
Rank Reduction Autoencoders - Enhancing interpolation on nonlinear manifolds.

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 The efficiency of classical Autoencoders (AEs) is limited in many practical situations.
2 When the latent space is reduced through autoencoders, feature extraction
3 becomes possible. However, overfitting is a common issue, leading to “holes” in
4 AEs’ interpolation capabilities. On the other hand, increasing the latent dimension
5 results in a better approximation with fewer non-linearly coupled features (e.g.,
6 Koopman theory or kPCA), but it doesn’t necessarily lead to dimensionality re-
7 duction, which makes feature extraction problematic. As a result, interpolating
8 using Autoencoders gets harder. In this work, we introduce the Rank Reduction
9 Autoencoder (RRAE), an autoencoder with an enlarged latent space, which is
10 constrained to have few dominant singular values (i.e., low-rank). The latent space
11 of RRAEs is large enough to enable accurate predictions while enabling efficient
12 feature extraction. As a result, the proposed autoencoder features a minimal rank
13 linear latent space. To achieve what’s proposed, two formulations are presented, a
14 strong and a weak one, that build a reduced basis accurately representing the latent
15 space. The first formulation consists of a truncated SVD in the latent space, while
16 the second one adds a penalty term to the loss function. We show the efficiency of
17 our formulations by using both of them for interpolation tasks and comparing the
18 results to state-of-the-art autoencoders on both synthetic data and MNIST.

19

1 Introduction

20 Interpolation of vector functions over a parametric space is an active research topic since accurate
21 interpolation allows the reconstruction of a physical solution in an entire parametric space from
22 a set of pre-computed samples. Multiple techniques have been proposed to perform interpolation,
23 to mention a few, the Proper Orthogonal Decomposition with Interpolation (PODI) [24, 17, 19],
24 or the sparse-PGD (sPGD) [7]. Most of these techniques are based on Model Order Reductions,
25 such as the Proper Orthogonal Decomposition (POD) [12], the Proper Generalized Decomposition
26 (PGD) [21, 6], and the Principal Component Analysis (PCA) [9]. These techniques stack the vector
27 functions in what is called the solution matrix, and their efficiency is inversely proportional to the
28 rank of this matrix. If the solution matrix only has a few dominant singular values (i.e., low-rank), it
29 is easier for the aforementioned techniques to reduce the problem and interpolate. However, when
30 this assumption does not apply, they fail to define an efficient surrogate for the correct prediction
31 of physical phenomena. A high-rank solution matrix reduces the efficiency of techniques based on
32 different formulations such as those based on Grassmann manifolds [1], the Optimal Transport (OT)
33 [25], or every low-rank technique (e.g. [22, 14]).

34 On the other hand, the nonlinearity in autoencoders with Neural Networks as their encoding and
35 decoding functions makes them appealing to be used as interpolators [10]. Their modified archi-
36 tectures have been used for different purposes in many applications such as speech recognition [4],

37 biophysics [13], medicine [18], and others [2]. Yet, the efficiency of Vanilla Autoencoders is limited.
 38 On one hand, Autoencoders with reduced latent spaces (or Diabolo Autoencoders (DAEs)), can
 39 easily overfit the data and are known for having “holes” in their latent spaces. On the other hand,
 40 even though enlarged latent spaces lead to better approximations with less non-linear behavior (as
 41 stated in multiple theories like the Koopman theory or the kPCA), the representations learned are
 42 usually of a large dimension which limits both interpolation and feature extraction. This has led to
 43 multiple enhancements such as Variational AEs [8, 26], Sparse AEs [20], and denoising AEs [27]
 44 which improved Autoencoders overall but did not definitively solve the interpolation issues.

45 Neural Networks are increasingly being used for nonlinear reduction techniques [3, 5]. Recently, the
 46 Implicit Rank-Minimizing Autoencoder (IRMAE) [11], and the Low-Rank Autoencoder (LoRAE)
 47 [15] showcased how increasing the latent space dimension while encouraging a low-rank achieves
 48 better results, for both approximation and interpolation. If the latent space is of low rank, the
 49 efficiency of all presented interpolation techniques (including basic linear interpolation) in the latent
 50 space is enhanced. The resulting Autoencoder would benefit from the large data dimensionality
 51 of the latent space to find better approximations while allowing feature extraction because of its
 52 low rank. The architecture of IRMAEs consists of adding linear layers between the encoder and
 53 the latent space, while LoRAE only adds one linear layer as well as its nuclear norm as a penalty
 54 term in the loss. While both papers show how their resulting latent spaces may exhibit a lower rank
 55 compared to Vanilla and Variational Autoencoders, their work has some limitations. First, while
 56 both architectures may find a low-rank latent space with singular values that are sharply decreasing,
 57 they do not enforce the small singular values to go to zero. Accordingly, the decoder always has
 58 some noise from the small singular values, even though ideally, we would like to remove their effect
 59 entirely. In addition, the computational time of both architectures highly depends on the latent space
 60 dimension L . Since a long latent space is crucial for achieving better results, both IRMAEs and
 61 LoRAEs can be computationally expensive. Finally, both architectures do not provide explicit control
 62 over the rank of the latent space. While they include some tuning parameters, we show later in the
 63 paper that their proposed parameters can not reach a satisfying low-rank space.

64 In this paper, we present the Rank Reduction Autoencoder (RRAE), which has a large latent space
 65 restricted to have a low rank. By enforcing the latent space to accept a linear reduction (hence a
 66 lower rank), we show that our model resolves the issues previously mentioned. Our architecture
 67 includes two proposed formulations: (i) a strong and (ii) a weak one. Throughout the paper, we show
 68 that the strong formulation finds orthogonal basis vectors through a principal component analysis
 69 of the latent space, while the weak formulation is allowed to find non-orthogonal ones. Further,
 70 our results illustrate that both proposed formulations can interpolate efficiently whether between
 71 high-rank synthetic solutions, or between MNIST pictures while achieving both a lower latent space
 72 rank than the IRMAE and the LoRAE, and a lower computational overhead.

73 The present paper is structured as follows: Section 2 presents the architecture and both proposed
 74 formulations. Section 3 explains the insights behind long latent spaces with a low rank on two
 75 synthetic examples. Then, we compare the interpolation capabilities of RRAEs with IRMAEs,
 76 and LoRAEs on a variety of problems in section 4. We explain the limitations of the proposed
 77 formulations in section 5, before finally summarizing the main original contributions of the present
 78 paper in section 6.

79 2 Rank Reduction Autoencoders (RRAEs)

80 To define the architecture of RRAEs, we begin by defining autoencoder notations. Let $\{X_i\}_{i \in [1, D]} \in$
 81 \mathbb{R}^T be a set of D series of observations, each having T degrees of freedom. We define our input
 82 $X \in \mathbb{R}^{T \times D}$ with X_i as its i th column. Let, $Y \in \mathbb{R}^{L \times D}$, with L , the chosen dimension¹ of
 83 the latent space. We also define the encoding map $e : \mathbb{R}^{T \times D} \rightarrow \mathbb{R}^{L \times D}$ and the decoding map
 84 $d : \mathbb{R}^{L \times D} \rightarrow \mathbb{R}^{T \times D}$. The Vanilla autoencoder can be written as the following two operations,

$$Y = e(X), \quad \tilde{X} = d(Y). \quad (1)$$

85 In practice, we usually enforce that the output of the autoencoder gives us back the original data,
 86 hence the loss \mathcal{L} usually reads,

$$\mathcal{L}(X, \tilde{X}) = \|X - \tilde{X}\|_2, \quad \text{where, } \|\cdot\|_2 \text{ is the L2-norm.} \quad (2)$$

¹See Appendix B for details on the choice of L .

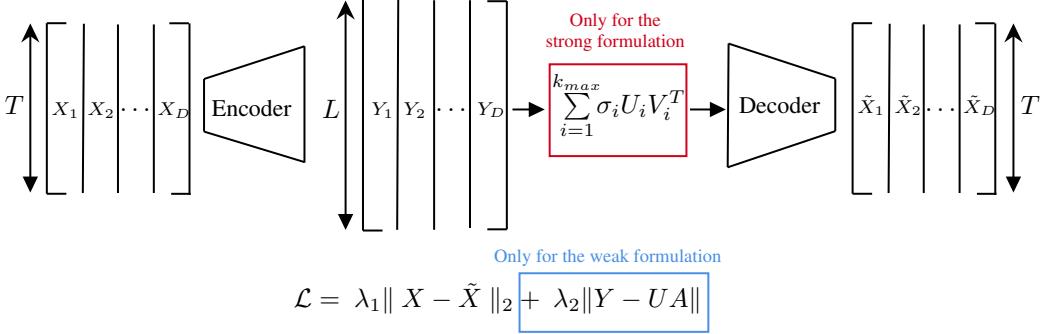


Figure 1: Schematic showing the autoencoder in use as well as both formulations. There are two terms in the loss function for the [Weak formulation](#). On the other hand, there's an additional step before the decoder for the [Strong formulation](#).

87 The idea behind RRAEs is to enforce the latent matrix to have a low rank while finding a reduced
 88 basis. In other words, if Y has a rank r , let $Y = U\Sigma V^T$ be the Singular Value Decomposition (SVD)
 89 [23, 16] of Y , with $U \in \mathbb{R}^{L \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, and $V^T \in \mathbb{R}^{r \times D}$. Let $\{\sigma_i\}_{i \in [1, r]}$ be the sorted diagonal
 90 values of Σ . Thus, by considering the k most significant modes (choice of k discussed in Section 4.2
 91 and Appendix B), it results,

$$Y = \sum_{i=1}^r \sigma_i U_i V_i^T \quad \Rightarrow \quad Y \approx \sum_{i=1}^k \sigma_i U_i V_i^T, \quad k \ll r, \quad (3)$$

92 where U_i is the i th column of U and V_i^T is the i th row of V^T . In other words, we can write Y_d , the
 93 d th column of Y as,

$$Y_d \approx \sum_{i=1}^k (\sigma_i U_i V_i^T)_d = \sum_{i=1}^k \sigma_i U_i V_{i,d}^T = \sum_{i=1}^k \alpha_{i,d} U_i, \quad \forall d \in [1, D], \quad (4)$$

94 with $V_{i,d}^T$ being entry d of vector V_i^T .

