Learned SVD: solving inverse problems via hybrid autoencoding
Yoeri E. Boink†* and Christoph Brune†

Abstract. Our world is full of physics-driven data where effective mappings between data manifolds are desired. There is an increasing demand for understanding combined model-driven and data-driven learning methods. We propose a nonlinear, learned singular value decomposition (L-SVD), which combines autoencoders that simultaneously learn and connect latent codes for desired signals and given measurements. Classical solution methods for inverse problems are based on regularisation techniques via SVD and variational methods. An open topic in deep learning for inverse problems is how to achieve model reduction via data dimensionality reduction to obtain a regularised inversion. We investigate this topic and provide a promising direction for solving inverse problems in cases where the underlying physics are not fully understood or have very complex behaviour. We show that the building blocks of learned inversion maps can be obtained automatically, with improved performance upon classical methods and better interpretability than black-box methods.

Key words. inverse problems, neural networks, dimensionality reduction, autoencoders, SVD, regularisation.

1. Introduction. We are living in a world full of physics-driven data with an increasing demand for combining model-driven and data-driven approaches in areas of science, industry and society. In many cases it is essential to reliably recover hidden multi-dimensional model parameters (signals) $x \in \mathcal{X}$ from noisy indirect observations (measurements) $y^\delta \in \mathcal{Y}$, e.g. in imaging or sensing technology in medicine, engineering, astronomy or geophysics. These inverse problems, $y^\delta = \mathbf{A}(x) + \eta^\delta$, are often ill-posed, suffering from non-uniqueness and instability in direct inversion. Classical model-driven research on inverse problems has focused on variational regularisation methods to guarantee existence and stable approximation of solutions under uncertainty like noise $\eta^\delta$ in the measurements [13, 6]. For linear inverse problems the singular value decomposition (SVD) [17] is a classical tool to directly construct a regularised inverse, e.g. in the sense of Tikhonov regularisation [13]. Recent research in inverse problems has focused on combining deep learning with model-driven approaches based on knowledge of the underlying physics [3]. Precise knowledge is often not available; for now we rely mainly on empirical evidence that such approaches can still be applied when one makes use of inexact operators that approximate the exact physical process [19]. The main limitation of such methods are that they require an iterative application of expensive, possibly nonlinear operators. Moreover, they are hard to interpret because the connection between data structure and underlying operator structure is not explainable.

Figure 1.1: L-SVD learns the inversion mapping via a hybrid nonlinear data manifold learning.

*Multi-Modality Medical Imaging and Biomedical Photonic Imaging, Technical Medical Centre, University of Twente, NL (y.e.boink@utwente.nl)
†Department of Applied Mathematics, University of Twente, NL (y.e.boink@utwente.nl, c.brune@utwente.nl)
We propose the ‘learned singular value decomposition’ (L-SVD): a direct method that provides the inversion procedure with an explainable connection between measurements and signals that does not rely on an iterative application of expensive operators. It makes use of two connected autoencoders: the first one encodes measurement \( y^\delta \) to latent code \( z_y \), while the second one encodes signal \( x \) to latent code \( z_x \) in a nonlinear way; both latent codes are connected with a linear ‘scaling’ layer. The training of all parameters is done simultaneously, which enforces the latent codes to preserve as much information on the measurements and signals as possible, while making sure that the codes have very similar structure. After training, a reconstruction is obtained by consecutive application of encoding, linear scaling and decoding (see Figure 1.1).

The advantage of such a method is that the nonlinear inversion dynamics are captured in the autoencoders, while the scaling layer is linear, making it easy to understand and analyse. Assuming that the autoencoders can be trained with high accuracy, finding the connection between both codes is a much lower dimensional problem than finding a nonlinear map between the measurements and signals directly.

### 1.1. Contribution

This paper proposes the learned singular value decomposition (L-SVD), a general data-driven method that nonlinearly compresses data in two vector spaces and connects them in an easy-to-understand way. It can be seen as an extension of existing methods from the following perspectives:

1. **Data-driven solution of inverse problems**: L-SVD is a nonlinear generalisation of Tikhonov regularisation in Bayesian inverse problems [14, 22, 39] and piecewise linear estimates [41]. Those methods can be seen as linear data-driven variants of several versions of classical SVD approaches.

2. **Nonlinear encoding**: Autoencoders show that nonlinear compression provides better quality compressions than linear compression [20]. L-SVD shows that this is also the case when compressions are used to solve an inverse problem.

3. **Hybrid autoencoding**: It is understood that an autoencoder can act as a regulariser when attached to a supervised neural network that is trained for a supervised task, e.g. classification [43, 25]. L-SVD makes use of two autoencoders for the task of solving an inverse problem, which provides improved generalisation performance.

### 1.2. Overview of the paper

Throughout the paper, we make use of the previous three perspectives to show the advantage of using the L-SVD method for addressing the limitations mentioned above. In section 2, a brief overview of the classical SVD and inversion methods is given, which serves as motivation for the L-SVD method. Next, a precise definition of L-SVD is provided in section 3, accompanied with various architecture choices that exploit its potential. Then we analyse the method by showing its connection with Bayesian inverse problems and providing a stability and error estimate in section 4. After that, in Section 5, the connection of our work with several fields of research are discussed. In section 6, we explain experiments that show the transition from non-learned to learned, linear to nonlinear, and single compression to hybrid compression. Results are provided in section 7, where we visualise these transitions by looking at the latent space, the decoded representations of the latent space and the dependency of L-SVD on noise. In section 8, we conclude with some remarks and outlook for future work.

### 2. Motivation: SVD and inversion methods

The motivation of our L-SVD method can be found in the application of classical SVD and its variants in inversion methods. In our work we consider the finite dimensional version of the equation introduced in section 1. That is, we make use of the ‘first discretise, then optimise’ approach. We define the inverse
problem as
\begin{equation}
    y^\delta = A(x) + \eta^\delta,
\end{equation}
where we wish to reconstruct the signals \( x \in X \subseteq \mathbb{R}^m \) from measurements \( y^\delta \in Y \subseteq \mathbb{R}^n \) corrupted by additive noise \( \eta^\delta \sim \mathcal{N}(0, \delta \text{Id}) \). Here \( X \) and \( Y \) are Banach spaces. The mapping \( A : X \rightarrow Y \) is in general a nonlinear one. For this section however, we assume a linear operator that we call \( A \). Any \( A \) can be written in its singular value decomposition: 
\[ A = USV^* \]
where \( U \in \mathbb{R}^{n \times n} \) and \( V \in \mathbb{R}^{m \times m} \) are unitary matrices and \( S \in \mathbb{R}^{n \times m} \) is a diagonal matrix with nonnegative real numbers \( s_i \) (singular values) on the diagonal. We now summarise well-known inversion methods that can be written as the application of an SVD [17].

2.1. Maximum likelihood estimator (MLE). The MLE is defined via the maximisation over \( x \) given measurements \( y^\delta \) [39]. Its solution \( x_{\text{MLE}} \) is obtained by applying the Moore-Penrose inverse \( (A^*A)^{-1}A^* \) to the measurements \( y^\delta \).
\begin{equation}
    x_{\text{MLE}} := \arg\max_x p(x|y^\delta) = \arg\min_x \|Ax - y^\delta\|_2^2 = (A^*A)^{-1}A^*y^\delta.
\end{equation}

2.2. Tikhonov regularisation. Tikhonov regularisation is a method which puts a uniform variance prior on the desired solution \( x \). It solves an \( \alpha \)-weighted minimisation problem that can be solved directly via its regularised Moore-Penrose inverse:
\begin{equation}
    x_\alpha := \arg\min_x \|Ax - y^\delta\|_2^2 + \alpha\|x\|_2^2,
\end{equation}
\[ x_\alpha = (A^*A + \alpha \text{Id})^{-1}A^*y^\delta = V \underbrace{(S^2 + \alpha \text{Id})^{-1}}_{S_\alpha^{-1}} S^*U^*y^\delta. \]
The diagonal elements of \( S_\alpha^{-1} \) are defined as \( s_i/(s_i^2 + \alpha) \), which means that for smaller scales \( s_i \), the new inverse scale goes to zero as \( \alpha \) gets larger. The optimal \( \alpha \) depends on the type of noise; usually \( \alpha \) increases with noise level \( \delta \).

