{10}	$\psi_{15}, \psi_{18}, \psi_{20}, \psi_{22}, \psi_{23}, \psi_{45}, \psi_{46}$
E^{11}	$\psi_{19}, \psi_{21}, \psi_{24}, \psi_{25}, \psi_{26}, \psi_{27}, \psi_{44}, \psi_{47}, \psi_{48}, \psi_{49}$
E^{12}	$\psi_{28}, \psi_{30}, \psi_{31}, \psi_{33}, \psi_{34}, \psi_{39}, \psi_{41}, \psi_{43}$
E^{13}	$\psi_{29}, \psi_{32}, \psi_{35}, \psi_{36}, \psi_{37}, \psi_{38}, \psi_{40}, \psi_{42}$

2. Finding an appropriate linear combination of basis elements of E^k, which optimally approximates ψ_h, for $h = 1, \ldots, 49$: For example, applying MATLAB's linear regression software, we obtain the following best approximations for the functions ψ_h, $h = 1, 2, 3, 4, 38, 49$, shown in Figure 4.4. Each plot displays: (1) the original function ψ_h to be approximated, (2) the approximation using the best set of functions E^k (see Table 4.1), and (3) the approximation error. The coefficients of the best approximating linear combinations of the functions ψ_h, where $h = 1, 2, 3, 4, 38, 49$, are shown in Table 4.2.

TABLE 4.2
Functions ψ_h, for $h = 1, 2, 3, 4, 38, 49$, as linear combinations of the eigenfunctions $\mathbf{v}_{k,l}$, in the discretized setting.

Set	Linear combination
E^1	$\psi_1 = -0.02188\mathbf{v}_{0,0}$
E^2	$\psi_2 = 0.03360\mathbf{v}_{1,1}$
E^3	$\psi_3 = 0.01243\mathbf{v}_{2,0} + 0.03831\mathbf{v}_{2,2}$, $\psi_4 = -0.03693\mathbf{v}_{2,0} + 0.01436\mathbf{v}_{2,2}$
E^{11}	$\psi_{49} = 0.02541\mathbf{v}_{10,0} - 0.01122\mathbf{v}_{10,2} - 0.00060\mathbf{v}_{10,4} - 0.00072\mathbf{v}_{10,6} - 0.00061\mathbf{v}_{10,8} + 0.00003\mathbf{v}_{10,10}$
E^{13}	$\psi_{38} = 0.04555\mathbf{v}_{12,0} - 0.02561\mathbf{v}_{12,2} - 0.00396\mathbf{v}_{12,4} - 0.00004\mathbf{v}_{12,6} + 0.00012\mathbf{v}_{12,8} + 0.00003\mathbf{v}_{12,10} - 0.00153\mathbf{v}_{12,12}$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