95 Accordingly, for k modes, each column of Y is defined by k coefficients and k vectors. Further,
 96 vectors U_i form a basis for the latent space. We can write (4) in matrix form as follows,

$$Y \approx UA, \quad \text{with: } A_{i,j} = \alpha_{i,j}, \quad U \in \mathbb{R}^{L \times k}, \quad A \in \mathbb{R}^{k \times D}, \quad (5)$$

97 Based on (4), and (5), we propose two formulations that enforce the low rank of the latent space and
 98 find its reduced basis. The architecture is sketched in Figure 1.

- 99 1. **The Weak formulation:** After choosing the maximum allowed number of modes k_{max} , we
 100 generate two trainable matrices $A \in \mathbb{R}^{k_{max} \times D}$, and $U \in \mathbb{R}^{L \times k_{max}}$. Afterward, we add
 101 a term to the loss as seen in blue in Figure 1. By doing so, minimizing the loss means
 102 that the latent space would have at most a rank of k_{max} . After convergence, the columns
 103 of our trainable matrix U form the reduced basis of the latent space. Additionally, the
 104 coefficients found in matrix A describe how to reconstruct each column Y_d as a linear
 105 combination of the basis vectors. We will refer to this method as the Weak formulation
 106 since throughout training, the Network minimizes a sum of both terms and not each term
 107 individually. Accordingly, predictions \tilde{X} could be less accurate, and we might end up with
 108 more modes than the specified value of k_{max} .

109 Remark: The two trainable matrices can be computed from a one-rank greedy procedure, as
 110 PGD performs.

- 111 2. **The Strong formulation:** Unlike the weak formulation, this architecture enforces, in a strong
 112 manner, the maximum dimension of the reduced basis of the latent space. Similarly to the
 113 weak formulation, we begin by choosing the maximum rank k_{max} of the latent space. Then,
 114 as seen in red in Figure 1, a truncated SVD (of order k_{max}) of the latent space is given to the
 115 decoder, instead of the latent space itself. Accordingly, the input of the decoder will have

116 at most k_{max} dominant singular values. We refer to this method as the Strong formulation
 117 since we strictly enforce the latent space to have a rank that's lower or equal to k_{max} . In
 118 this case, the basis vectors and coefficients are simply the ones found by the truncated SVD.

119 In both formulations, k_{max} is a hyperparameter to be chosen. We propose a strategy to choose this
 120 hyperparameter and discuss its effect in Section 4.2 and in Appendix B.

121 When using the strong formulation, we compute a POD basis, where the vectors are by construction
 122 orthogonal. The orthogonality of the basis vectors, as well as refraining from adding terms in the
 123 loss, can enhance both the training and interpolation results. On the other hand, backpropagation
 124 through the singular value decomposition is not common in practice. All the work presented in this
 125 paper was performed using equinox in JAX, where gradients of the singular value decomposition are
 126 implemented and accessible.

127 Both formulations reduce the limitations of IRMAE and LoRAE. We sum up our contributions as
 128 follows:

- 129 1. RRAEs with a strong formulation lead to low-rank latent spaces that have many singular
 130 values exactly equal to zero. In other words, the decoder will get a sum of exactly k_{max}
 131 rank-one updates. As will be shown later in the paper, this gives the strong formulation an
 132 advantage for training and interpolation.
- 133 2. The computational overhead of RRAEs is reduced compared to other architectures, especially
 134 for large latent spaces. For the strong formulation, when batches are used, the SVD is only
 135 performed on a matrix of size $L \times bs$, bs being the batch size. Similarly, for the weak
 136 formulation, the added computational cost is minimal since the trainable matrices are of
 137 shape $L \times k_{max}$ and $k_{max} \times D$ with $k_{max} \ll L$. On the other hand, the IRMAE or
 138 the LoRAE either performs a gradient descent or finds the nuclear norm of an $L \times L$
 139 matrix. Since a large latent space dimension L usually helps in achieving better results, both
 140 IRMAEs and LoRAEs can be computationally challenging.
- 141 3. Both formulations give us explicit control over the rank of the latent space. As shown
 142 next in the paper, we can enforce the latent space to have a lower rank than IRMAE and
 143 LoRAE, which leads to better interpolation and could help for feature extraction in future
 144 applications.

145 3 Insights behind Long latent spaces with low rank

146 An enlarged latent space can exhibit a linear behavior (as explored for instance in the Koopman
 147 theory, or the kPCA). Furthermore, a latent space with a reduced basis allows easier interpolation and
 148 feature extraction. The Diabolo Autoencoder on the other hand has “holes” in its interpolation [11],
 149 since it does not find a basis, but only a set of coefficients that are helpful for the decoder to retrieve
 150 the solution. Since the decoder is highly nonlinear, these coefficients can be anything, which leads to
 151 overfitting.

152 To illustrate the aforementioned arguments, we test DEAs and our Strong formulation on two examples
 153 characterized by one parameter. The first curves we propose are shifted sine curves since these have
 154 a simple nonlinearity, but they are hard to separate (nonmonotonic and cross each other multiple
 155 times). For our second example, we chose curves with stair-like behavior. In that case, we create
 156 highly nonlinear curves (different supports, different numbers of jumps of different magnitudes), but
 157 we define them to be monotonic and only cross each other occasionally (i.e. easier to separate). The
 158 equations used to define the columns of our input matrix X in each case are as follows,

$$\begin{cases} X_d(t_v, p_d) = f_{shift}(t_v, p_d) = \sin(t_v - p_d\pi), & p_d \in [0, 1.7], \\ X_d(t_v, p_d) = f_{stair}(t_v, p_d, \text{args}) & p_d \in [1, 5], \end{cases}$$

159 where $t_v \in \mathbb{R}^T$ is the time discretization vector, and f_{stair} takes some arguments “args” as detailed
 160 in the algorithm in Appendix A. The training is performed over 17, and 40 equidistant values of p_d
 161 for the shifted sine curves and the stair-like curves respectively. Later on, we interpolate the resulting
 162 curves in the latent space of the Autoencoders on 80 and 300 random values of p_d , respectively,
 163 chosen inside the training domain. The large number of tests guarantees that the models are learning

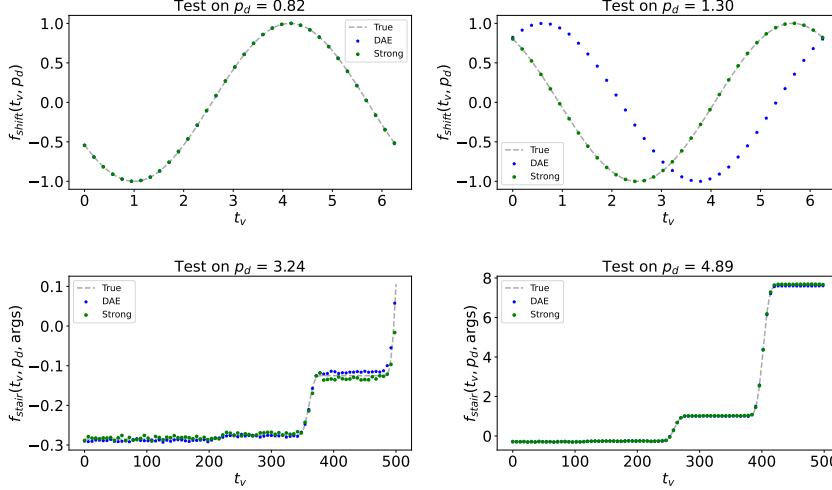


Figure 2: Predictions of DAEs and RRAEs with both formulations over two particular values of p_d for the shifted sine (above) and the stair-like examples (below).

164 the dynamics and not just the training curves and some tests nearby. Since the solution curves depend
 165 on one parameter, we use a DAE with a single scalar latent space and an RRAE with a longer latent
 166 space of rank one. We then linearly interpolate in the latent space to predict the test set. The training
 167 parameters, including the dimension of the latent space, can be found in Appendix B. The relative
 168 error over all p_d values for both the train and test sets is summarized in Table 1. Further, for each
 169 example, the predictions over some selected test cases are plotted in Figure 2.

Table 1: Relative error (in %) for all three architectures on both the train and test sets for both the examples of shifted sin curves and stair-like ones.

Model	Shifted sine		Stair-like	
	Train Error	Test error	Train Error	Test error
DAE	2.12	32.42	2.97	3.74
RRAE (strong)	1.73	1.90	1.87	3.2

170 The results show that when curves are hard to separate, RRAEs are better interpolators than DAEs.
 171 On the other hand, the effect of longer latent spaces is reduced for simple curves that can be highly
 172 nonlinear, but characterized by one parameter, and easily separable.

173 To further investigate the results, we plot the coefficients to be interpolated in the latent space as a
 174 function of the corresponding parameter p_d in Figure 3. It is important to note that the coefficients are
 175 defined differently between the RRAE and the DAE. For RRAEs, when $k_{max} = 1$, the coefficients
 176 are simply the entries of $A \in \mathbb{R}^D$ in equation (5). On the other hand, for a Diabolo Autoencoder with
 177 a scalar latent space, the values in the latent space themselves are the coefficients.