2.3. Truncated SVD. The best Frobenius approximation of \( A \) with rank \( r \) is given [12] by the truncated SVD (T-SVD):
\begin{equation}
    A_r := \arg\min_{\tilde{A}} \|A - \tilde{A}\|_{\text{Fro}} \quad \text{s.t.} \quad \text{rank}(\tilde{A}) = r.
\end{equation}
Here we made use of the ‘thin’ representation, where \( U_r \in \mathbb{R}^{n \times r} \) and \( V_r \in \mathbb{R}^{m \times r} \) consist of the top \( r \) rows of \( U \) and \( V \) respectively. \( S_r \in \mathbb{R}^{r \times r} \) is a diagonal square matrix that consists of the largest \( r \) singular values of \( A \). With the thin representation, we lose one desirable property, namely that of unitary matrices: while \( U_r^*U_r = \text{Id}_r = V_r^*V_r \) still holds, generally \( U_rU_r^* \neq \text{Id} \neq V_rV_r^* \).

T-SVD can be applied in an inversion method instead of the standard SVD for noisy measurements \( y^\delta \): when \( s_i \) becomes small for \( i \) large, noise is amplified by \( 1/s_i \) in (2.2). This problem is mitigated by solving \( x_{\text{trunc}} := V_rS_r^{-1}U_r^*y^\delta \) instead. It has the additional benefit that the thin decomposition is smaller than the full SVD, requiring less memory and computation time.
3. The Learned Singular Value Decomposition. In this section, we provide the general L-SVD method for solving inverse problems. It aims to solve the inverse problem as defined in (2.1), where the forward mapping $\mathcal{A}$ may be nonlinear. L-SVD can be seen as a nonlinear learned variant of the inversion methods in section 2, where $U^*$ is replaced by a nonlinear encoder and $V$ by a nonlinear decoder.

3.1. Model statement. The L-SVD model (Figure 3.1) is a trained neural network that consists of a measurement autoencoder (green), a signal autoencoder (blue) and a reconstruction component (red). Reconstruction $\hat{x}$ from measurement $y^\delta$ is obtained via the latent representations $z_x \in Z^x \subseteq \mathbb{R}^k$ and $z_y \in Z^y \subseteq \mathbb{R}^k$, which are part of the autoencoders. The latent space $\mathbb{R}^k$ is a low-dimensional space, i.e. $k \leq \min\{m, n\}$.

![Figure 3.1: Schematic of the L-SVD method.](image)

**Green**: autoencoder for measurement $y^\delta$. **Blue**: autoencoder for signal $x$. **Red**: reconstruction procedure. The standard network does not use the gray connections for training. Note that $\varphi_{\text{dec}}^x$ is used multiple times, but has shared weights.

A more formal definition is given as:

**Definition 3.1.** We define the nonlinear functions

$$\varphi_{\text{enc}}^y : Y \mapsto Z^y, \quad \varphi_{\text{dec}}^y : Z^y \mapsto Y, \quad \varphi_{\text{enc}}^x : X \mapsto Z^x, \quad \varphi_{\text{dec}}^x : Z^x \mapsto X$$

and we define the square matrix $\Sigma \in \mathbb{R}^{k \times k}$. Moreover we define the variables

$$z_y := \varphi_{\text{enc}}^y(y^\delta), \quad z_x^{\text{AE}} := \varphi_{\text{enc}}^x(x), \quad z_x^\Sigma := \Sigma z_y,$$

$$\hat{y}^{\text{AE}} := \varphi_{\text{dec}}^y(z_y), \quad \hat{x}^{\text{AE}} := \varphi_{\text{dec}}^x(z_x^{\text{AE}}), \quad \hat{x}^\Sigma := \varphi_{\text{dec}}^x(z_x^\Sigma).$$

The L-SVD model is obtained by minimising a neural network loss function that consists of three parts:

$$\min_{\text{par}_{\text{NN}}} \left\{ \#\text{train} \sum_{i=1}^\#\text{train} \left( D_1(\hat{x}_i^\Sigma, x_i) + \alpha_g D_2(\hat{y}_i^{\text{AE}}, y_i) + \alpha_x D_3(\hat{x}_i^{\text{AE}}, x_i) \right) \right\},$$

where we minimise over all trainable parameters $\text{par}_{\text{NN}}$ and over all samples $(i)$ in the training set. The distance functions $D_j(\cdot, \cdot)$ can be any metric; often used in neural networks are $\ell^2, \ell^1$ and $W^2$ (Wasserstein) metrics. The L-SVD model compresses measurements $y^\delta$ into a representation that contains sufficient information to approximately reconstruct the clean data $y$, while being able to linearly map to a compressed representation that can approximately be ‘decompressed’ to the desired signal $x$. Since the output of the data autoencoder is the clean
data $y$, instead of the corrupted measurements $y^\delta$, it can be seen as a denoising autoencoder (DAE) \cite{11}. This means that noise will not necessarily be represented in the latent variable $z_y$. Note that this model only assumes nonlinear compression and does not say anything about the way it is compressed: one can use any type of neural network that is suitable for the specific application, including fully-connected, sparse and convolutional neural networks. This includes regularised autoencoders, such as sparse \cite{12} or contractive autoencoders \cite{13}.

3.1.1. Opportunities by various model choices. Below some variations on the standard model are discussed, which establish certain capabilities of the L-SVD model:

- **[noise-aware $\Sigma$]** If the autoencoder on the measurement side is chosen to be a regular autoencoder instead of a denoising autoencoder, noise will still be represented in the encoded version $\tilde{z}_y$. This means that the latent dimension should be large enough, since unstructured noise can not be compressed. Moreover, this means that $\Sigma$ should be able to remove (part of) the noise, since the latent variable $z_x$ is noise-free. The linear matrix $\Sigma$ can be changed to a nonlinear function $\Sigma(z_y)$, that treats latent variables with different noise-levels differently.

- **[scaling layer $\Sigma$]** The matrix $\Sigma$ can be chosen to be diagonal instead of full. This means that it serves as a scaling layer between $\tilde{z}_y$ and $z_x$, which is also the case in classical SVD (see section 2).

- **[structure latent space]** No specific structures of the latent spaces $Z^x$ and $Z^y$ are imposed. If control on these spaces is desired, one could sample from a desired set in the latent space $\tilde{z}_x \in E \subset Z^x$ and add one of the following losses to (3.1):

  $\alpha_{z_x} D_4(\tilde{z}_x, z^\Delta_x)$ or $\alpha_{z_x} D_5(\tilde{z}_x, z^\Sigma_x)$.

This means the sampled latent code is decoded to a signal $x$ (gray in Figure 3.1), after which it takes either the blue autoencoder path or the red reconstruction path, without the final decoder step $\varphi_{dec}$. Although not guaranteed, it is likely that due to this additional loss, the encoder $\varphi^x_{enc}$ will map all samples $u(i)$ in the training set to this subset $E$. If this is the case, it means that we have control over the latent space $Z^x$. Moreover it turns out that having a bound on $D_5(\tilde{z}_x, z^\Sigma_x)$ enables us to compute a uniform error bound for the reconstruction procedure (see section 4.3).

