



MOX-Report No. 74/2026

Deep Symmetric Autoencoders from the Eckart-Young-Schmidt Perspective

Brivio, S.; Franco, N. R.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it

<https://mox.polimi.it>

DEEP SYMMETRIC AUTOENCODERS FROM THE ECKART-YOUNG-SCHMIDT PERSPECTIVE*

Simone Brivio[†] and Nicola Rares Franco[†]

[†]MOX, Department of Mathematics, Politecnico di Milano, Milan, Italy

`{simone.brivio, nicolarares.franco}@polimi.it`

ABSTRACT. Deep autoencoders have become a fundamental tool in various machine learning applications, ranging from dimensionality reduction and reduced order modeling of partial differential equations to anomaly detection and neural machine translation. Despite their empirical success, a solid theoretical foundation for their expressiveness remains elusive, particularly when compared to classical projection-based techniques. In this work, we aim to take a step forward in this direction by presenting a comprehensive analysis of what we refer to as symmetric autoencoders, a broad class of deep learning architectures that have surfaced frequently in the recent literature. Specifically, we introduce a formal distinction between different classes of symmetric architectures, analyzing their strengths and limitations from a mathematical perspective. For instance, we show that the reconstruction error of symmetric autoencoders with orthonormality constraints can be understood by leveraging the well-renowned Eckart-Young-Schmidt (EYS) theorem. As a byproduct of our analysis, we end up developing the EYS initialization strategy for symmetric autoencoders, which is based on an iterated application of the Singular Value Decomposition (SVD). To validate our findings, we conduct a series of numerical experiments where we benchmark our proposal against conventional deep autoencoders, discussing the importance of model design and initialization.

KEYWORDS: autoencoders, error bounds, low-rank, Eckart-Young-Schmidt
MSC: 68T07, 47N40

1. Introduction

Originally developed between the '80s and '90s as a nonlinear alternative to Principal Component Analysis (PCA) [32, 21, 20, 17], deep autoencoders have now gained traction in several fields, ranging from statistical learning to scientific machine learning. They are currently employed for tasks involving dimensionality reduction [16], anomaly detection [6], image processing [4], neural machine translation [7], generative modeling [19], reduced order modeling of partial differential equations (PDEs) [24, 12], and more.

The widespread adoption of autoencoders can be attributed to their ability to deliver competitive performance compared to traditional linear techniques —such as PCA, Proper Orthogonal Decomposition (POD), and the Kosambi-Karhunen-Loève (KKL) [5, 23]— particularly when modeling complex phenomena that exhibit nonlinear behavior. However, our theoretical understanding of deep autoencoders is still far from being exhaustive. Compared to linear methods, which are very well-understood, autoencoders remain sort of a black-box approach. For this reason, domain practitioners are often brought to rely upon empirical rules of thumb when it comes to their design, implementation and training. In this work, we aim to take a step forward in this direction by presenting a comprehensive analysis focused over a particular class of architectures which we refer to as *symmetric autoencoders*.

*Paper accepted for publication in the *SIAM Journal on Scientific Computing* (SISC) on July 7, 2026.

Existing literature. In the literature of deep autoencoders, symmetric architectures have emerged consistently during the years, with different researchers independently converging towards this design paradigm in an attempt to enhance model interpretability and reliability. For instance, in a recent work, Otto et al. [26] have proposed the use of symmetric autoencoders with biorthogonality constraints for the study of dynamical systems. Conversely, in [1], a symmetric denoising autoencoder is proposed for the imputation of missing data. Autoencoder networks with a symmetric-like structure are also commonly found in the field of reduced order modeling of parametrized PDEs, see, e.g. [24, 29, 13].

Despite this convergence, however, most of these works have been carried out independently, typically motivated by different applications and spanning across different research domains. In turn, no clear systematic investigation has been carried out yet, either theoretically or experimentally.

Our contribution. Within this work, we provide a mathematically rigorous introduction to symmetric autoencoders. We first discuss the general idea, with a particular focus on the choice of the activation function. Then, we identify two specialized classes of architectures—symmetric autoencoders with biorthogonality and orthogonality constraints, respectively. In doing so, we discuss the inherent mathematical properties of these architectures, outlining their strengths and limitations. In particular, by drawing a parallel with linear reduction methods, we are able to derive novel error bounds connecting symmetric autoencoders to the well-known Singular Value Decomposition (SVD). Then, by leveraging the Eckart-Young-Schmidt theorem, we design a new initialization strategy, specifically tailored for symmetric autoencoder networks, which proves superior to classical initialization routines in terms of first-epoch loss, oftentimes benefiting training dynamics and convergence.

Work organization. The paper is organized as follows. First, in Section 2, we introduce the some notation and recall fundamental results in low-rank approximation theory that are essential to our analysis. Then, in Section 3, we formally introduce and characterize deep symmetric autoencoders, along with a suitable theoretical framework to analyze their properties and capabilities, deriving the aforementioned error bounds. The practical implications of our findings, instead, are discussed in Section 4, where we present our new initialization strategy. Finally, in Section 5, we complement our analysis through an extensive set of numerical experiments, whereas, in the last section we draw some conclusions and indicate promising directions for future research.

2. Mathematical backbone and notation

Before coming to our main topic, which is the study of deep symmetric autoencoders, we provide a synthetic overview of the fundamental ingredients necessary for our analysis. We start by introducing some notation and recalling certain classical results concerning linear reduction, which will form the foundation of our approach. In doing so, we shall highlight the importance of the celebrated low-rank approximation theorem named after Eckart and Young [11]. Actually, hereby we present its extension to compact operators on Hilbert spaces—a result originally discovered by Schmidt back in 1907 during his studies on integral operators [33].