178 The main problem with the coefficients found by the DAE for the shifted sine curves (the blue crosses
 179 and dots in Figure 3 (left)) is that the resulting curve from linearly interpolating the coefficients
 180 is not an injection, over two significant parts of the domain. Specifically, for any value of p_d in
 181 approximately $[0, 0.3]$ and $[1.3, 1.5]$ (the dotted lines), there exists another value with the same
 182 coefficient α , leading to the same decoded curve. Accordingly, the decoder will find the same curve
 183 for two different parameters, which is wrong since p_d defines a shift. This explains why the DAE
 184 can interpolate well in the top left subplot in Figure 2, but not in the top right one. These results also
 185 show what is meant by “holes” in the latent space for DAEs.

186 On the other hand, as proposed earlier, a longer latent space allows us to find better features. This is
 187 clearly shown by the coefficients of the strong method in Figure 3 (left), which have a monotonic
 188 behavior.

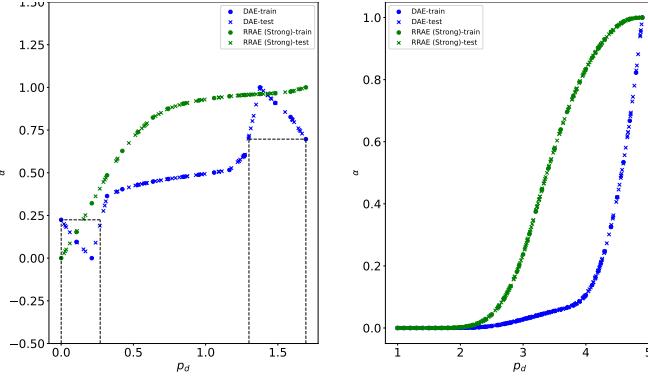


Figure 3: Normalized latent coefficients to be interpolated (dots) for DAE and RRAE with a strong formulation, and the interpolated values for the test set (crosses) for the shifted sine curves (left) and the stair-like curves (right).

Finally, the right part of Figure 3 depicts that when the curves are simple to separate and are characterized by only one parameter, both architectures can find monotonic coefficients that fit both the train and test sets.

4 Testing on Numerical Data

The solutions interpolated in the previous section were only characterized by one parameter. In this section, we test RRAEs and compare them to IRMAEs, and LoRAEs on two examples with a parametric space of dimension two, as well as on the MNIST dataset. Throughout the paper, we don't compare RRAEs with different variations of VAEs. Our future work will include a Variational version of RRAEs and its comparison to different VAE architectures.

4.1 Examples with two parameters

We generated two challenging synthetic tests for interpolation. First, we propose the sum of two sine curves with different frequencies, as well as two Gaussian bumps in two different locations. We show how in such examples both our formulations result in latent spaces with a lower rank and better results than IRMAEs and LoRAEs for the hyperparameters chosen (again, training details can be found in Appendix B). We define the columns of our input matrix $X_d(t_v, p_d) = f_{prob}$ for each problem as follows,

$$\begin{cases} f_{freqs}(t_v, \mathbf{p}_d) = \sin(p_d^1 \pi t_v) + \sin(p_d^2 \pi t_v), & p_d^1 \in [0.3, 0.5], \quad p_d^2 \in [0.8, 1], \\ f_{gauss}(t_v, \mathbf{p}_d) = 1.3e^{-\frac{(t_v - p_d^1)^2}{0.08}} + 1.3e^{-\frac{(t_v - p_d^2)^2}{0.08}}, & p_d^1 \in [1, 3], \quad p_d^2 \in [4, 6]. \end{cases}$$

We distinguish between the **bold** notation for vectors and non-bold ones for scalars. In both expressions, our parametric space is of dimension 2 and so $\mathbf{p}_d = [p_d^1, p_d^2] \in \mathbb{R}^2$. For each example and each architecture, we present some interpolated predictions in Figure 4, and the error over all the training/testing sets in Table 2 as well as the average training time for 100 batches.

As can be seen in Table 2, RRAEs with the Strong formulation are the most efficient in interpolation. Additionally, we note that increasing the parameter l for the IRMAE leads to divergence of the gradient descent (hence the N/A). Note that we only used the parameters specified in both papers for IRMAE and LoRAE. A fine-tuning of the parameters may potentially lead to better results for these architectures, but this venue is not investigated in this work. On the other hand, the table shows that RRAEs are faster than both IRMAEs and LoRAEs for the latent space dimension chosen. Our formulations are fast since we only add small matrices to the loss in the weak formulation, and we compute an SVD of an $L \times bs$ matrix, bs being the batch size, in the Strong formulation. On the other hand, IRMAEs and LoRAEs find the gradient/the nuclear norm of an $L \times L$ matrix respectively, with L relatively large.

Table 2: Relative error (in %) for all architectures on both the train and test with a latent space of dimension 2800 for the two examples presented, and the average time (in s) for 100 batches (size 20).

Model	Mult. Frequencies		Mult. Gausses		
	Train Error	Test error	Train Error	Test error	Average time
RRAE (strong)	6.33	12.83	4.46	8.75	1.61
RRAE (weak)	10.33	15.09	8.50	10.69	0.52
IRMAE ($l=2$)	6.95	17.35	4.68	13.93	3.6
IRMAE ($l=4$)	N/A	N/A	8.41	14.78	7.50
LoRAE	5.40	13.83	3.03	9.39	420.4

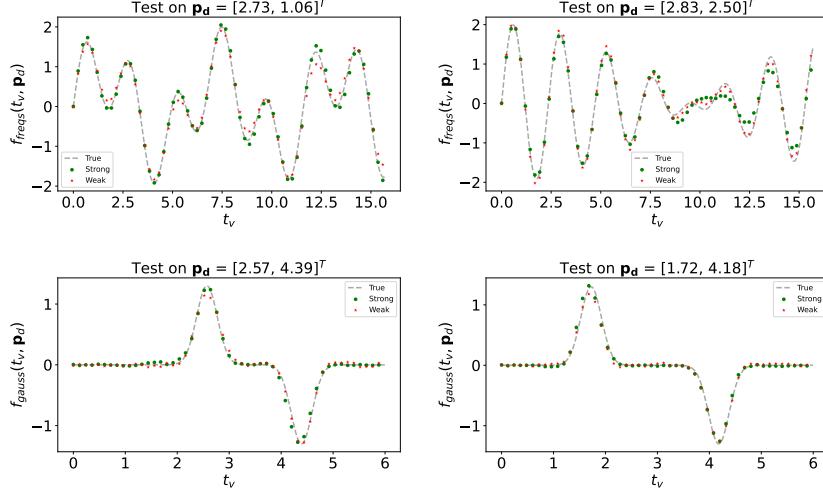


Figure 4: Interpolated results of RRAEs with both formulations on both examples presented with bilinear interpolation in the latent space.

219 Additionally, we draw parts of the normalized singular values of the latent space for the multiple
 220 gausses problem in Figure 5. The figure illustrates that adding the number of linear layers for
 221 the IRMAE (i.e. increasing l) indeed reduces the rank of the latent space. However, both of our
 222 formulations can be forced to find latent spaces with lower ranks. In addition, it is important to note
 223 that even though the LoRAE has a low error overall, the latent space rank is still relatively high
 224 compared to the other techniques (as can be seen from the slowly decreasing singular values in violet

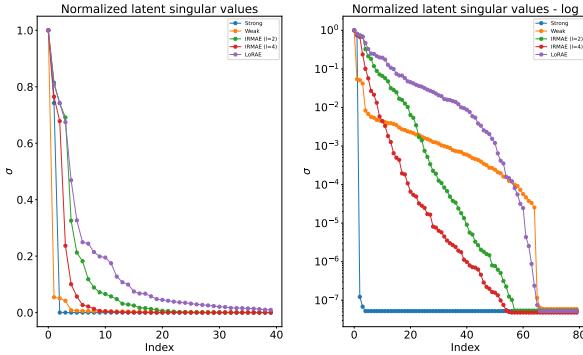


Figure 5: Normalized singular values of the latent space Y when trained over multiple gausses. The first 40 singular values are shown to the left, while many of the small singular values are shown on a log scale to the right.

225 in Figure 5 (left)). Hence, the trained model can not efficiently be used for other tasks, such as feature
226 extraction.

227 From the log-scaled graph illustrated in Figure 5 (right), we can understand why the strong formulation
228 achieves the best results. As previously mentioned, the IRMAE, the LoRAE and our weak formulation
229 don't enforce the singular values to fall to zero. So even though many singular values are small, they
230 still have a noise effect over the decoder, which reduces their efficiency in interpolation.