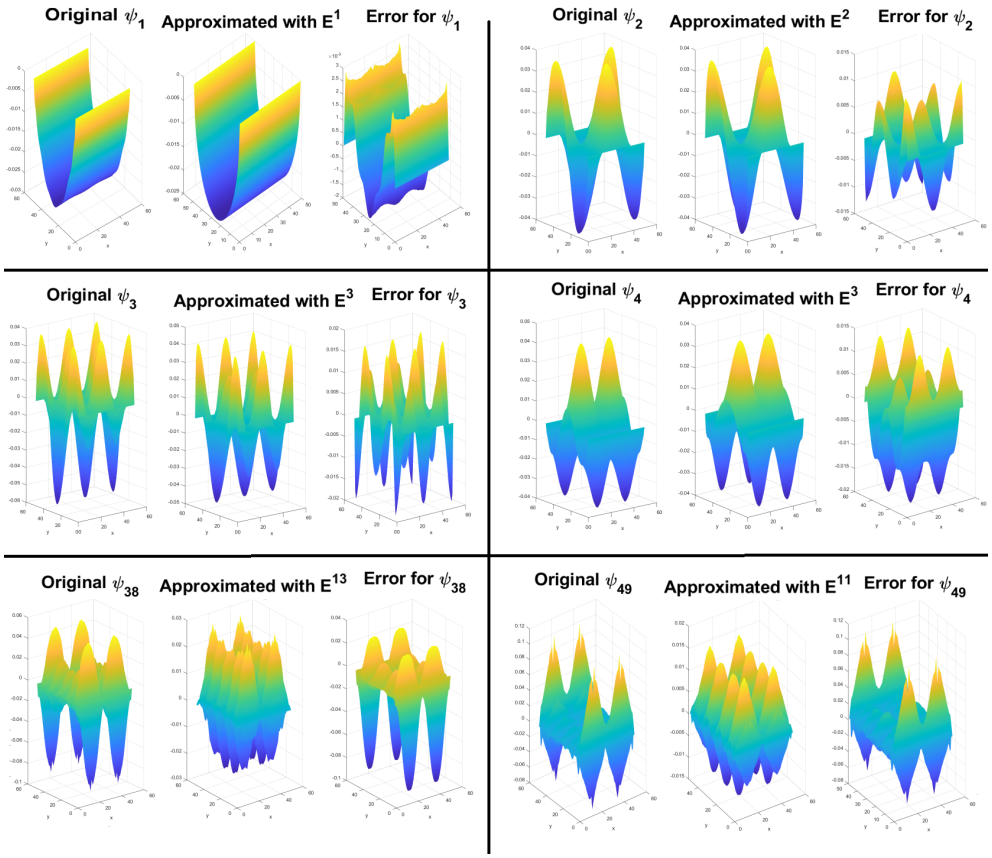


FIG. 4.4. *Results of the linear combinations for ψ_h, with $h = 1, 2, 3, 4, 38, 49$.*

The numerical reconstructions of the singular functions is satisfactory for the first 12 (the first four are shown in the first and second rows of Figure 4.4). For higher indices, the errors are significant, and the error values are of the same order of magnitude as those of the functions ψ_h (see, for example, the plots in the last row of Figure 4.4).

The example has shown that reconstruction of the singular functions and values of operators is possible from training data without knowing the operator explicitly. However, for higher order of indices, the spectral functions become more oscillatory and the numerical treatment becomes increasingly complicated.

EXAMPLE 4.8 (Learning spectral functions from arbitrary test functions). In this example, we consider a dataset, composed of 20 elements, that we have constructed using our own MATLAB function. Specifically, each discrete image, of numerical dimension 50×50, is composed of 10 ellipses that have been generated randomly. In fact, we want to create several images that mimic the structure of the Shepp–Logan phantom. Each ellipse is described by assigning six parameters (intensity, length of the major semi-axis, length of the minor semi-axis, x coordinate of the center, y coordinate of the center, rotation angle) that vary within the ranges reported in Table 4.3.

In Figures 4.5 and 4.6, we present plots of the images constructed with our MATLAB routine and their orthonormalized version. In Figure 4.6, we show the projection of the values onto the xy plane for better visualization of the orthonormalization procedure. The

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 4.3
Parameters and their corresponding ranges for the ellipses of the dataset described in Example 4.8.

Parameter of ellipse	Range
Intensity	$[0.01, 2]$
Length of major semi-axis	$[0.1, 0.7]$
Length of minor semi-axis	$[0.1, 0.7]$
x coordinate of center	$[-0.6, 0.6]$
y coordinate of center	$[-0.6, 0.6]$
Rotation angle	$[-45°, 134°]$

corresponding data (that is, the sinograms) of the orthonormalized images, see (2.2), $\overline{\mathbf{y}}_k = R\overline{\mathbf{x}}_k$, are plotted in Figure 4.7. We are switching here between the discrete and the continuous setting to simplify the presentation.