4. Analysis. In this section, we show that training the L-SVD model coincides with learning the covariance matrix of a prior in Bayesian inverse problems for a specific linear choice of the encoder and decoder. Moreover, we provide a stability estimate and an error estimate for the L-SVD model.

4.1. Connection with Bayesian inverse problems. Here we provide an explicit connection between a linear L-SVD model with the solution of a Bayesian inverse problem with Gaussian noise, Gaussian prior and known forward operator $A$. For an introduction to statistical and Bayesian inverse problems, we refer the reader to \cite{14, 4, 39}.

4.1.1. Learning the prior covariance matrix.

**Proposition 4.1.** Let $x \in X \subset \mathbb{R}^m$, $\tilde{y} \in Y \subset \mathbb{R}^n$ and $\eta \sim \mathcal{N}(0, B)$, where $X$ and $Y$ are Banach spaces. Consider the inverse problem

$$\tilde{y} = Ax + \eta,$$

where $A \in \mathbb{R}^{n \times m}$ has full row-rank, i.e. $\text{rank}(A) = n \leq m$, with thin SVD decomposition $A = US_nV_n^*$. Moreover, let $\mu_0 \sim \mathcal{N}(0, C_0)$ be a Gaussian prior measure on $x$. 5
We define $\tilde{B} := U^* B U$ and restrict the covariance matrix $C_0$ to be of rank $n$ that can be written as $C_0 = V_n C_{V_n} V_n^*$, where $C_{V_n}$ is positive definite.

Then the maximum a posteriori (MAP) estimate $x_{\text{MAP}} := \arg\max_x p(y|x)$ can be written as an SVD inversion method in the following way:

$$
(4.1) \quad x_{\text{MAP}} = V_n \Sigma U^* \tilde{y}
$$

with $\Sigma = \left[B(C_{V_n} S_n)^{-1} + S_n\right]^{-1}$.

For the proof we refer to appendix A. The connection between (4.1) and L-SVD is clear if we define the linear measurement encoder to be $\varphi_{\text{enc}}^y := U^*$, the linear signal decoder to be $\varphi_{\text{dec}}^z := V_n$ and we assume the noise covariance matrix $B$ to be known. Then it can be seen in (4.1) that learning the matrix $\Sigma$ is equivalent to optimising over the prior covariance matrix $C_0$, defined via $C_{V_n}$.

### 4.1.2. Scale dependency on Gaussian noise level

For many inverse problems one assumes an additive noise term that originates from the Gaussian distribution $\eta^\delta \sim \mathcal{N}(0, \delta \mathbb{I})$, where the noise level $\delta$ is either known or estimated. If the data covariance matrix $B$ is replaced with $\delta \mathbb{I}$, (4.1) is simplified to

$$
(4.2) \quad \Sigma = \left[\delta(C_{V_n} S_n)^{-1} + S_n\right]^{-1}.
$$

This implies a higher regularisation $(C_{V_n} S_n)^{-1}$ with increasing noise level $\delta$. Thus, by learning the scales $\Sigma$, we learn the prior distribution on $x$ which regularises our inverse problem. For a prior distribution $C_{V_n} = \lambda \mathbb{I}$, it is easily shown that we get the formulation for classical Tikhonov-regularisation (2.3) back:

$$
\Sigma = \left[\delta(\lambda \mathbb{I} S_n)^{-1} + S_n\right]^{-1} = \left[(\delta/\lambda + S_n^2 S_n^{-1}) S_n^{-1}\right]^{-1} = S_n^{-\alpha} \quad \text{for} \quad \alpha := \delta/\lambda.
$$

### 4.2. Stability estimate

Here we provide an estimate on the stability of the model; that is, how much the reconstruction output $\hat{x}^y$ will change as an effect of a deviation in measurement input $y^\delta$. For two measurement inputs $y^\delta_{(1)}$ and $y^\delta_{(2)}$, we compute the bound

$$
(4.3) \quad \|\hat{x}^y_{(1)} - \hat{x}^y_{(2)}\|_\ell^2 \leq \|\varphi_{\text{dec}}^y\|_{\text{op}} \|\hat{z}_{x,(1)} - \hat{z}_{x,(2)}\|_\ell^2
$$

$$
\leq |\sigma_{\text{max}}| \|\varphi_{\text{dec}}^y\|_{\text{op}} \|\hat{z}_{y,(1)} - \hat{z}_{y,(2)}\|_\ell^2
$$

$$
\leq \|\varphi_{\text{enc}}^y\|_{\text{op}} |\sigma_{\text{max}}| \|\varphi_{\text{dec}}^y\|_{\text{op}} \|y_{(1)} - y_{(2)}\|_\ell^2.
$$

Here $\|\cdot\|_{\text{op}}$ represents the operator norm. Note that for a nonlinear function

$$
\varphi(x) := \tau(W_1 \tau(W_{L-1} \ldots \tau(W_1 x) \ldots))
$$

consisting of $L$ layers of weight matrices $W_l$ and pointwise nonlinearities $\tau(x)$ s.t. $\|\tau\|_{\text{op}} = C$, we have the bound $\|\varphi\|_{\text{op}} \leq C L \prod_{l=1}^L \|W_l\|_{\ell^2}$. In case the nonlinearity is a ReLU, leaky ReLU or hyperbolic tangent, $C = 1$ and thus the norm only depends on the norms of the weight matrices. This means we have some influence on the error estimate of (4.3) by controlling $\sigma_{\text{max}}$ and the $\ell^2$-norms of the weight matrices in $\varphi_{\text{enc}}^y$ and $\varphi_{\text{dec}}^y$. This can be achieved by regularisation in the neural network objective function.
4.3. Reconstruction error estimate. Finally, we prove that an error estimate on the difference between any reconstructed signal and the true solution exists, provided that its associated measurement maps to a ball in the latent space. Before we prove that such an error estimate exists, we proof a supporting Lemma.

Lemma 4.2. Let $z \in \mathbb{R}^k$, let $F : \mathbb{R}^k \to \mathbb{R}^k$ be a continuous function. Assume $\forall z \in B_1$, $\|z - F(z)\|_2 < \varepsilon$ for some given $0 \leq \varepsilon < 1$. Then $B_{1-\varepsilon} \subset F(B_1)$, where $B_r := \{ z \in \mathbb{R}^k \mid \|z\|_2 \leq r \}$ is the closed ball centered at 0 with radius $r$.

Proof. Let us first define a scaled function $\tilde{F} : \mathbb{R}^k \to \mathbb{R}^k$, with $\tilde{F}(z) := \frac{1}{1-\varepsilon}F(z)$. For this scaled function, $\tilde{F}(B_1) \subset B_1$.

The closed unit ball $B_1$ is a contractible space. By definition of a contractible space [16], $\exists G : B_1 \times [0,1] \to B_1$ continuous such that for all $z$, $G(z,0) = z_0$ and $G(z,1) = z$, where $z_0$ is the contraction point. Since $\tilde{F}(B_1) \subset B_1$ for all $z \in \tilde{F}(B_1)$, it holds that $G(\tilde{F}(z),0) = z_0$ and $G(\tilde{F}(z),1) = \tilde{F}(z)$. Since both $G$ and $F$ are continuous, its composition is continuous, hence $\tilde{F}(B_1)$ is also a contractible space. Since $\tilde{F}$ is just a scaled version of $F$, the same holds for the $F$. This means that the scaled image under the unit ball can not have any ‘holes’ in it.