231 4.2 Testing on MNIST

232 In this section, we compare our architecture to the IRMAE and the LoRAE on the MNIST dataset.
233 First, each autoencoder is trained on all 60,000 pictures from the training test (training parameters
234 are available in Appendix B). Then, since the main application of the paper is interpolation, we use
235 each model to create interpolated pictures and form an “interpolated set”. Interpolation is done by
236 randomly choosing two pictures from the training set (e.g., leftmost and rightmost in Figure 6) and
237 linearly interpolating their latent variables to find five new pictures in between. Interpolated pictures
238 (in the red rectangle from left to right in Figure 6) are expected to transition from the leftmost picture
239 to the rightmost one in an equidistant manner. For instance, the pictures in the second column of
240 subplots in Figure 6 take 5/6 of the first picture (i.e. number 7) and 1/6 of the last picture (i.e. number
241 3). Similarly, pictures in the third column would have proportions of 4/6 and 2/6 respectively. This
242 procedure is then repeated 2000 times to create, for each architecture, an interpolated set of size
243 10,000. To quantify the quality of the generated images, we train a classifier on the original training
244 set and test it on the interpolated set generated by each architecture. Our classifier is a multilayer
245 perceptron with a softmax final activation function. It takes as input the latent space vector and
246 outputs a probability for every possible class (shape \mathbb{R}^{10}). Since the labels of the interpolated images
247 are unknown, we measure the success of classification by the certainty of the classifier, which we
248 quantify by the entropy of probability distribution written as follows,

$$H = \frac{-1}{10000} \sum_{i=1}^{10000} \sum_{j=1}^{10} p_j^i \log(p_j^i), \quad (6)$$

249 with p_j^i being the probability of class j found by the softmax activation function for sample i . More
250 details about the use of entropy can be found in Appendix E. The lower the entropy, the more certain
251 the classifier is about the class prediction of the interpolated image. We perform training using our

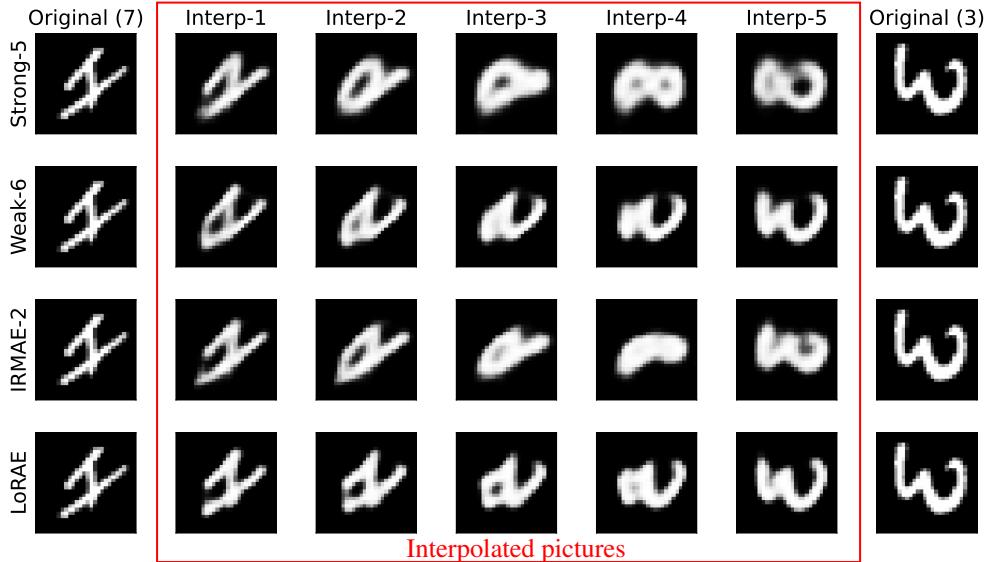


Figure 6: Five-step interpolation between a rotated number 7 (leftmost) and a number 3 (rightmost) on MNIST using RRAEs with both formulations, IRMAEs, and LoAREs.

252 Strong formulation with three choices of k_{max} to illustrate how the choice of this hyperparameter
 253 affects the model. We also generate five different interpolated tests for each architecture using
 254 different random training images for interpolation. The mean and standard deviation of the entropy
 255 are presented in Table 3.

Table 3: Mean entropy, standard deviation (over five different interpolation sets) and the rank of the latent space for the Strong formulation with three values of k_{max} , the weak formulation with $k_{max} = 6$, the IRMAE with $l = 2$, and the LoRAE, on their corresponding MNIST interpolated sets.

Model	Strong-5	Strong-8	Strong-12	Weak-6	IRMAE-2	LoRAE
H	$0.50 \pm 6e-3$	$0.45 \pm 4e-3$	$0.49 \pm 4e-3$	$0.51 \pm 5e-3$	$0.50 \pm 6e-3$	$0.46 \pm 5e-3$
Rank	5	8	12	6	8	30

256 The table illustrates how the choice of k_{max} can be made. We propose to start with a small value and
 257 increase it until the error stagnates or increases again. In this case, $k_{max} = 8$ is the best choice, but
 258 the Strong formulation can interpolate the MNIST pictures even when restricted to only 5 features.
 259 Both Table 3 and Figure 6 illustrate how both our formulations can interpolate well between MNIST
 260 pictures. The strong formulation, for instance, recognizes that it is hard to go from 7 to 3 and goes
 261 through 6 (interp-2) and 8 (interp-3,4,5). Further, while the LoRAE has a low error, its latent space
 262 has 30 dominant singular values, which doesn't allow any feature extraction. On the other hand,
 263 compared to IRMAEs in Table 3, the noise from the smaller singular values as well as the explicit
 264 control over the rank allows us to get either a smaller entropy for the same rank (Strong-8) or almost
 265 the same entropy with fewer features to extract (Strong-5 and Weak-6).

266 5 Limitations

267 As illustrated in the paper's results, RRAEs with both formulations can interpolate well while using a
 268 latent space with a low rank. However, our proposed model has some limitations:

- 269 Even though both of our formulations allow explicit control over the rank of the latent space,
 270 k_{max} is a hyperparameter to be tuned. In practice, starting with a small value of k_{max} and
 271 increasing it until error convergence is a good strategy. In general, other techniques (e.g.,
 272 PCA) could be used to approximate the intrinsic dimension of the latent space "à priori".
- 273 The weak formulation adds regularisation constants to the loss, which can be hard to tune.
 274 In practice, we had to repeat training multiple times to tune the parameters, which isn't ideal,
 275 especially for a larger dataset such as the MNIST.
- 276 For high dimensional problems with very long latent spaces, the strong formulation can be
 277 computationally expensive. Even though the SVD is only performed on a matrix of size
 278 $L \times bs$, bs being the batch size and L the length of the latent space, the cost of computing
 279 an SVD and backpropagating through it when L is excessively large can be high.
- 280 The effect of a long latent space is reduced when the solution is simple and separable. In
 281 such cases, an increased dimension of the latent space, hence RRAEs, may not be necessary.

282 6 Summary and Conclusions

283 In this article, we presented Rank Reduction Autoencoders (RRAEs), Autoencoders with latent spaces
 284 that accept linear reduction. We proposed two formulations, a weak and a strong one to find the latent
 285 space while building its reduced basis. Even though the basis vectors in the strong formulation are
 286 orthogonal, and they need not be in the weak formulation, we showed that both formulations can
 287 interpolate correctly between curves. Overall, our results show that the Strong formulation has a
 288 superior capability of interpolation since it doesn't have any noise from small nonzero singular values
 289 in the latent space. We also showed that both the Strong and the Weak formulations can achieve
 290 lower ranks in the latent space while being able to efficiently interpolate vector functions. Finally,
 291 both formulations are fast to train, with the weak formulation being the fastest. While the Strong
 292 formulation leads to better predictions, the Weak formulation is much simpler to implement since it
 293 only adds a penalty term to the loss.