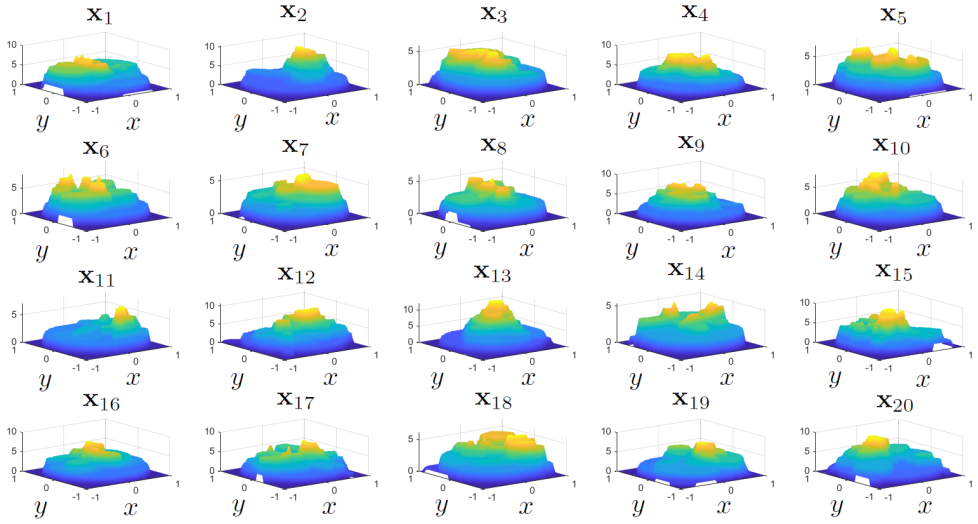


FIG. 4.5. *Training images* $\mathbf{x}_k$, *for* $k = 1, \ldots, 20$.

As in the previous numerical example, see Example 4.6, we utilize the eigenspaces E^k for $k = 1, \ldots, 8$, to identify the sets to which the numerically computed functions $\boldsymbol{\psi}_h$, for $h = 1, \ldots, 20$, most likely belong. We obtained with our MATLAB code the classification given in Table 4.4. As before, we identify suitable linear combinations of elements of E^k in order to rewrite the term $\boldsymbol{\psi}_h$. We present in Figure 4.8 the first four results of the linear combinations, allowing for a direct comparison with the outcomes previously obtained in Example 4.6. The reader can observe, as expected, that the approximation results, expressed as linear combinations, are less precise than those found in Example 4.6.

4.2.2. Decoding. Using the results from the previous section, we implement the reconstruction formula derived in (3.2).

Test 1. In the first example, the ground truth is given by

$$\mathbf{x}_{true} = 0.1134\mathbf{u}_{5,1} + 0.3984\mathbf{u}_{3,1} + 0.8840\mathbf{u}_{0,0} + 0.1803\mathbf{u}_{6,4}. \tag{4.2}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

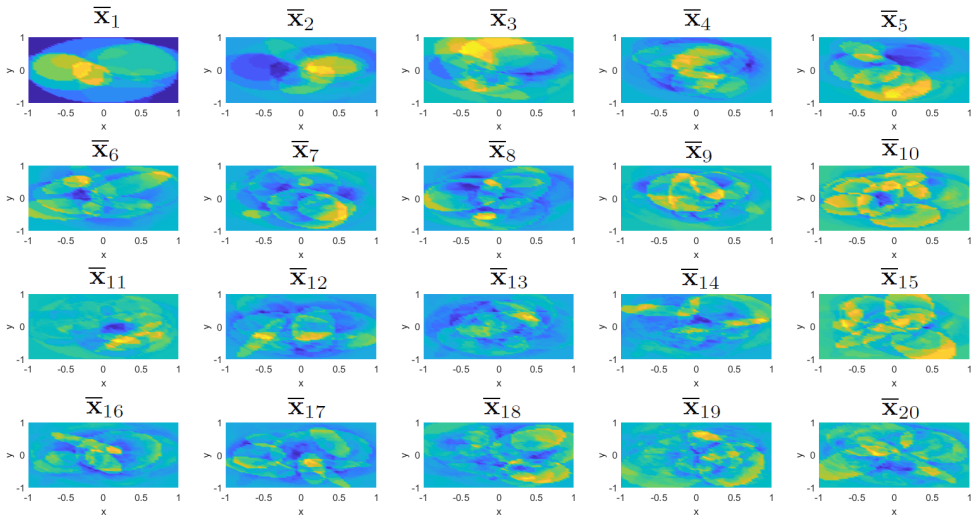


FIG. 4.6. *Orthonormalized images* $\overline{\mathbf{x}}_k$*, for* $k = 1, \ldots, 20$*, projected onto the plane* xy*.*