Left to show is that the boundary of $F(B_1)$ lies outside $B_{1-\varepsilon}$, which implies that the whole ball $B_{1-\varepsilon}$ is in the range of $B_1$. For this we make use of [32][Theorem 4.22]: since $F$ is a continuous mapping of a metric space $(\mathbb{R}^k, \|\cdot\|_2)$ into a metric space $(\mathbb{R}^k, \|\cdot\|_2)$, and the boundary of the unit ball (i.e. $\partial B_1$) is a connected subset of $\mathbb{R}^k$, this implies $F(\partial B_1)$ is connected. Moreover, $\forall z \in \partial B_1$, $F(z) \in B_{1+\varepsilon}\setminus B_{1-\varepsilon}$. This implies that the boundary of the unit ball lies completely outside $B_{1-\varepsilon}$, which completes the proof.

Theorem 4.3. Consider the L-SVD network as defined in section 3.1. Assume that for some $0 < \varepsilon_z < 1$ we have that for all $\tilde{z}_x \in B_1$, $\|\tilde{z}_x - z_x^*\|_2 < \varepsilon_z$. Then there exists an $\varepsilon_x > 0$ s.t. for all $\tilde{z}_x \in B_1$, $\|\varphi_{\text{dec}}(\tilde{z}_x) - \varphi_{\text{dec}}(\Sigma \varphi_{\text{enc}}(A \varphi_{\text{dec}}(\tilde{z}_x)))\|_2 < \varepsilon_x$. Moreover, for all $x \in \mathbb{R}^n$ for which $\varphi_{\text{enc}}(x) \in B_{1-\varepsilon_z}$, we have the error estimate $\|x - \varphi_{\text{enc}}(\Sigma \varphi_{\text{enc}}(A x))\|_2 < \varepsilon_x$.

Proof. The first part of the proof can be obtained by combining the operator norm of the decoding function in $x$ and the given bound:

\begin{equation}
\|\varphi_{\text{dec}}(\tilde{z}_x) - \varphi_{\text{dec}}(\Sigma \varphi_{\text{enc}}(A \varphi_{\text{dec}}(\tilde{z}_x)))\|_2 \\
\leq \|\varphi_{\text{dec}}\|_\text{op} \|\tilde{z}_x - \Sigma \varphi_{\text{enc}}(A \varphi_{\text{dec}}(\tilde{z}_x))\|_2 \\
< \varepsilon_x \|\varphi_{\text{dec}}\|_\text{op} =: \varepsilon_x
\end{equation}

For the second part of the proof, we make use of Lemma 4.2. We define $F(\tilde{z}_x) := \Sigma \varphi_{\text{enc}}(A \varphi_{\text{dec}}(\tilde{z}_x))$, which is a continuous function from $\mathbb{R}^k$ to $\mathbb{R}^k$. Since for all $\tilde{z}_x \in B_1$, $\|\tilde{z}_x - F(\tilde{z}_x)\| < \varepsilon_z$, we know that all elements in the ball $B_{1-\varepsilon_z}$ are in the range of $F(B_1)$. Therefore, for all $\tilde{z}_x^* \in B_{1-\varepsilon_z}$, there exists a $\tilde{z}_x \in B_1$, such that the same bound (4.4) holds.

Note that the reconstruction error estimate depends on $\|\varphi_{\text{dec}}\|_\text{op}$ and $\varepsilon_z$. The former can be kept small by regularising the weights of the decoder in the training phase. The latter can be kept small by including $\|\tilde{z}_x - \Sigma \varphi_{\text{enc}}(A \varphi_{\text{dec}}(\tilde{z}_x))\|_2$ in the cost function, as described in the last point of section 3.1.1. After training, points in the unit ball can be sampled uniformly and passed through the network to examine an actual value for $\varepsilon_z$.
5. Research Context. Our paper presents the L-SVD method for solving inverse problems via hybrid autoencoding. Within the method, a low-dimensional (i.e. sparse) representation or manifold is explicitly learned. It is clear that this model has connections with many research fields, which we will point out in this section.

5.1. Solving inverse problems using partially data-driven models. Recent research in inverse problems seeks to develop a mathematically coherent foundation for combining data-driven deep learning with model-driven approaches based on physical-analytical domain knowledge [3].

A first class of methods are partially learned variational and iterative methods [1, 24, 19, 8]. These methods can be seen as a learned variant of gradient, proximal or primal-dual methods. They require less iterations than their non-learned counterparts, but the demand on training time is substantial, while the mathematical analysis of these methods is limited.

A second approach is to learn an explicit regularisation term [11, 27, 2]. Signals affected by artefacts are penalised, while the desired signals are not. Reconstructions are of higher quality compared to classical regularisation choices, but their computation time is of the same order.

A third approach is to perform learned post-processing of initial reconstructions obtained by classical methods, which may be affected by artefacts [21, 35]. Data-consistent reconstructions can be obtained without an iterative procedure [35]. However, the quality of the reconstructions heavily depends on the initial reconstruction, which is often obtained by applying a pseudo-inverse to the data.

All methods depend on precise knowledge of the physical process, which is not always available. It is an open question if such methods can be used in case of inexact forward operators [19]. Moreover, interpretation of the methods is limited, since the connection between architecture and operator structure is not clear.

5.2. Fully learned image reconstruction. Our work is closely related to the work of Zeng et al. [42]. In this paper, the task of superresolution is solved by autoencoding patches of both a low- and high-resolution image and finding a nonlinear mapping between them. Gupta et al. [18] used this method for the task of removing motion blur, which is a specific case of a deconvolution problem. In both cases, a one-layered autoencoder was applied to patches of the distorted image (measurement) and desired image (signal). This is only possible if measurement and signal lie in the same domain and if the forward mapping has no to little effect outside the local patch. We consider a more general method that does not work patch-based and therefore does not assume identical domains for measurement and signal. As a result, the forward mapping may have a global behaviour.

A different fully learned reconstruction method without this restriction is proposed by Zhu et al. [44], where the problem of finding a reconstruction from undersampled MRI data is considered. Their neural network consists of three fully-connected layers, followed by three convolutional layers, which maps Fourier measurements directly to the desired image. A joint low-dimensional manifold is learned implicitly, since there is no explicit low-dimensional representation within the network architecture. In our work, an explicit representation of a joint manifold is learned in the form of two linearly connected latent codes. Results in [44] show a clear improvement over non-learned state-of-the-art methods for in-vivo data.

These works display the potential for fully learned methods in image reconstruction and inverse problems in general: high quality reconstructions are obtained, while no exact knowledge of underlying physics or specifics of the measurement system is required.
5.3. Manifold learning. Many relevant inverse problems in medicine, engineering, astronomy or geophysics are large-scale in both signal and measurement space. However, seen from a statistical point of view, probability mass concentrates near manifolds of a much lower dimension than the original data spaces [4]. To detect linear manifolds, principal component analysis is a suitable and simple method. However, since manifolds for real data are expected to be strongly nonlinear [4], one needs to make use of nonlinear techniques. One of the best known methods that achieves this is kernel PCA [34]. In this method, data is mapped to a reproducing kernel Hilbert space by applying a nonlinear kernel, after which the linear PCA is applied. Other methods are principal geodesic analysis [15], which can be applied for Riemannian manifolds and geodesic PCA [7], which acts in a Wasserstein space, which is nonlinear.