294 **References**

- 295 [1] David Amsallem and Charbel Farhat. Interpolation method for adapting reduced-order models
296 and application to aeroelasticity. *Aiaa Journal - AIAA J*, 46:1803–1813, 07 2008. doi: 10.2514/
297 1.35374.
- 298 [2] Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *Machine learning for data*
299 *science handbook: data mining and knowledge discovery handbook*, pages 353–374, 2023.
- 300 [3] Joshua L Barnett, Charbel Farhat, and Yvon Maday. Neural-network-augmented projection-
301 based model order reduction for mitigating the kolmogorov barrier to reducibility of cfd models,
302 2022.
- 303 [4] Sarath Chandar A P, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran,
304 Vikas C Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word
305 representations. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger,
306 editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates,
307 Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/2bcab9d935d219641434683dd9d18a03-Paper.pdf.
- 308 [5] Yingyi Chen, Qinghua Tao, Francesco Tonin, and Johan Suykens. Primal-attention: Self-
309 attention through asymmetric kernel svd in primal representation. In A. Oh, T. Nau-
310 mann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neu-*
311 *ral Information Processing Systems*, volume 36, pages 65088–65101. Curran Associates,
312 Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/cd687a58a13b673eea3fc1b2e4944cf7-Paper-Conference.pdf.
- 313 [6] Francisco Chinesta and Elias Cueto. *PGD-based modeling of materials, structures and processes*.
314 Springer, Switzerland, 2014.
- 315 [7] Francisco Chinesta, Pierre Ladeveze, and Elias Cueto. A short review on model order reduction
316 based on proper generalized decomposition. *Archives of Computational Methods in Engineering*,
317 18(4):395–404, 2011.
- 318 [8] Eizaburo Doi and Michael Lewicki. Sparse coding of natural images using an over-
319 complete set of limited capacity units. In L. Saul, Y. Weiss, and L. Bottou, ed-
320 itors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press,
321 2004. URL https://proceedings.neurips.cc/paper_files/paper/2004/file/309a8e73b2cdb95fc1affa8845504e87-Paper.pdf.
- 322 [9] David González, José Vicente Aguado, E Cueto, E Abisset-Chavanne, and F Chinesta. kPCA-
323 based parametric solutions within the pgd framework. *Archives of Computational Methods in*
324 *Engineering*, 25:69–86, 2018.
- 325 [10] Guillaume Huguet, Daniel Sumner Magruder, Alexander Tong, Oluwadamilola Fasina, Manik
326 Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold interpolating optimal-transport flows
327 for trajectory inference. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh,
328 editors, *Advances in Neural Information Processing Systems*, volume 35, pages 29705–29718.
329 Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/bfc03f077688d8885c0a9389d77616d0-Paper-Conference.pdf.
- 330 [11] Li Jing, Jure Zbontar, and yann lecun. Implicit rank-minimizing autoencoder. In
331 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neu-*
332 *ral Information Processing Systems*, volume 33, pages 14736–14746. Curran Associates,
333 Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/a9078e8653368c9c291ae2f8b74012e7-Paper.pdf.
- 334 [12] Gaetan Kerschen, Jean-claude Golinval, Alexander F Vakakis, and Lawrence A Bergman. The
335 method of proper orthogonal decomposition for dynamical characterization and order reduction
336 of mechanical systems: an overview. *Nonlinear dynamics*, 41:147–169, 2005.
- 337 [13] Tianxiao Li, Hongyu Guo, Filippo Grazioli, Mark Gerstein, and Martin Renqiang Min.
338 Disentangled wasserstein autoencoder for t-cell receptor engineering. In A. Oh, T. Nau-
339 mann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neu-*
340 *ral Information Processing Systems*, volume 36, pages 73604–73632. Curran Associates,
341 Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/e95da8078ec8389533c802e368da5298-Paper-Conference.pdf.
- 342
- 343
- 344
- 345
- 346
- 347
- 348
- 349
- 350
- 351
- 352
- 353
- 354
- 355
- 356
- 357
- 358
- 359
- 360
- 361
- 362
- 363
- 364
- 365
- 366
- 367
- 368
- 369
- 370
- 371
- 372
- 373
- 374
- 375
- 376
- 377
- 378
- 379
- 380
- 381
- 382
- 383
- 384
- 385
- 386
- 387
- 388
- 389
- 390
- 391
- 392
- 393
- 394
- 395
- 396
- 397
- 398
- 399
- 400
- 401
- 402
- 403
- 404
- 405
- 406
- 407
- 408
- 409
- 410
- 411
- 412
- 413
- 414
- 415
- 416
- 417
- 418
- 419
- 420
- 421
- 422
- 423
- 424
- 425
- 426
- 427
- 428
- 429
- 430
- 431
- 432
- 433
- 434
- 435
- 436
- 437
- 438
- 439
- 440
- 441
- 442
- 443
- 444
- 445
- 446
- 447
- 448
- 449
- 450
- 451
- 452
- 453
- 454
- 455
- 456
- 457
- 458
- 459
- 460
- 461
- 462
- 463
- 464
- 465
- 466
- 467
- 468
- 469
- 470
- 471
- 472
- 473
- 474
- 475
- 476
- 477
- 478
- 479
- 480
- 481
- 482
- 483
- 484
- 485
- 486
- 487
- 488
- 489
- 490
- 491
- 492
- 493
- 494
- 495
- 496
- 497
- 498
- 499
- 500
- 501
- 502
- 503
- 504
- 505
- 506
- 507
- 508
- 509
- 510
- 511
- 512
- 513
- 514
- 515
- 516
- 517
- 518
- 519
- 520
- 521
- 522
- 523
- 524
- 525
- 526
- 527
- 528
- 529
- 530
- 531
- 532
- 533
- 534
- 535
- 536
- 537
- 538
- 539
- 540
- 541
- 542
- 543
- 544
- 545
- 546
- 547
- 548
- 549
- 550
- 551
- 552
- 553
- 554
- 555
- 556
- 557
- 558
- 559
- 560
- 561
- 562
- 563
- 564
- 565
- 566
- 567
- 568
- 569
- 570
- 571
- 572
- 573
- 574
- 575
- 576
- 577
- 578
- 579
- 580
- 581
- 582
- 583
- 584
- 585
- 586
- 587
- 588
- 589
- 590
- 591
- 592
- 593
- 594
- 595
- 596
- 597
- 598
- 599
- 600
- 601
- 602
- 603
- 604
- 605
- 606
- 607
- 608
- 609
- 610
- 611
- 612
- 613
- 614
- 615
- 616
- 617
- 618
- 619
- 620
- 621
- 622
- 623
- 624
- 625
- 626
- 627
- 628
- 629
- 630
- 631
- 632
- 633
- 634
- 635
- 636
- 637
- 638
- 639
- 640
- 641
- 642
- 643
- 644
- 645
- 646
- 647
- 648
- 649
- 650
- 651
- 652
- 653
- 654
- 655
- 656
- 657
- 658
- 659
- 660
- 661
- 662
- 663
- 664
- 665
- 666
- 667
- 668
- 669
- 670
- 671
- 672
- 673
- 674
- 675
- 676
- 677
- 678
- 679
- 680
- 681
- 682
- 683
- 684
- 685
- 686
- 687
- 688
- 689
- 690
- 691
- 692
- 693
- 694
- 695
- 696
- 697
- 698
- 699
- 700
- 701
- 702
- 703
- 704
- 705
- 706
- 707
- 708
- 709
- 710
- 711
- 712
- 713
- 714
- 715
- 716
- 717
- 718
- 719
- 720
- 721
- 722
- 723
- 724
- 725
- 726
- 727
- 728
- 729
- 730
- 731
- 732
- 733
- 734
- 735
- 736
- 737
- 738
- 739
- 740
- 741
- 742
- 743
- 744
- 745
- 746
- 747
- 748
- 749
- 750
- 751
- 752
- 753
- 754
- 755
- 756
- 757
- 758
- 759
- 760
- 761
- 762
- 763
- 764
- 765
- 766
- 767
- 768
- 769
- 770
- 771
- 772
- 773
- 774
- 775
- 776
- 777
- 778
- 779
- 780
- 781
- 782
- 783
- 784
- 785
- 786
- 787
- 788
- 789
- 790
- 791
- 792
- 793
- 794
- 795
- 796
- 797
- 798
- 799
- 800
- 801
- 802
- 803
- 804
- 805
- 806
- 807
- 808
- 809
- 810
- 811
- 812
- 813
- 814
- 815
- 816
- 817
- 818
- 819
- 820
- 821
- 822
- 823
- 824
- 825
- 826
- 827
- 828
- 829
- 830
- 831
- 832
- 833
- 834
- 835
- 836
- 837
- 838
- 839
- 840
- 841
- 842
- 843
- 844
- 845
- 846
- 847
- 848
- 849
- 850
- 851
- 852
- 853
- 854
- 855
- 856
- 857
- 858
- 859
- 860
- 861
- 862
- 863
- 864
- 865
- 866
- 867
- 868
- 869
- 870
- 871
- 872
- 873
- 874
- 875
- 876
- 877
- 878
- 879
- 880
- 881
- 882
- 883
- 884
- 885
- 886
- 887
- 888
- 889
- 890
- 891
- 892
- 893
- 894
- 895
- 896
- 897
- 898
- 899
- 900
- 901
- 902
- 903
- 904
- 905
- 906
- 907
- 908
- 909
- 910
- 911
- 912
- 913
- 914
- 915
- 916
- 917
- 918
- 919
- 920
- 921
- 922
- 923
- 924
- 925
- 926
- 927
- 928
- 929
- 930
- 931
- 932
- 933
- 934
- 935
- 936
- 937
- 938
- 939
- 940
- 941
- 942
- 943
- 944
- 945
- 946
- 947
- 948
- 949
- 950
- 951
- 952
- 953
- 954
- 955
- 956
- 957
- 958
- 959
- 960
- 961
- 962
- 963
- 964
- 965
- 966
- 967
- 968
- 969
- 970
- 971
- 972
- 973
- 974
- 975
- 976
- 977
- 978
- 979
- 980
- 981
- 982
- 983
- 984
- 985
- 986
- 987
- 988
- 989
- 990
- 991
- 992
- 993
- 994
- 995
- 996
- 997
- 998
- 999
- 1000

- 348 [14] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery
 349 of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and*
 350 *machine intelligence*, 35(1):171–184, 2012.
- 351 [15] Alokendu Mazumder, Tirthajit Baruah, Bhartendu Kumar, Rishab Sharma, Vishwajeet Pattanaik,
 352 and Punit Rathore. Learning low-rank latent spaces with simple deterministic autoencoder:
 353 Theoretical and empirical insights, 2023.
- 354 [16] Yuji Nakatsukasa. Accuracy of singular vectors obtained by projection-based svd methods. *BIT*
 355 *Numerical Mathematics*, 57(4):1137–1152, 2017.
- 356 [17] Minh-Nhan Nguyen and Hyun-Gyu Kim. An efficient podi method for real-time simulation of
 357 indenter contact problems using rbf interpolation and contact domain decomposition. *Computer*
 358 *Methods in Applied Mechanics and Engineering*, 388:114215, 2022.
- 359 [18] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-
 360 assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation*
 361 *Letters*, 3(3):1544–1551, 2018.
- 362 [19] RR Rama and S Skatulla. Towards real-time modelling of passive and active behaviour of the
 363 human heart using podi-based model reduction. *Computers & Structures*, 232:105897, 2020.
- 364 [20] Marc' aurelio Ranzato, Y-lan Boureau, and Yann Cun. Sparse feature learning for
 365 deep belief networks. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors,
 366 *Advances in Neural Information Processing Systems*, volume 20. Curran Associates,
 367 Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/c60d060b946d6dd6145dcbad5c4ccf6f-Paper.pdf.
- 369 [21] S Rodriguez, David Néron, P-E Charbonnel, Pierre Ladevèze, and G Nahas. Non incremental
 370 latin-pgd solver for non-linear vibratory dynamics problems. In *14ème Colloque National en*
 371 *Calcul des Structures, CSMA 2019*, 2019.
- 372 [22] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the*
 373 *20th international conference on machine learning (ICML-03)*, pages 720–727, 2003.
- 374 [23] Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM review*, 35
 375 (4):551–566, 1993.
- 376 [24] Marco Tezzele, Nicola Demo, and Gianluigi Rozza. Shape optimization through proper
 377 orthogonal decomposition with interpolation and dynamic mode decomposition enhanced by
 378 active subspaces, 2019.
- 379 [25] Sergio Torregrosa, Victor Champaney, Amine Ammar, Vincent Herbert, and Francisco Chinesta.
 380 Hybrid twins based on optimal transport. *Computers & Mathematics with Applications*, 127:
 381 12–24, 2022. ISSN 0898-1221. doi: <https://doi.org/10.1016/j.camwa.2022.09.026>. URL
 382 <https://www.sciencedirect.com/science/article/pii/S0898122122004060>.
- 383 [26] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation
 384 learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and
 385 R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran
 386 Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.
- 388 [27] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and
 389 composing robust features with denoising autoencoders. In *Proceedings of the 25th international*
 390 *conference on Machine learning*, pages 1096–1103, 2008.