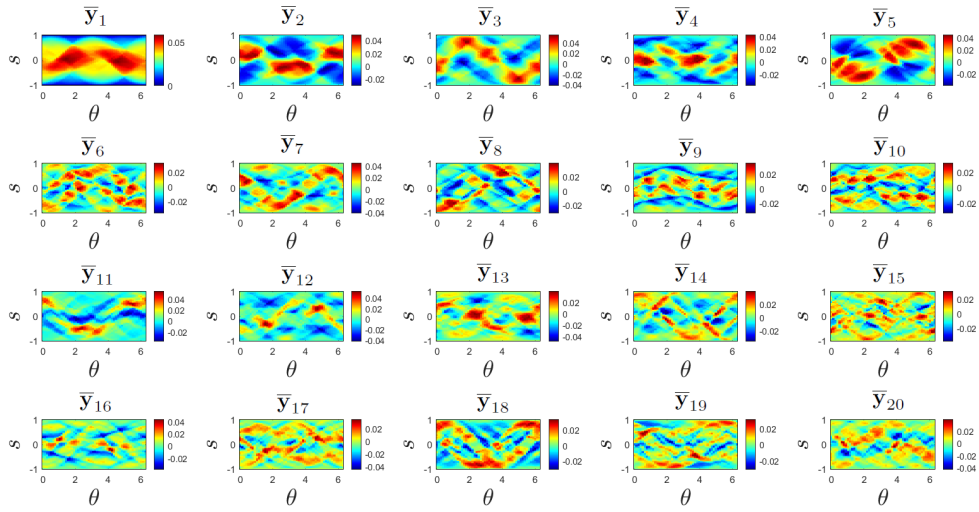


FIG. 4.7. *Orthonormalized training data* $\overline{\mathbf{y}}_k$*, for* $k = 1, \ldots, 20$*.*

In Figure 4.9 we present an example where the ground truth is constructed as a linear combination of singular functions, see (4.2). This setting represents the most favorable case for our approach, as the reconstruction essentially relies on recovering the coefficients of the linear expansion. As expected, the method provides a reasonable approximation of the profile of the ground truth.

Test 2. We introduce a scenario similar to **Test 1**, but now involving a nonlinear combination of functions $u_{k,l}$. Specifically, for each function $u_{k,l}$ in the nonlinear combination, we apply the operation $0.1u_{k,l}^2 + e^{u_{k,l}/\max(u_{k,l})}$, scaled by a specific weight. Results and comparison with ground truth are provided in Figure 4.10. In contrast to Figure 4.9, Figure 4.10 illustrates a more challenging scenario where the ground truth is generated

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 4.4
Set E^k to which ψ_h most likely belongs in the case of Example 4.8.

Set	Singular values
E^1	ψ_1
E^2	ψ_2, ψ_3
E^3	ψ_4, ψ_5, ψ_6
E^4	ψ_9
E^5	ψ_{10}
E^6	$\psi_7, \psi_8, \psi_{18}, \psi_{19}$
E^7	$\psi_{11}, \psi_{12}, \psi_{13}, \psi_{14}, \psi_{17}, \psi_{20}$
E^8	ψ_{15}, ψ_{16}

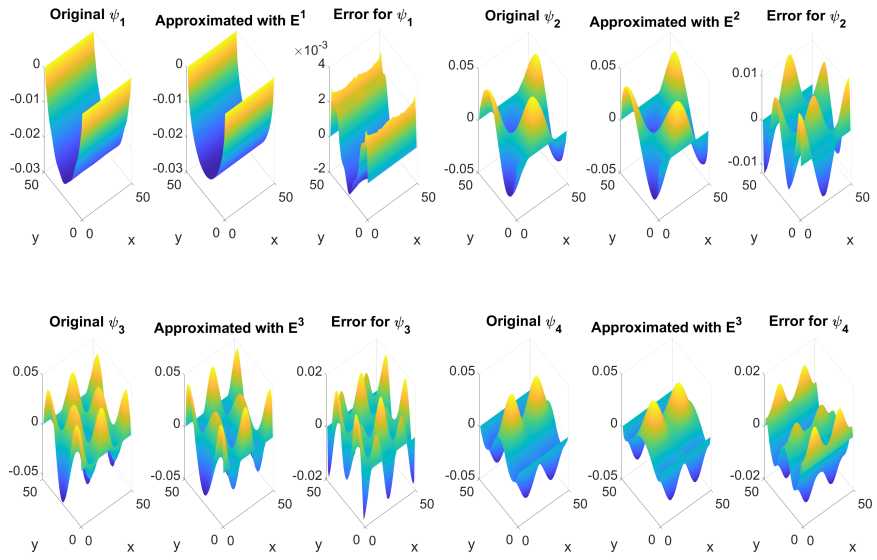
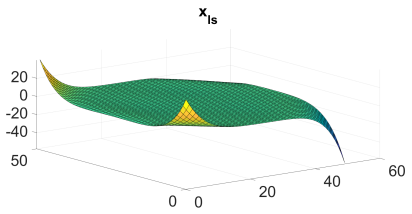