Another approach to learn nonlinear manifolds is to use autoencoders, which can also be seen as a generalisation of linear PCA [20]. Autoencoders have shown to learn explicit representations of nonlinear manifolds [31] and provide better low-dimensional latent code in terms of clustering and reconstruction performance [20] than their linear counterpart. For the inverse problems (2.1) that we consider, there is an explicit relationship between signals and measurement via the forward mapping that models the physics. This means that signals and measurements share one data manifold, which we learn by connecting two autoencoders.

The observation of a shared manifold, or “an unknown underlying relationship between two domains” [45], is also the idea driving the cycle GAN. Unlike in our paper, the goal of the cycle GAN is not to find a unique and supervised one-to-one mapping from one domain to the other, but to identify the shared parameters and add such elements so that the output is realistic in its respective domain. This cycle GAN has been applied for inverse problems in different forms [36, 37]. In these works, the manifold is implicitly learned, unlike the explicitly learned representation that we study in our paper.

5.4. Transfer learning with autoencoders. Transfer learning is used to exploit similarities between different tasks to share information necessary for both tasks. Representation learning, such as manifold learning, has a strong influence in transfer learning scenarios [4] since the learned representation can guide the supervised reconstruction task. Recent research has shown that autoencoders can be used as a regulariser for a supervised training task, such as classification [43, 25]. Such networks, coined supervised autoencoders (SAE) help to generalise the supervised problem. They are specifically useful in a semi-supervised scenario, where a lot of unsupervised training data is available, but supervised training pairs are scarce.

L-SVD profits from the same regularisation and generalisation effect by attaching two autoencoders to the supervised reconstruction problem. For inverse problems a semi-supervised scenario is also often encountered: imagine a training set of undersampled medical MRI or CT data sets and a set of high-quality reconstructions; not all pairs are available because not all patients have had a fully sampled scan that is needed for a high-quality reconstruction.

5.5. Model reduction and learning. Model reduction is a mathematical and computational field of study that derives low-dimensional models of complex systems [10, 5]. Via projections and decompositions it is possible to represent approximations of large-scale high-fidelity computational models resulting from discretization of partial differential equations. Recent developments focus on data-driven learning of governing equations [9, 33, 29] and learned model reduction [28].

Our work focuses on learning the inverse map for problems that are often physics-driven. The nonlinear equations or parameters of this map are implicitly learned through the latent representations via autoencoders.
5.6. Bayesian inversion and sparsity. The goal of the Bayesian approach to solving inverse problems is to find the posterior measure, given sampled data and a prior measure \[39\]. The posterior measure will contain information about the relative probability of different outputs, given the data. Often the posterior is too complex to recover and the goal is shifted to finding a maximum a posteriori (MAP) estimate. In section 4.1 we have shown that a linear variant of L-SVD coincides with learning the covariance matrix of the prior measure.

Yu et al. [41] developed piecewise linear estimates (PLE), a method based on Gaussian mixture models (GMM), which are estimated via the MAP expectation-maximization algorithm (MAP-EM). The method makes use of GMM as prior measures on local patches, which results in a linear reconstruction model for each patch. One could think of this procedure as finding a patchwork of locally linear tangent spaces which approximate a nonlinear manifold. If measurement and signal domain are the same, it can be shown that PLE is equivalent to learning a linear L-SVD method for a group of similar patches.

6. Experiments and implementation. In this section the neural network architectures together with all their parameter choices are provided. Before that, two simulation experiments that demonstrate the contributions as stated in section 1.1 are explained. The forward operator in all experiments is chosen to be the Radon transform [23], which is a nonlocal linear operator that produces an ill-posed inverse problem. The operator is applied to the MNIST [26] data set, after rescaling it to \(64 \times 64\) pixels. We will refer to the measurements as ‘sinograms’ and to the signals as ‘images’ in their respective spaces \(Y\) and \(X\).

6.1. Experiment 1: from model-driven to data-driven. The goal of this experiment is to demonstrate the first perspective of section 1.1: data-driven solution of inverse problems. This is done by comparing classical non-learned methods to learned methods. For fair comparison and clarity, only linear inversion methods are considered.

We apply the Radon transform that makes use of 64 uniformly samples angles, i.e. a ‘full-angle’ Radon operator \(A\). Moreover, there is no bottleneck latent space. This results in equally large spaces \(Y = Z^y = Z^x = X = \mathbb{R}^{4096}\). For training, Gaussian noise with a noise level \(\delta = 0.05\) was added to the simulated full-angle measurements, i.e. a standard deviation of 0.05 times the maximum sinogram intensity. The following linear reconstruction methods will be compared:

(a) Tikhonov-regularised reconstruction with optimally chosen \(\alpha\) (section 2.2);
(b) truncated SVD reconstruction with optimal truncation number \(k\) (section 2.3);
(c) reconstruction from a learned covariance matrix of the prior, i.e. \(U\) and \(V\) are obtained from the SVD, while \(\Sigma\) is a learned matrix that is either diagonal or full (section 4.1.1);
(d) L-SVD: fully learned reconstruction, where \(\Sigma\) is chosen to be a diagonal matrix. We use a regular autoencoder on the sinogram side, which means that noise should be reconstructed after the sinogram is encoded and decoded.

Besides comparing the reconstruction quality of above methods, we will examine the ‘dictionary’ of elements by decoding canonical basis vectors of the latent space: since all reconstruction methods are linear, the attainable reconstructions lie in the span of this dictionary. By this examination, the change from model-driven to data-driven methods becomes evident.

Furthermore, by training on sinograms with a noise level between 0 and 0.2, instead of the aforementioned 0.05, the effect of noise on the matrix \(\Sigma\) will be investigated. Instead of training for several noise levels individually, only one training set is created, where every sinogram has a randomly chosen noise level \(\delta \in [0, 0.2]\). To process this data set adequately, the static scaling matrix \(\Sigma\) is exchanged for a ‘noise-dependent’ component (c.f. section 3.1.1). This component is a nonlinear fully connected network which takes \(z_y\) as input and provides the diagonal scales \((\sigma_1, \sigma_2, \ldots, \sigma_k)\) as output, which are multiplied with \(z_y\).
6.2. Experiment 2: from linear autoencoding to hybrid nonlinear autoencoding. The goal of this experiment is to demonstrate the second and third perspective of section 1.1: nonlinear encoding is more effective than linear encoding; moreover, combining two nonlinear autoencoders has a regularising effect on the reconstruction and gives a more insightful latent representation than one autoencoder.

A limited-angle Radon transform of 8 uniformly sampled angles is applied in this simulation experiment. The latent space is chosen to have 64 dimensions, which means that it acts as a bottleneck. This results in the spaces \( Y = \mathbb{R}^{256}, Z_y = Z^x = \mathbb{R}^{64}, X = \mathbb{R}^{4096}, \) providing a dimensionality reduction of 12.5% and 1.56% compared to sinogram and image space respectively. Gaussian noise with a noise level \( \delta = 0.05 \) is added to the limited-angle measurements, i.e. a standard deviation of 0.05 times the maximum sinogram intensity. We analyse the following methods:

(a) linear autoencoder;
(b) nonlinear autoencoder;
(c) linear L-SVD;
(d) nonlinear L-SVD;
(e) nonlinear L-SVD (\( \alpha = 0 \)).

Here the first two methods are only applied on the image side and not on the sinogram side, meaning that the autoencoder only connects \( X \) and \( Z^x \). As can be seen in Figure 3.1, L-SVD connects the sinogram side with the image side, where a denoising autoencoder (DAE) is used on the sinogram side. This means that noise should not be reconstructed after decoding to \( Y \), which allows for a dimensionality reduction. The last method has the same network structure of nonlinear L-SVD, but without the autoencoders on either side (i.e. \( \alpha_y = \alpha_x = 0 \) in (3.1)), which allows us to investigate the regularising effect of the autoencoders. It is obvious that this experiment investigates both the second and third perspective from section 1.1, since both the transition from linear to nonlinear is made, as well as the transition from a single to hybrid autoencoder.