391 **NeurIPS Paper Checklist**

392 **1. Claims**

393 Question: Do the main claims made in the abstract and introduction accurately reflect the
394 paper's contributions and scope?

395 Answer: [Yes]

396 Justification: We include mainly three claims (zero singular values, computational time, and
397 explicit control over the rank), which are all tackled later in the paper.

398 Guidelines:

- 399 • The answer NA means that the abstract and introduction do not include the claims
400 made in the paper.
- 401 • The abstract and/or introduction should clearly state the claims made, including the
402 contributions made in the paper and important assumptions and limitations. A No or
403 NA answer to this question will not be perceived well by the reviewers.
- 404 • The claims made should match theoretical and experimental results, and reflect how
405 much the results can be expected to generalize to other settings.
- 406 • It is fine to include aspirational goals as motivation as long as it is clear that these goals
407 are not attained by the paper.

408 **2. Limitations**

409 Question: Does the paper discuss the limitations of the work performed by the authors?

410 Answer: [Yes]

411 Justification: Section 5 basically discusses the main limitations of our proposed model. We
412 also discussed how Diabolo Autoencoders can achieve similar results to RRAEs in certain
413 training conditions in Section 3.

414 Guidelines:

- 415 • The answer NA means that the paper has no limitation while the answer No means that
416 the paper has limitations, but those are not discussed in the paper.
- 417 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 418 • The paper should point out any strong assumptions and how robust the results are to
419 violations of these assumptions (e.g., independence assumptions, noiseless settings,
420 model well-specification, asymptotic approximations only holding locally). The authors
421 should reflect on how these assumptions might be violated in practice and what the
422 implications would be.
- 423 • The authors should reflect on the scope of the claims made, e.g., if the approach was
424 only tested on a few datasets or with a few runs. In general, empirical results often
425 depend on implicit assumptions, which should be articulated.
- 426 • The authors should reflect on the factors that influence the performance of the approach.
427 For example, a facial recognition algorithm may perform poorly when image resolution
428 is low or images are taken in low lighting. Or a speech-to-text system might not be
429 used reliably to provide closed captions for online lectures because it fails to handle
430 technical jargon.
- 431 • The authors should discuss the computational efficiency of the proposed algorithms
432 and how they scale with dataset size.
- 433 • If applicable, the authors should discuss possible limitations of their approach to
434 address problems of privacy and fairness.
- 435 • While the authors might fear that complete honesty about limitations might be used by
436 reviewers as grounds for rejection, a worse outcome might be that reviewers discover
437 limitations that aren't acknowledged in the paper. The authors should use their best
438 judgment and recognize that individual actions in favor of transparency play an impor-
439 tant role in developing norms that preserve the integrity of the community. Reviewers
440 will be specifically instructed to not penalize honesty concerning limitations.

441 **3. Theory Assumptions and Proofs**

442 Question: For each theoretical result, does the paper provide the full set of assumptions and
443 a complete (and correct) proof?

444 Answer: [NA]

445 Justification: The paper does not include theoretical results.

446 Guidelines:

- 447 • The answer NA means that the paper does not include theoretical results.
- 448 • All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- 449 • All assumptions should be clearly stated or referenced in the statement of any theorems.
- 450 • The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- 451 • Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- 452 • Theorems and Lemmas that the proof relies upon should be properly referenced.

453 4. Experimental Result Reproducibility

454 Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

455 Answer: [Yes]

456 Justification: We include all the necessary details for creating the synthetic datasets throughout the article. We also detail all the hyperparameters used for training in Appendix B.

457 Guidelines:

- 458 • The answer NA means that the paper does not include experiments.
- 459 • If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- 460 • If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- 461 • Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- 462 • While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - 463 (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - 464 (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - 465 (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - 466 (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

496 5. Open access to data and code

497 Question: Does the paper provide open access to the data and code, with sufficient instruc-
498 tions to faithfully reproduce the main experimental results, as described in supplemental
499 material?

500 Answer: [No]

501 Justification: While we can not share the code used, we provide all the necessary details for
502 generating the data and performing training throughout the article and in the appendices.

503 Guidelines:

- 504 • The answer NA means that paper does not include experiments requiring code.
- 505 • Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- 506 • While we encourage the release of code and data, we understand that this might not be
507 possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not
508 including code, unless this is central to the contribution (e.g., for a new open-source
509 benchmark).
- 510 • The instructions should contain the exact command and environment needed to run to
511 reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- 512 • The authors should provide instructions on data access and preparation, including how
513 to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- 514 • The authors should provide scripts to reproduce all experimental results for the new
515 proposed method and baselines. If only a subset of experiments are reproducible, they
516 should state which ones are omitted from the script and why.
- 517 • At submission time, to preserve anonymity, the authors should release anonymized
518 versions (if applicable).
- 519 • Providing as much information as possible in supplemental material (appended to the
520 paper) is recommended, but including URLs to data and code is permitted.

523 6. Experimental Setting/Details

524 Question: Does the paper specify all the training and test details (e.g., data splits, hyper-
525 parameters, how they were chosen, type of optimizer, etc.) necessary to understand the
526 results?

527 Answer: [Yes]

528 Justification: We provide the training parameters for each example in the corresponding
529 section. All the parameters, including JAX random keys to exactly reproduce the test sets
530 can be found in Appendix B.

531 Guidelines:

- 532 • The answer NA means that the paper does not include experiments.
- 533 • The experimental setting should be presented in the core of the paper to a level of detail
534 that is necessary to appreciate the results and make sense of them.
- 535 • The full details can be provided either with the code, in appendix, or as supplemental
536 material.

537 7. Experiment Statistical Significance

538 Question: Does the paper report error bars suitably and correctly defined or other appropriate
539 information about the statistical significance of the experiments?

540 Answer: [Yes]

541 Justification: On the MNIST dataset, we repeated experiments five times with different
542 interpolation sets and we documented both the mean and the standard deviation of the results
543 in Table 3 (with an explanation of how they were computed).

544 Guidelines:

- 545 • The answer NA means that the paper does not include experiments.
- 546 • The authors should answer “Yes” if the results are accompanied by error bars, confi-
547 dence intervals, or statistical significance tests, at least for the experiments that support
548 the main claims of the paper.

- 549 • The factors of variability that the error bars are capturing should be clearly stated (for
 550 example, train/test split, initialization, random drawing of some parameter, or overall
 551 run with given experimental conditions).
 552 • The method for calculating the error bars should be explained (closed form formula,
 553 call to a library function, bootstrap, etc.)
 554 • The assumptions made should be given (e.g., Normally distributed errors).
 555 • It should be clear whether the error bar is the standard deviation or the standard error
 556 of the mean.
 557 • It is OK to report 1-sigma error bars, but one should state it. The authors should
 558 preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis
 559 of Normality of errors is not verified.
 560 • For asymmetric distributions, the authors should be careful not to show in tables or
 561 figures symmetric error bars that would yield results that are out of range (e.g. negative
 562 error rates).
 563 • If error bars are reported in tables or plots, The authors should explain in the text how
 564 they were calculated and reference the corresponding figures or tables in the text.

565 **8. Experiments Compute Resources**

566 Question: For each experiment, does the paper provide sufficient information on the com-
 567 puter resources (type of compute workers, memory, time of execution) needed to reproduce
 568 the experiments?

569 Answer: [Yes]

570 Justification: All of the details related to the PC specifications were mentioned in Appendix
 571 B.

572 Guidelines:

- 573 • The answer NA means that the paper does not include experiments.
 574 • The paper should indicate the type of compute workers CPU or GPU, internal cluster,
 575 or cloud provider, including relevant memory and storage.
 576 • The paper should provide the amount of compute required for each of the individual
 577 experimental runs as well as estimate the total compute.
 578 • The paper should disclose whether the full research project required more compute
 579 than the experiments reported in the paper (e.g., preliminary or failed experiments that
 580 didn't make it into the paper).

581 **9. Code Of Ethics**

582 Question: Does the research conducted in the paper conform, in every respect, with the
 583 NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

584 Answer: [Yes]

585 Justification: The code of ethics has been thoroughly read by the corresponding author who
 586 made sure that the research conducted in the paper conforms to it. The format of the paper
 587 is anonymous (following the guidelines of the template provided).

588 Guidelines:

- 589 • The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
 590 • If the authors answer No, they should explain the special circumstances that require a
 591 deviation from the Code of Ethics.
 592 • The authors should make sure to preserve anonymity (e.g., if there is a special consid-
 593 eration due to laws or regulations in their jurisdiction).

594 **10. Broader Impacts**

595 Question: Does the paper discuss both potential positive societal impacts and negative
 596 societal impacts of the work performed?

597 Answer: [No]

598 Justification: Since we are working on foundational research with no particular applications.
 599 We believe our topic is too broad to have any specific societal impacts.

600 Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

623 **11. Safeguards**

624 Question: Does the paper describe safeguards that have been put in place for responsible
625 release of data or models that have a high risk for misuse (e.g., pretrained language models,
626 image generators, or scraped datasets)?

627 Answer: [NA]

628 Justification: The paper poses no such risks.

629 Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

640 **12. Licenses for existing assets**

641 Question: Are the creators or original owners of assets (e.g., code, data, models), used in
642 the paper, properly credited and are the license and terms of use explicitly mentioned and
643 properly respected?