FIG. 4.8. *Results of the linear combinations for ψ_h, with $h = 1, 2, 3, 4$.*

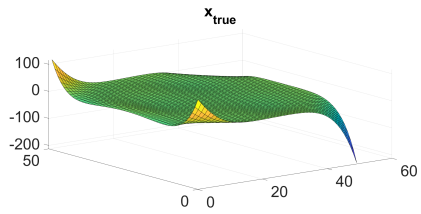
from a nonlinear combination of singular functions. This case goes beyond the exact spectral representation of the operator and is therefore harder to approximate with our current framework. The reconstructions are not only visibly less accurate but, in many cases, also fail to faithfully reproduce the ground truth, which demonstrates the limitations of the method in nonlinear settings. For the nonlinear case, different techniques are required in order to obtain reasonable results.

5. Conclusion. In this paper we have shown that a double orthonormalization strategy, consisting first of orthonormalization of training images and second of a principal component analysis of the training data, theoretically provides us with the singular value decomposition of a linear operator (without making use of an explicit physical model). Two further interesting aspects are shown: First, orthonormalization, like Gram–Schmidt, can be expressed as a deep neural network, and this opens up exploiting synergies with decoder and encoder strategies. Secondly, not only can the singular value decomposition be implemented purely data-driven, but also the decoding, meaning that for arbitrary data the minimum-norm solution can be computed data-driven.

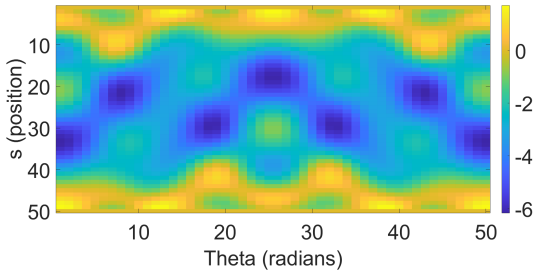
ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) Decoding

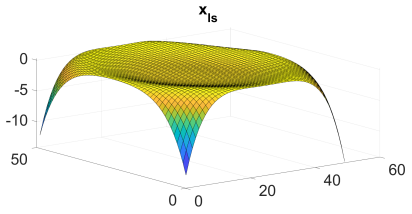


(b) Ground truth

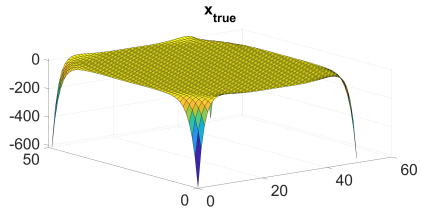


(c) $\mathbf{y}_{true}$

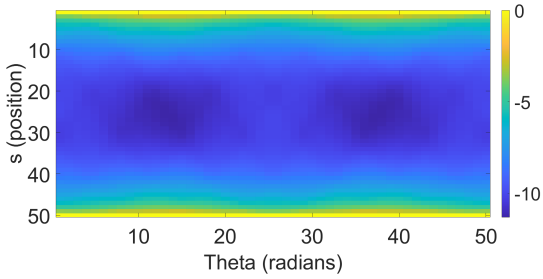
FIG. 4.9. *Decoding, ground truth as represented in* (4.2)*, and* $\mathbf{y}_{true}$.