6.3. Implementation. In the first experiment, all encoders and decoders are one-layer neural networks. In the second experiment, the encoder and decoder on the sinogram side are also one-layer networks, while the encoder and decoder on the image size consist of four layers. After each layer, except the last layer, a leaky ReLU function is applied with a parameter \( \gamma = 0.1 \). All networks are chosen to be without biases. The initial weights are normally distributed with a standard deviation of 0.01. All loss-functions \( D_j \) in (3.1) are chosen to be of \( \ell^2 \)-type (mean squared error) and in all experiments \( \alpha_y = 2 \) and \( \alpha_x = 1 \), except for the last network in experiment 2, which uses \( \alpha_y = \alpha_x = 0 \). The scale-dependent network component of experiment 1 consists of five fully connected layers with biases. Its initial weights are normally distributed with a standard deviation of 0.01; biases in the first four layers are initialised with 0 and in the fifth layer with 1. After each layer a leaky ReLU with parameter \( \gamma = 0.1 \) is applied, except for the last layer, which uses a softplus function.

Training is done in Tensorflow on the complete MNIST training set (50000 training samples), where each image has been rescaled to \( 64 \times 64 \) using bilinear interpolation. Testing is done on the complete MNIST test set (10000 samples), using the same interpolation. The Radon transform has been applied using the ‘scikit-image’ toolbox in Python.

Training is done for 250 epochs using a batch size of 100. We use the ADAM optimiser with a learning rate with exponential weight decay, starting at \( 10^{-3} \) and ending at \( 2 \cdot 10^{-4} \). All other parameters are the default choices of ADAM in Tensorflow. Gradient norm clipping with a value of 10 is applied for training stability. No regularisation, dropout or batch normalisation were used.
7. Results. In this section, the results of the simulation experiments explained in section 6 are shown and discussed.

7.1. Experiment 1: from model-driven to data-driven. Results for the methods as explained in section 6.1 were obtained by finding the optimal parameters $\alpha$ and $k$ for methods (a) and (b) and training the networks (c) and (d) with the implementation as described in section 6.3. For a randomly chosen MNIST sample in the test set, the outputs of all methods are shown in Figure 7.1. For this sample, Gaussian noise with a noise level $\delta = 0.05$ was added. Visually, the reconstruction improves gradually as we move from model-driven methods with only one tunable parameter (b,c) to combinations of model- and data-driven methods with tunable scaling matrix (d,e) to fully data-driven methods (f,g). This is most noticeable in the background of the reconstruction, which should be constant.

![Figure 7.1: Reconstructions $\hat{x}$ (see Figure 3.1) for different models ranging from fully model-driven (left) to more data-driven (right) with increasing amount of learning.](image)

In Table 7.1, the mean squared error (MSE) test losses of all compared methods are shown. For this test set, Gaussian noise with a standard deviation of 0.05 times the maximum sinogram intensity was added. The reconstruction test loss shows that higher quality reconstructions are obtained as the methods become more data-driven. For the fully data-driven method of L-SVD, autoencoder (AE) test losses are much smaller than reconstruction test losses, which is most likely due to the noise in the sinograms that is not added to the AE input.

| Network                   | Reconstruction test loss | AE($x$) test loss | AE($y$) test loss |
|---------------------------|--------------------------|-------------------|-------------------|
| Tikhonov                  | 17.4                     | 0                 | 0                 |
| truncated SVD             | 6.70                     | 3.72              | 4.94              |
| learned diagonal $\Sigma$, fixed $U,V$ | 5.00                     | 0                 | 0                 |
| learned full $\Sigma$, fixed $U,V$ | 3.88                     | 0                 | 0                 |
| L-SVD initialised with SVD | 2.24                     | 0.095             | 0.26              |
| L-SVD initialised randomly | 1.93                     | 0.058             | 0.25              |

Table 7.1: Comparison of test losses (MSE) for different models ranging from fully model-driven to more data-driven with increasing amount of learning.

To understand the transition from model-driven to data-driven, canonical basis vectors in the latent space are decoded to the image space for both the regular SVD and learned SVD. Four selected elements from this ‘dictionary’ are shown in Figure 7.2.

SVD decomposes the Radon operator in different elements with a different geometrical scale. Moreover, it combines higher order harmonics in the image space and the sinogram.
(a) SVD decoded to $X$
(b) L-SVD decoded to $X$
(c) SVD decoded to $Y$
(d) L-SVD decoded to $Y$

Figure 7.2: Selected elements in the latent space $Z^x$, decoded to the image space $X$ and the sinogram space $Y$. SVD only makes use of the operator, while L-SVD combines operator information with image and sinogram information. This results in more localised information in the decoded elements of the data-driven L-SVD approach.

Finally, we analyse the effect of noise on the diagonal scaling $\Sigma$ with the network component as explained at the end of section 6.1. We compare the network where encoder and decoder are fixed as $U^*$ and $V$ with the network in which encoder and decoder are also learned. Both networks are compared with Tikhonov regularisation, in which only one tunable parameter $\alpha$ is chosen in such a way that the smallest MSE is obtained. Gaussian noise has been added with 9 different intensities (noise levels) to all sinograms in the test set. The average scales per noise level are shown in Figure 7.3. For visualisation purposes, the graphs have been smoothed by applying Gaussian filtering with a scale of 10.

It can be seen that all methods show a decay of scales as the noise level grows. For the scales that coincide with small $s_i$ in the SVD case, i.e. the first dimensions in (a-c), this decay is relatively large. A similar effect is seen in the middle of (d). In (b-d), where every scale was learned individually, it seems that only about 500 dimensions are relatively noise independent, as they decay less with increasing noise. This can be seen in dimensions 2500 to 3000 in (b-c) and the far right dimensions in (d). Finally, all graphs show more than a thousand dimensions that have a small scale for all noise levels: dimensions 3000-4000 in (a-c) and dimensions most left in (d). To sum up, the behaviour of the scales to noise is similar for all methods in which the scales are learned individually, regardless of the encoding and decoding used. Moreover, most structural information is encoded in a limited number of latent dimensions, while the other dimensions encode a substantial amount of noise: a compression is learned.

7.2. Experiment 2: from linear autoencoding to hybrid nonlinear autoencoding. The autoencoder and L-SVD networks as explained in section 6.2 were trained with the implementation as described in section 6.3. For a randomly chosen MNIST sample in the test set,
Figure 7.3: Comparison of the noise dependency of the scales $s_i$ for methods with increasing amount of learning. Although similar, methods (b-d), in which the scales are learned individually, show a greater noise dependency than (a) Tikhonov regularisation (see (2.3)). Note that methods (a-c) use the SVD ordering, while (d) is ordered from small to large scales.

The outputs of all methods are shown in Figure 7.4. Note that for all autoencoder outputs $\hat{x}^{AE}$, no noise was added to their inputs $x$, while for the reconstruction outputs $\hat{x}^{SE}$, noise with a noise-level of 0.05 was added to their inputs $y^\delta$.

Figure 7.4: Comparison of the outputs $\hat{x}^{AE}$ and $\hat{x}^{SE}$ (see Figure 3.1) for linear and nonlinear variants of the AE and L-SVD network.

It can be seen that all linear methods produce side-lobes to the main intensities, which are often observed in frequency-based compressions. The nonlinear methods provide a more homogeneous background in the reconstruction, especially L-SVD. Visually, there is little difference between the L-SVD autoencoder output $\hat{x}^{AE}$ and the L-SVD reconstruction output $\hat{x}^{E}$.