644 Answer: [Yes]

645 Justification: We are the owners of the model presented and we credited the libraries and
646 languages that were used for the code.

647 Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- 653 • For scraped data from a particular source (e.g., website), the copyright and terms of
 654 service of that source should be provided.
 655 • If assets are released, the license, copyright information, and terms of use in the
 656 package should be provided. For popular datasets, paperswithcode.com/datasets
 657 has curated licenses for some datasets. Their licensing guide can help determine the
 658 license of a dataset.
 659 • For existing datasets that are re-packaged, both the original license and the license of
 660 the derived asset (if it has changed) should be provided.
 661 • If this information is not available online, the authors are encouraged to reach out to
 662 the asset's creators.

663 **13. New Assets**

664 Question: Are new assets introduced in the paper well documented and is the documentation
 665 provided alongside the assets?

666 Answer: [Yes]

667 Justification: The model we propose with both formulations is detailed as a concept in
 668 Section 2, our appendices include more details on how to use the model with both our
 669 proposed formulations.

670 Guidelines:

- 671 • The answer NA means that the paper does not release new assets.
 672 • Researchers should communicate the details of the dataset/code/model as part of their
 673 submissions via structured templates. This includes details about training, license,
 674 limitations, etc.
 675 • The paper should discuss whether and how consent was obtained from people whose
 676 asset is used.
 677 • At submission time, remember to anonymize your assets (if applicable). You can either
 678 create an anonymized URL or include an anonymized zip file.

679 **14. Crowdsourcing and Research with Human Subjects**

680 Question: For crowdsourcing experiments and research with human subjects, does the paper
 681 include the full text of instructions given to participants and screenshots, if applicable, as
 682 well as details about compensation (if any)?

683 Answer: [NA]

684 Justification: The paper does not involve crowdsourcing or research with human subjects.

685 Guidelines:

- 686 • The answer NA means that the paper does not involve crowdsourcing nor research with
 687 human subjects.
 688 • Including this information in the supplemental material is fine, but if the main contribu-
 689 tion of the paper involves human subjects, then as much detail as possible should be
 690 included in the main paper.
 691 • According to the NeurIPS Code of Ethics, workers involved in data collection, curation,
 692 or other labor should be paid at least the minimum wage in the country of the data
 693 collector.

694 **15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human
 695 Subjects**

696 Question: Does the paper describe potential risks incurred by study participants, whether
 697 such risks were disclosed to the subjects, and whether Institutional Review Board (IRB)
 698 approvals (or an equivalent approval/review based on the requirements of your country or
 699 institution) were obtained?

700 Answer: [NA]

701 Justification: The paper does not involve crowdsourcing or research with human subjects.

702 Guidelines:

- 703 • The answer NA means that the paper does not involve crowdsourcing nor research with
 704 human subjects.

- 705 • Depending on the country in which research is conducted, IRB approval (or equivalent)
706 may be required for any human subjects research. If you obtained IRB approval, you
707 should clearly state this in the paper.
708 • We recognize that the procedures for this may vary significantly between institutions
709 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
710 guidelines for their institution.
711 • For initial submissions, do not include any information that would break anonymity (if
712 applicable), such as the institution conducting the review.

713 **Appendix A Algorithm for the stair-like function**

Algorithm 1: Algorithm to find f_{stair} for a parameter p_d .

Input: $p_d \in \mathbb{R}$, $t_v \in \mathbb{R}^T$, $(\text{Ph}_0, \text{Amp}_0, \kappa, y_0, w) \in \mathbb{R}$

$$\text{Amp}_{p_d} = p_d$$

$$\text{Ph}_{p_d} = \text{Ph}_0 + \kappa(\text{Amp}_{p_d} - \text{Amp}_0)$$

714 $g_{p_d}(t_v) = \text{Amp}_{p_d} \sqrt{t_v} \sin(w(t_v - \text{Ph}_{p_d})) - y_0$

$$h_{p_d}(t) = \left(\frac{|g_{p_d}(t)| + g_{p_d}(t)}{2} \right)^5$$

$$X_d(t_v, p_d) = \text{cumsum}(h_{p_d}(t_v))$$

Output: $X_d(t_v, p_d)$ for each parameter p_d .

715 In this paper, we choose the initial parameters of the stair function to be,

$$\begin{cases} \text{Ph}_0 = 0.875, & \text{Amp}_0 = 1 \\ \kappa = 2.286, & y_0 = 2.3, \\ & w = 2\pi. \end{cases}$$

716 **Appendix B Training details**

717 **B.1 Hyperparameters for synthetic data and PC properties**

718 The main purpose of this section is to allow readers to reproduce the results, and share the parameters
 719 we used to train the models. In general, the main parameters for RRAEs are the dimension of the
 720 latent space, k_{max} , the encoder/decoder architectures, learning rates, epochs, and batch sizes. For
 721 each problem, we fix the common parameters between all architectures. We try to only change
 722 necessary parameters between different examples, to show that training RRAEs, especially with
 723 the strong formulation, doesn't require too much hyperparameters tuning. For all problems and all
 724 formulations, we use an encoder of depth 1 and width 64, and a decoder of depth 6 and width 64.
 725 Additionally, we use batches of size 20, the softplus as activation function between all layers, and
 726 the adabeleif optimizer, for all problems and formulations. Furthermore, we use multiple learning
 727 rates, starting with $1e-3$ and dividing by 10 until reaching $1e-5$ (3 steps). In each step, we train for
 728 2000 batches. however, we impose stagnation criteria which usually stop training earlier. Further, we
 729 normalize the data by subtracting the mean and dividing it by the standard deviation. We found that
 730 normalization was necessary, especially for the stair-like functions.

731 Throughout the paper, the only two parameters that we vary for RRAEs are the length of the latent
 732 space L and a coefficient κ_w that changes the learning rates for the trainable matrices of the weak
 733 method. In practice, we found that changing the learning rate of the trainable matrix A for the weak
 734 formulation is easier than changing the weights in the loss. Accordingly, while we use the same
 735 learning rate strategy proposed before for the encoder/decoder, we propose to multiply the learning
 736 rate by a constant κ_w before applying it to the trainable matrix A . By doing so, we find that there is
 737 no need to tune the loss parameters (i.e. both are equal to one). It is important to note that the vectors
 738 in the trainable matrix U are normalized at every training step so A captures the coefficients. In Table
 739 4, we illustrate the latent space dimension L and the constant κ_w used for all the illustrated examples
 740 in this work (N/A means the weak method was not used for this example).

Table 4: Different values of the latent space length L and the constant κ_w that are used for all the examples in this work (except MNIST).

Param.	Shifts	Stair-like	Mult. Freqs.	Mult. Gauss.
L	4500	4500	2800	2800
κ_w	N/A	N/A	0.66	0.13

741 Next, we detail how the choice of k_{max} was made for each example. In general, an approximation of
 742 k_{max} can be found using multiple techniques (e.g. PCA). However, in this paper, we chose a simpler
 743 approach. If this hyperparameter is too small, the model will not converge, but if it is too large, the
 744 Neural Network will simply learn a latent space of a higher rank. Accordingly, we started with a
 745 small value of k_{max} and increased it until the error converged. The values chosen for each example
 746 are detailed in Table 5.

Table 5: Different values of k_{max} that are used for all the examples in this work (except MNIST).

Param.	Shifts	Stair-like	Mult. Freqs.	Mult. Gauss.
k_{max}	1	1	12	2

747 As can be seen in the table, while we were able to choose exactly the dimension of the parametric
 748 space for most of the examples, for the sine curves with different frequencies, the method needed a
 749 higher rank in the latent space to converge to the low errors presented. This is mainly because the
 750 latent space was not long enough for the problem. However, we tried to fix the parameter L , so we
 751 had to change k_{max} accordingly. On the other hand, we chose the parameters that were shown to give
 752 the best results for the IRMAE and the LoRAE. We tried to have two and four linear layers for the
 753 IRMAE (i.e. $l = 2$ and $l = 4$), and we used a weight of 0.001 in the loss for the LoRAE (the optimal
 754 value specified in the presenting paper). Other parameter values for LoRAE and IRMAE were tested
 755 with little improvements overall. However, a fine-tuned choice of parameters could lead to better
 756 results than the ones presented in the paper. To ensure a fair comparison, every other parameter of
 757 these models was chosen to be the same as RRAEs (the ones listed before).

758 Since we provide average computational times in Section 4.1, we give some details about the machine
 759 used to generate these results. The PC used is an MSI Stealth 17Studio A13VH. The processor is
 760 Intel (13th generation), Core i9, 2600 MHz, 14 CPUs. The PC also has 64 GB of RAM. The library
 761 used was equinox, in JAX for the training. However, the code was only run on CPUs.

762 B.2 Hyperparameters for MNIST

763 The architecture for the MNIST dataset was different since convolutional Neural Networks were used.
 764 We fixed the kernel size to 4, the padding to 1, and the strid to 2 for each convolution/convolution
 765 transpose. For the encoder, we used convolutions with output 32, 64, 128, and 256 respectively, with
 766 relu activation functions in between. These were followed by a flattening layer, and a Multilayer
 767 Perceptron (MLP) with softplus activation functions, of depth 2, and width 64. The output of the
 768 MLP was fixed to be 128, the dimension of the latent space. On the other hand, the decoder included
 769 an MLP with depth 2, width 64, and softplus activation functions with an output dimension of 1568.
 770 The vector was then reshaped into a tensor of shape (32, 7, 7), which was then followed by two
 771 transposed convolutions with output shapes 8, and 1 respectively. For training, we used a learning
 772 rate of 0.0001, the optimizer adabeleif, and a total of 50 epochs.