(a) Decoding



(b) Ground truth



(c) $\mathbf{y}_{true}$

FIG. 4.10. *Decoding, ground truth as represented in* (4.2)*, and* $\mathbf{y}_{true}$ *in the nonlinear combination setting.*

Acknowledgements. This research was funded in whole, or in part, by the Austrian Science Fund (FWF) 10.55776/P34981 – New Inverse Problems of Super-Resolved Microscopy (NIPSUM) and SFB 10.55776/F68 “Tomography Across the Scales”, project F6807-N36

ETNA
Kent State University and
Johann Radon Institute (RICAM)

(Tomography with Uncertainties). For open access purposes, the authors have applied a CC BY public copyright license to any author-accepted manuscript version arising from this submission. Financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged. A.A. is a member of the group GNAMPA (Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni) of INdAM (Istituto Nazionale di Alta Matematica). The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC).

The authors wish to thank Martin Rumpf (University of Bonn) for stimulating discussions and feedback.

REFERENCES

[1] S. ARRIDGE, P. MAASS, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *Solving inverse problems using data-driven models*, Acta Numer., 28 (2019), pp. 1–174.

[2] A. ASPRI, *Learning linear operators solely through training pairs by applying encoding and decoding procedures, without directly utilizing the operators themselves*, Source Code, 2024. https://gitlab.com/csc1/learning-op (commit 66893bab) https://phaidra.univie.ac.at/o:2087334

[3] A. ASPRI, L. FRISCHAUF, Y. KOROLEV, AND O. SCHERZER, *Data driven reconstruction using frames and Riesz bases*, in Deterministic and Stochastic Optimal Control and Inverse Problems, B. Jadamba, A. A. Khan, S. Migórski, and M. Sama, eds., CRC Press, Boca Raton, 2022, pp. 303–318.

[4] A. ASPRI, Y. KOROLEV, AND O. SCHERZER, *Data driven regularization by projection*, Inverse Problems, 36 (2020), Art. No. 125009, 35 pages.

[5] P. BALDI AND K. HORNIK, *Neural networks and principal component analysis: learning from examples without local minima*, Neural Networks, 2 (1989), pp. 53–58.

[6] K. BHATTACHARYA, B. HOSSEINI, N. B. KOVACHKI, AND A. M. STUART, *Model reduction and neural networks for parametric PDEs*, SMAI J. Comput. Math., 7 (2021), pp. 121–157.

[7] C. M. BISHOP, *Neural Networks for Pattern Recognition*, Clarendon Press, New York, 1995.

[8] H. BOURLARD AND Y. KAMP, *Auto-association by multilayer perceptrons and singular value decomposition*, Biol. Cybernet., 59 (1988), pp. 291–294.

[9] J. BRAUNSMANN, M. RAJKOVIC, M. RUMPF, AND B. WIRTH, *Learning low bending and low distortion manifold embeddings*, in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference Proceedings, Los Alamitos, 2021, pp. 4411–4419.

[10] J. CALDER AND P. J. OLVER, *Linear Algebra, Data Science, and Machine Learning*, Springer, Cham, 2025.

[11] E. CARSON, K. LUND, M. ROZLOŽNÍK, AND S. THOMAS, *Block Gram-Schmidt algorithms and their stability properties*, Linear Algebra Appl., 638 (2022), pp. 150–195.

[12] M. E. DAVISON, *A singular value decomposition for the Radon transform in n-dimensional Euclidean space*, Numer. Funct. Anal. Optim., 3 (1981), pp. 321–340.

[13] M. A. G. DUFF, N. D. F. CAMPBELL, AND M. J. EHRHARDT, *Regularising inverse problems with generative machine learning models*, J. Math. Imaging Vision, 66 (2024), pp. 37–56.

[14] H. W. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, Dortrecht, 1996.

[15] A. FAWZI, M. BALOG, A. HUANG, T. HUBERT, B. ROMERA-PAREDES, M. BAREKATAIN, A. NOVIKOV, F. J. R. RUIZ, J. SCHRITTWIESER, G. SWIRSZCZ, D. SILVER, D. HASSABIS, AND P. KOHLI, *Discovering faster matrix multiplication algorithms with reinforcement learning*, Nature, 610 (2022), pp. 47–53.

[16] J. G. F. FRANCIS, *The QR transformation a unitary analogue to the LR transformation–part 1*, The Computer Journal, 4 (1961), pp. 265–271.