In Table 7.2, the train and test losses of all compared methods are shown. The first thing that can be seen is that for the linear networks, errors are larger then for the nonlinear networks. The second thing is that there is no significant difference between train and test loss.
| Type   | Network | Output | Train loss | Test loss |
|--------|---------|--------|------------|-----------|
| linear | AE      | \(\hat{x}^{AE}\) | 10.2      | 10.1      |
|        | L-SVD   | \(\hat{x}^{AE}\) | 10.5      | 10.3      |
|        | L-SVD   | \(\hat{x}^{\Sigma}\) | 13.7      | 13.4      |
| nonlinear | AE    | \(\hat{x}^{AE}\) | 0.40      | 4.61      |
|        | L-SVD\(\alpha=0\) | \(\hat{x}^{\Sigma}\) | 1.22      | 5.34      |
|        | L-SVD   | \(\hat{x}^{AE}\) | 1.49      | 3.97      |
|        | L-SVD\(\alpha=0\) | \(\hat{x}^{\Sigma}\) | 2.00      | 3.94      |

Table 7.2: Comparison of train and test loss (MSE) for linear and nonlinear variants of the AE and L-SVD network. Nonlinear L-SVD with hybrid autoencoder generalises better than nonlinear AE and nonlinear L-SVD\(\alpha=0\).

for the linear networks, which indicates that they generalise well. This difference is present in the nonlinear networks, but for L-SVD not to the same extent as for AE en L-SVD\(\alpha=0\). From the autoencoder point of view, it seems that L-SVD benefits from the sinogram branch of the network in terms of generalisation. In other words, adding a second autoencoder to the network regularises the autoencoder output \(\hat{x}^{AE}\). From a reconstruction point of view, L-SVD benefits from the incorporation of the two autoencoders, which contribute to its generalisation capacity in the reconstruction output \(\hat{x}^{\Sigma}\). Finally, since the test loss for nonlinear L-SVD is smaller than for L-SVD\(\alpha=0\), we can conclude that the two autoencoders act as regularisers for the reconstruction.

Next, canonical basis vectors in the latent space are decoded to the image space, to compose a ‘dictionary’ of elements. While in the linear case all outputs can be reconstructed from these elements in a linear way, this is not true for the nonlinear case. This means that the dictionary only gives a partial view on the decoder.

![Figure 7.5](image)

Figure 7.5: Selected elements in the latent space \(Z^x\), decoded to the image space \(X\). Nonlinear L-SVD\(\alpha=0\) and nonlinear L-SVD learn a more interpretable representation than other methods by combining features from sinogram and image space in their ‘dictionary’. Due to its similarity to linear L-SVD, linear AE is not shown here. All 64 elements are shown in Appendix B.

Figure 7.5 shows four selected elements that are exemplary for the complete dictionaries, which are provided in Appendix B. Because the elements of linear AE are very similar to linear L-SVD, they are not shown here. The top left image of the linear network shows the Euclidean mean of all training images, while the top right and bottom images show low
and high frequency components that are only active in the part of the image where MNIST digits are positioned. Nonlinear AE shows much smaller structures in its elements, which do not seem to have a visual coherent structure. The nonlinear networks $\text{L-SVD}_{\alpha=0}$ and L-SVD do provide this visually coherent structure, where L-SVD seems to provide somewhat ‘smoother’ and better connected structures than $\text{L-SVD}_{\alpha=0}$. Their dictionaries consist of various elements, of which one is similar to the Euclidean mean (top left), some are similar to digits (top right), some that show a combination of line segments (bottom left) and some with high-frequency components (bottom right) which were also visible in the linear dictionary. With this diversity, L-SVD combines information from images, sinograms and operator.

Figure 7.6: Euclidean interpolation between two randomly chosen test samples in $Z^2$, decoded to the image space $X$. Nonlinear L-SVD creates interpolations with clean curves on a constant background that have a strong perceptual similarity with actual handwritten digits.

The dictionary in Figure 7.5 was created by decoding basis vectors in latent space. In the nonlinear case, it is hard to interpret these dictionary elements, because actual digits might not be located close to these basis vectors. For this reason, an interpolation between two digits is shown in Figure 7.6. For this, two random samples in the MNIST test set have been encoded to the latent space. Next, the two latent codes have been linearly interpolated in the latent space. Figure 7.6 shows the images after decoding. Again linear AE is not shown due to its similarity to linear L-SVD.

The linear methods show a pointwise linear interpolation in the image space. Nonlinear AE provides a sudden appearance and disappearance of line segments; its background shows undesirable local intensity changes. The interpolation in nonlinear $\text{L-SVD}_{\alpha=0}$ shows likewise but less severe intensity changes in its background. nonlinear $\text{L-SVD}_{\alpha=0}$ and L-SVD both provide sudden appearance and disappearance of line segments. Combined with a background intensity that stays constant throughout the interpolation, L-SVD generates images that have a strong perceptual similarity with actual handwritten digits, which is not the case with the other methods.

8. Conclusion and Outlook. We have proposed the learned SVD for inverse problems: a general method for fully learned reconstruction that connects low-dimensional nonlinear representations of corrupted measurements and desired signals. Depending on the specific
task at hand, the connecting layer can be fully connected or sparse, linear or nonlinear, noise dependent or noise independent. One specific choice is to incorporate the necessary nonlinearity of the learned inverse function in the autoencoders, while the sparse diagonal scaling layer is chosen to be linear, making the connection between measurement and signal manifold easy to understand. In simulation experiments, it was shown that this nonlinear reconstruction gives superior performance to other methods, while providing interpretable autoencoding. Moreover, since the reconstruction error estimate depends on the autoencoding quality, L-SVD can greatly benefit greatly from general advances in nonlinear autoencoding.

Results show that L-SVD makes use of information from both measurements and signals; by doing so, it learns elements of the physics operator, although not explicitly provided. Therefore the method is especially promising in applications where the forward physics are not completely understood or computationally expensive to simulate. Learning a joint manifold by two connected autoencoders also enables the possibility of a semi-supervised setup: the autoencoders provide regularisation for reconstruction of the signal.

Due to its generic formulation, L-SVD is very flexible for other architecture choices in autoencoding. Therefore, future efforts will lie in investigating other architectures, such as convolutional autoencoders and ladder variational autoencoders [38], for their inclusion in L-SVD for large scale inverse problems. Finally, other loss functions, such as the Wasserstein loss or a learned discriminator could be investigated.

Acknowledgements. We thank Kathrin Smetana, Leonie Zeune and Srirang Manohar for valuable discussions on the topic. The collaboration project is co-funded by the PPP allowance made available by Health~Holland, Top Sector Life Sciences & Health, to stimulate public private partnerships. CB acknowledges support by the 4TU programme Precision Medicine.