773 Even though the architecture and training are probably not the best ones, our purpose was to show
 774 that for a fixed architecture, RRAEs can outperform other existing methods and not achieve SOTA
 775 results over the MNIST since this was done for many architectures before.

776 While the choice of the hyperparameter k_{max} and l (for IRMAE) has been mentioned in Section 4.2,
 777 we used again the optimal weight in the loss proposed in the paper of LoRAE as $\lambda = 0.001$. We also
 778 used a factor $\kappa_w = 0.8$ for the weak formulation of RRAEs.

779 Now, we detail how our interpolation sets were created. As previously mentioned in the paper, we
 780 choose two random figures from the training test and generate five pictures by interpolating the latent
 781 space. This procedure is done 2000 time to generate the interpolation set. The entire thing is done
 782 5 times to provide statistically significant results (i.e. with a mean and a standard deviation). For
 783 readers who would like to reproduce our results, we used the seeds 0, 10, 100, 1000, and 10000 to
 784 generate a `jax.random.keys` respectively. The key was then split into 2000 other keys (by using
 785 `jax.random.split`). Finally, we fed these keys to create 2000 permutations of the indices of the
 786 training figures (i.e. 0 until 60,000) and only took the first two numbers as our choice of the figures
 787 to be interpolated. By using these seeds, readers should be able to exactly generate the interpolation
 788 sets used in the paper. In addition, the example presented in Figure 6 is the interpolation between
 789 pictures of indices 58, 300 and 12.

790 **B.3 Choice of parameters**

791 Throughout the paper, we mentioned the range in which the values of \mathbf{p}_d were chosen for each
 792 example. In this subsection, we provide some details on the chosen values of \mathbf{p}_d , mainly to show
 793 that the test covers most of the parametric space. Throughout the paper, we presented curves with
 794 parametric spaces of one and two. The following figures show the plot of the second parameter
 795 against the first one when the space is of dimension two (Figure 8). On the other hand, when the curve
 796 is only characterized by one parameter, we plot the vector of the parameter against itself (Figure 7).
 797 Hence, we plot dots on a diagonal line to show where the test values lie compared to the train values.
 798 Our test set was chosen randomly but using a JAX seed to ensure reproducibility. As can be seen in
 799 the figures, we carefully chose the seeds and the number of tests to represent most of the parametric
 800 space.

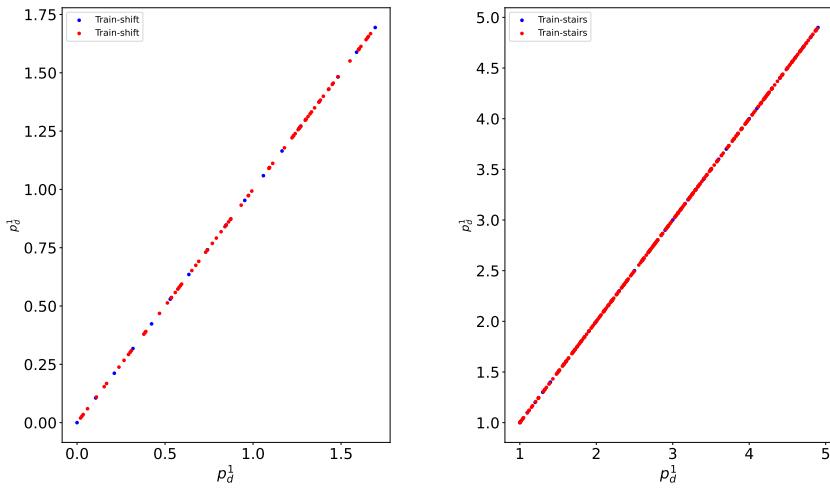


Figure 7: Train and test parameter values for the example with two shifted sine curves (left), and stair-like curves (right).

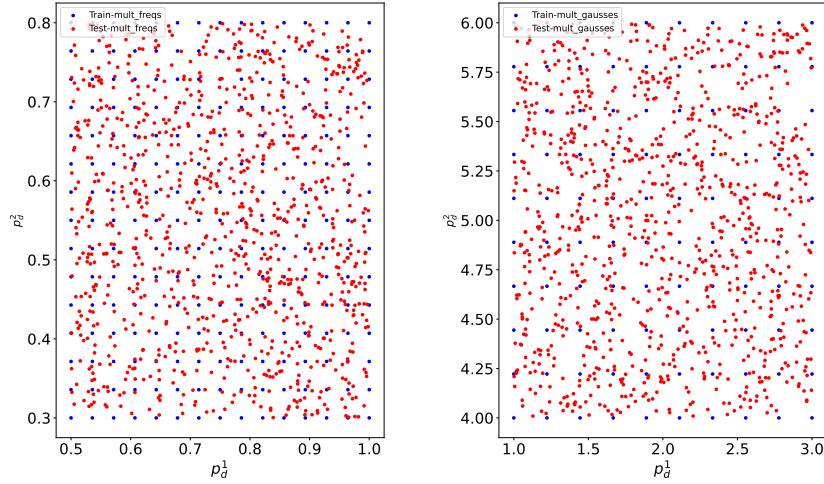


Figure 8: Train and test parameter values for the example with two accelerated sine curves (left), and two Gausses (right).

801 We now explain how to exactly reproduce the test set we had. Each random parameter was created
 802 using `jax.random.uniform` which takes a `jax.random.key(seed)` as its first parameter. By
 803 choosing the same seed, the random numbers are guaranteed to be the same. Accordingly, we provide

804 the seeds used to generate the test sets for each synthetic example in the paper. These can be found in
 805 Table 6.

Table 6: Chosen `jax.random` seeds for generating test parameters.

Problem	Shifted sine	Stairs	Mult. Freqs	Mult. Gausses
Parameter	p_d	p_d	p_d^1	p_d^2
Seed	0	0	140	8

806 Appendix C Comparing RRAEs to AEs with long latent dimensions

807 Throughout the paper, we only compared our proposed models with others that allow feature ex-
 808 traction, since it is of interest to us. However, Vanilla Autoencoders with long latent spaces can
 809 interpolate very well! On both the multiple frequencies and the multiple gausses, the comparison
 810 between our Strong formulation and a Vanilla Autoencoder with a latent space of dimension 2800
 811 (same length) is shown in table 7.

Table 7: Error (in %) on our two synthetic problems for our Strong formulation and an AE with the same latent dimension with no rank restriction.

Model	Mult. Freqs		Mult Gausses	
	Train Error	Test error	Train Error	Test error
AE (long)	7.96	14.08	3.37	10.56
RRAE (strong)	6.33	12.83	4.46	8.75

812 As can be seen in the table, reducing the rank not only allows feature extraction, it can also help in
 813 training to achieve better results and interpolation.

814 Appendix D Batching

815 In this section, we detail how batching was performed for both the Weak and the Strong formulations.

816 The weak formulation: We remind the reader that the weak formulation had the norm of $Y - UA$ in
 817 the loss, with $Y \in \mathbb{R}^{L \times D}$, $U \in \mathbb{R}^{L \times k_{max}}$, and $A \in \mathbb{R}^{k_{max} \times D}$. However, when training over batches
 818 of size bs , we have a batched latent space $Y^b \in \mathbb{R}^{L \times bs}$ and so the shape of A needs to be different.
 819 Accordingly, for each batch, we keep the indices of the vector functions used and take the column of
 820 the same indices from A to form $A^b \in \mathbb{R}^{k_{max} \times bs}$. Accordingly, for each forward/backward pass, we
 821 train different columns of matrix A .

822 The strong formulation: For the strong formulation, nothing changes. The truncated SVD is per-
 823 formed over the batched latent space $Y^b = U\Sigma V^T$. It is important to note though that since the
 824 columns change depending on the batch, the values of the right singular vector (i.e. V^T) fluctuate a
 825 lot in training. However, since the same vector U is used for all the batches, the RRAE converges
 826 towards the right basis vectors in U . After training, to be sure that training is performed over the
 827 whole dataset, we perform the truncated SVD over the entire latent space (i.e. without batching) to
 828 get the corresponding reduced basis and coefficients that are used for interpolation (i.e. the ones
 829 found in training are disregarded). Our findings are that the RRAE converges towards a unique U ,
 830 which is why interpolation is so successful throughout the paper.

831 Appendix E Entropy as a measure of uncertainty

832 In section 4.2, we used the entropy to measure the uncertainty of the model. Other measures such
 833 as the Fréchet inception distance (FID) and the inception score (IS) are conventionally used/trained
 834 for colored pictures. Accordingly, we chose to use a similar concept to evaluate the generation of
 835 gray pictures (MNIST numbers). Using the entropy was based on IS. In general, IS is a scalar that

836 gives an idea of how good the generated pictures are by evaluating the diversity and the quality of
837 the generated pictures. For the IS, the entropy is a measure of the quality of the generated pictures.
838 However, the entropy is computed for the pre-trained v3 model (on colored pictures). Accordingly,
839 we created our own classifiers (for each architecture, by using a binary cross-entropy loss, an MLP of
840 depth 2, width 4, the softplus activation function for the first layer, and a softmax activation in the
841 end), which predicted, from the latent space as input, the probability of being in each class (an output
842 of shape \mathbb{R}^{10}) for each architecture, on which we then computed the entropies. By the equation of
843 the entropy which multiplies the probability of each class by the logarithm of that probability, an
844 ideal prediction would simply be 1 for a class and 0 for every other class (hence 100% certainty).
845 The further the predicted values of our classifier are from 0 and 1, the harder it is to classify the
846 pictures in the interpolated test, which means that the generated pictures don't necessarily resemble
847 the original training set. However, we don't evaluate the diversity of the generated pictures, since our
848 interpolation process is between two pre-specified pictures (contrary to generative models where they
849 could generate anything).