[17] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[18] M. HALTMEIER AND L. NGUYEN, *Regularization of inverse problems by neural networks*, in Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging, K. Chen, C.-B. Schönlieb, X.-C. Tai, and L. Younes, eds., Springer, Cham, 2022, pp. 1065–1093.

[19] M. HANKE AND O. SCHERZER, *Addendum on data driven regularization by projection*, Preprint on arXiv, 2025. https://arxiv.org/abs/2508.07709

[20] E. HART, J. CHUNG, AND M. CHUNG, *A paired autoencoder framework for inverse problems via bayes risk minimization*, Preprint on arXiv, 2025. https://arxiv.org/abs/2501.14636

[21] N. KOVACHKI, S. LANTHALER, AND S. MISHRA, *On universal approximation and error bounds for Fourier neural operators*, J. Mach. Learn. Res., 22 (2021), Art. No. 290, 76 pages.

[22] N. KOVACHKI, Z. LI, B. LIU, K. AZIZZADENESHELI, K. BHATTACHARYA, A. STUART, AND A. ANAND-

ETNA
Kent State University and
Johann Radon Institute (RICAM)

KUMAR, *Neural operator: learning maps between function spaces with applications to PDEs*, J. Mach. Learn. Res., 24 (2023), Art. No. 89, 97 pages.

[23] S. LANTHALER, Z. LI, AND A. M. STUART, *Nonlocality and nonlinearity implies universality in operator learning*, Constr. Approx., 62 (2025), pp. 261–303.

[24] S. LANTHALER, S. MISHRA, AND G. E. KARNIADAKIS, *Error estimates for DeepONets: a deep learning framework in infinite dimensions*, Trans. Math. Appl., 6 (2022), Art. No. tnac001, 141 pages.

[25] S. LANTHALER AND A. M. STUART, *The parametric complexity of operator learning*, IMA J. Numer. Anal., (2025), Art. No. draf028, 66 pages.

[26] H. LIU, H. YANG, M. CHEN, T. ZHAO, AND W. LIAO, *Deep nonparametric estimation of operators between infinite dimensional spaces*, J. Mach. Learn. Res., 25 (2024), Art. No. 24, 67 pages.

[27] P. MAASS, *The X-ray transform: singular value decomposition and resolution*, Inverse Problems, 3 (1987), pp. 729–741.

[28] M. NASHED, ed., *Generalized Inverses and Applications*, Academic Press, New York, 1976.

[29] F. NATTERER, *The Mathematics of Computerized Tomography*, SIAM, Philadelphia, 2001.

[30] F. NATTERER AND F. WÜBBELING, *Mathematical Methods in Image Reconstruction*, SIAM, Philadelphia, 2001.

[31] N. H. NELSEN AND A. M. STUART, *Operator learning using random features: a tool for scientific computing*, SIAM Rev., 66 (2024), pp. 535–571.

[32] A. PAPOULIS, *The Fourier Integral and its Applications*, McGraw-Hill, New York, 1962.

[33] T. I. SEIDMAN, *Nonconvergence results for the application of least-squares estimation to ill-posed problems*, J. Optim. Theory Appl., 30 (1980), pp. 535–547.

[34] O. STEINBACH, *Lösungsverfahren für lineare Gleichungssysteme*, Vieweg+Teubner, Wiesbaden, 2005.

[35] Y. SU, X. GAO, X. LI, AND D. TAO, *Multivariate multilinear regression*, IEEE Trans. Syst. Man. Cybern. B Cybern., 42 (2012), pp. 1560–1573.

[36] D. N. TANYU, I. MICHEL, A. RADEMACHER, J. KUHNERT, AND P. MAASS, *Parameter identification by deep learning of a material model for granular media*, GEM Int. J. Geomath., 15 (2024), Art. No. 13, 24 pages.

[37] J. E. VAN ENGELEN AND H. H. HOOS, *A survey on semi-supervised learning*, Mach. Learn., 109 (2020), pp. 373–440.

[38] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, New York, 1988.

[39] F. WILLERS, *Methoden der praktischen Analysis*, 4th ed., de Gruyter, Berlin, 1971.