REFERENCES

[1] J. Adler and O. Öktem, Solving ill-posed inverse problems using iterative deep neural networks, Inverse Problems, 33 (2017), p. 124007, https://doi.org/10.1088/1361-6420/aa9581.
[2] H. K. Aggarwal, M. P. Mani, and M. Jacob, Modll: Model-based deep learning architecture for inverse problems, IEEE transactions on medical imaging, 38 (2018), pp. 394–405.
[3] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, Solving inverse problems using data-driven models, Acta Numerica, 28 (2019), pp. 1–174, https://doi.org/10.1017/S0962492919000069.
[4] Y. Bengio, A. Courville, and P. Vincent, Representation learning: A review and new perspectives, IEEE transactions on pattern analysis and machine intelligence, 35 (2013), pp. 1798–1828.
[5] P. Benner, S. Gugercin, and K. Willcox, A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems, SIAM Review, 57 (2015), pp. 483–531, https://doi.org/10.1137/130932715, http://epubs.siam.org/doi/10.1137/130932715.
[6] M. Benning and M. Burger, Modern regularization methods for inverse problems, Acta Numerica, 27 (2018), pp. 1–111.
[7] J. Bigot, R. Gouet, T. Klein, A. López, et al., Geodesic pca in the wasserstein space by convex pca, in Annales de l’Institut Henri Poincaré, Probabilités et Statistiques, vol. 53, Institut Henri Poincaré, 2017, pp. 1–26.
[8] Y. E. Bonk, S. Manohar, and C. Brune, A Partially Learned Algorithm for Joint Photoacoustic Reconstruction and Segmentation, IEEE Transactions on Medical Imaging, (2019), pp. 1–11, https://doi.org/10.1109/TMI.2019.2922026.
[9] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proceedings of the National Academy of Sciences, 113 (2016), pp. 3932–3937, https://doi.org/10.1073/pnas.1517384113.
[10] T. Bu-Thanh, K. Willcox, and O. Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space, SIAM Journal on Scientific Computing, 30 (2008), pp. 3270–3288.
[11] H. Chen, Y. Zhang, Y. Chen, J. Zhang, W. Zhang, H. Sun, Y. Lv, P. Liao, J. Zhou, and
[32] W. Rudin, Principles of Mathematical Analysis -, McGraw-Hill, New York, 3. auff. ed., 1976.
[33] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Data-driven discovery of partial differential equations, Science Advances, 3 (2017), p. e1602614, https://doi.org/10.1126/sciadv.1602614.
[34] B. Schölkopf, A. Smola, and K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural computation, 10 (1998), pp. 1299–1319.
[35] J. Schwab, S. Antholzer, and M. Haltmeier, Deep null space learning for inverse problems: convergence analysis and rates, Inverse Problems, (2018).
[36] O. Senouf, S. Vedula, T. Weiss, A. Bronstein, O. Michailovich, and M. Zibulevsky, Self-supervised learning of inverse problem solvers in medical imaging, arXiv preprint arXiv:1905.09325, (2019).
[37] B. Sim, G. Oh, S. Lim, and J. C. Ye, Optimal transport, cyclegan, and penalized ls for unsupervised learning in inverse problems, arXiv preprint arXiv:1909.12116, (2019).
[38] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, Ladder variational autoencoders, in Advances in neural information processing systems, 2016, pp. 3738–3746.
[39] A. M. Stuart, Inverse problems: A Bayesian perspective, Acta Numerica, 19 (2010), pp. 451–559, https://doi.org/10.1017/S0962492910000061.
[40] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion, Journal of Machine Learning Research, 11 (2010), pp. 3371–3408, http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf.
[41] G. Yu, G. Sapiro, and S. Mallat, Solving Inverse Problems With Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity, IEEE Transactions on Image Processing, 21 (2012), pp. 2481–2499, https://doi.org/10.1109/TIP.2011.2176743.
[42] K. Zheng, J. Yu, R. Wang, C. Li, and D. Tao, Coupled Deep Autoencoder for Single Image Super-resolution, IEEE Transactions on Cybernetics, 47 (2017), pp. 27–37, https://doi.org/10.1109/TCYB.2015.2501373, http://ieeexplore.ieee.org/document/7339460/.
[43] Y. Zhang, K. Lee, and H. Lee, Augmenting Supervised Neural Networks with Unsupervised Objectives for Large-scale Image Classification, in International Conference on Machine Learning (ICML), vol. 48, New York, jun 2016, JMLR, pp. 612–621, https://arxiv.org/abs/1606.06582.
[44] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, Image reconstruction by domain-transform manifold learning, Nature, 555 (2018), pp. 487–492, https://doi.org/10.1038/nature25988.
[45] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, in The IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2223–2232, https://arxiv.org/abs/1703.10593.
A. Proof of Proposition 4.1.

Proof. We will first take a look at the more general case where \( m_0 \sim \mathcal{N}(m_0, C_0) \) before we set \( m_0 = 0 \). We determine the posterior measure \( \mu_{\text{post}} \) for \( x \) given \( \tilde{y} \) (see equation (3.4) in [39]) as

\[
\mu_{\text{post}} = \mathcal{N}(m_{\text{post}}, C_{\text{post}})
\]

where

\[
m_{\text{post}} = m_0 + C_0 A^* (B + AC_0 A^*)^{-1} (\tilde{y} - A m_0)
\]

and

\[
C_{\text{post}} = C_0 - C_0 A^* (B + AC_0 A^*)^{-1} A C_0.
\]

The maximum a posteriori (MAP) estimate \( x_{\text{MAP}} := \arg \max_x p(y|x) \) coincides with the mean of \( \mu_{\text{post}} \), i.e.

\[
u_{\text{MAP}} = \arg \min_{x} \{ \| Ax - \tilde{y} \|_B^2 + \| x - m_0 \|_{C_0}^2 \} = m_{\text{post}}.
\]

We substitute \( A = U S_n V_n^* \) in (A.1), from which we obtain

\[
m_{\text{post}} = m_0 + C_0 V_n S_n^* U^* (B + U S_n V_n^* C_0 V_n S_n^* U^*)^{-1} (\tilde{y} - U S_n V_n^* m_0)
\]

\[
= m_0 + C_0 V_n S_n^* U^* (\tilde{B} U^* + U S_n V_n^* C_0 V_n S_n^* U^*)^{-1} (\tilde{y} - U S_n V_n^* m_0)
\]

\[
= m_0 + C_0 V_n (S_n^{-1} \tilde{B} (S_n^*)^{-1} + V_n^* C_0 V_n)^{-1} (S_n^{-1} U^* \tilde{y} - V_n^* m_0)
\]

\[
= m_0 + V_n C_{V_n} (S_n^{-1} \tilde{B} (S_n^*)^{-1} + C_{V_n})^{-1} (S_n^{-1} U^* \tilde{y} - V_n^* m_0),
\]

where we used \( \tilde{B} := U^* B U \) in the second equality and \( C_0 = V_n C_{V_n} V_n^* \) in the last equality. In case of a prior distribution with zero-mean, i.e. \( m_0 = 0 \), we get the expression

\[
m_{\text{post}} = V_n C_{V_n} (S_n^{-1} \tilde{B} (S_n^*)^{-1} + C_{V_n})^{-1} S_n^{-1} U^* \tilde{y}.
\]

The symmetric covariance matrix \( C_{V_n} \) is positive definite, hence it is also invertible. We simplify

\[
\Sigma^{-1} = S_n (S_n^{-1} \tilde{B} (S_n^*)^{-1} + C_{V_n})^{-1} S_n^{-1} U^* \tilde{y}
\]

\[
= \tilde{B} (C_{V_n} S_n)^{-1} + S_n.
\]

By substituting this expression for \( \Sigma \) in (A.2), we obtain

\[
m_{\text{post}} = V_n \Sigma U^* \tilde{y}
\]

with \( \Sigma = \left[ \tilde{B} (C_{V_n} S_n)^{-1} + S_n \right]^{-1} \).

\[
\square
\]

20
B. Visualisation of all latent space elements. Figure B.1 provides the complete dictionaries for all methods, from which a selection was shown in Figure 7.5. For a discussion on the results we refer to section 7.2.

Figure B.1: All 64 elements of the dictionary of all methods, from which a selection was shown in Figure 7.5.