2.1. Notation

For clarity in our presentation, we introduce some notation concerning standard concepts in probability theory (such as covariance operators and Bochner-integrable random variables) and operator theory (including Hilbert-Schmidt operators, adjoints, and matrices).

Probability setting. Let $(\mathcal{P}, \mathcal{A}, \mathbb{P})$ be a probability space, consisting of an sample space $\mathcal{P}$, a σ -field $\mathcal{A}$, and a probability measure $\mathbb{P}$. Given a measurable map $f : \mathcal{P} \rightarrow [0, +\infty)$, we set $\mathbb{E}[f] := \int_{\mathcal{P}} f(\boldsymbol{\mu}) \mathbb{P}(d\boldsymbol{\mu})$. For any given $q \in [1, +\infty)$, we denote by $L^q(\mathcal{P})$ the space of measurable maps $g : \mathcal{P} \rightarrow \mathbb{R}$ with finite q th-moment, i.e., $\mathbb{E}|g|^q < +\infty$. For all such maps, we define $\mathbb{E}[g] := \int_{\mathcal{P}} g(\boldsymbol{\mu}) \mathbb{P}(d\boldsymbol{\mu})$, as usual [10].

For any $n_0 \in \mathbb{N}_+$, we extend the above notation to the case of vector-valued maps $\mathbf{u} : \mathcal{P} \rightarrow \mathbb{R}^{n_0}$ using Bochner integration. Precisely, we write $L^q(\mathcal{P}; \mathbb{R}^{n_0})$ for the space of Bochner q -integrable maps, i.e. the collection of all measurable maps $\mathbf{u} : \mathcal{P} \rightarrow \mathbb{R}^{n_0}$ such that $\mathbb{E}\|\mathbf{u}\|^q < +\infty$, with $\|\cdot\|$ the Euclidean norm. For all such maps, we set

$$\mathbb{E}[\mathbf{u}] := [\mathbb{E}[u_1], \dots, \mathbb{E}[u_{n_0}]]^\top \in \mathbb{R}^{n_0},$$

with $u_j(\boldsymbol{\mu}) := \mathbf{e}_j^\top \mathbf{u}(\boldsymbol{\mu})$ the j th output of $\mathbf{u}$ and $\{\mathbf{e}_j\}_{j=1}^{n_0}$ the canonical basis of $\mathbb{R}^{n_0}$.

We recall that, up to quotient operations, $L^2(\mathcal{P})$ and $L^2(\mathcal{P}; \mathbb{R}^{n_0})$ are Hilbert spaces once equipped with the inner products $\langle f, g \rangle_{L^2(\mathcal{P})} := \mathbb{E}[fg]$ and $\langle \mathbf{u}, \mathbf{v} \rangle_{L^2(\mathcal{P}; \mathbb{R}^{n_0})} := \mathbb{E}[\mathbf{u}^\top \mathbf{v}]$, respectively. Lastly, for all $\mathbf{u} \in L^2(\mathcal{P}; \mathbb{R}^{n_0})$ we denote by

$$\boldsymbol{\Sigma}[\mathbf{u}] := \mathbb{E}[(\mathbf{u} - \mathbb{E}[\mathbf{u}])(\mathbf{u} - \mathbb{E}[\mathbf{u}])^\top] = (\mathbb{E}[(u_i - \mathbb{E}[u_i])(u_j - \mathbb{E}[u_j])])_{i,j=1}^{n_0} \in \mathbb{R}^{n_0 \times n_0}$$

the covariance matrix of the random vector $\mathbf{u}$.

Hilbert-Schmidt operators. We write $\mathcal{H}(H_1, H_2)$ for the space of Hilbert-Schmidt operators from H_1 to H_2 , each H_i being a separable Hilbert space. We recall that the latter is defined as the collection of all continuous linear operators $T : H_1 \rightarrow H_2$ such that $\|T\|_{\text{HS}}^2 := \sum_{i=1}^{+\infty} \|Te_i\|_{H_2}^2 < +\infty$, where $\{e_i\}_i$ is an orthonormal basis of H_1 . The quantity $\|T\|_{\text{HS}}$ is independent on the choice of such basis and it defines a norm over $\mathcal{H}(H_1, H_2)$ [8]. For all such operators, we write $\text{rk}(T) := \dim(T(H_1))$, to denote their *rank*, whereas we use T^* for their adjoint operator, i.e., $T^* : H_2 \rightarrow H_1$ such that $\langle Tg, v \rangle_{H_2} = \langle g, T^*v \rangle_{H_1}$ for all $g \in H_1$ and $v \in H_2$. We recall that, if H_1 and H_2 are both finite-dimensional then T and T^* can be represented by a matrix and its transpose.

Matrices. We write $\mathcal{O}_{m,n} := \{\mathbf{A} \in \mathbb{R}^{m \times n} : \mathbf{A}^\top \mathbf{A} = \mathbf{I}_n\}$ for the set of $m \times n$ orthonormal matrices. Given a positive-semidefinite matrix $\mathbf{M} \in \mathbb{R}^{m \times m}$, we write $\lambda_i(\mathbf{M})$ for its i th eigenvalue, listed in decreasing order and counting multiplicities. Finally, given any matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ we write $\|\mathbf{B}\|_{\text{HS}}$ and $\|\mathbf{B}\|_{\text{op}}$ for its Frobenius and operator norms, respectively.

2.2. Fundamental results in linear reduction

Given a random vector $\mathbf{u}$ taking values in $\mathbb{R}^{n_0}$, the purpose of linear reduction techniques is to find an approximate representation of $\mathbf{u}$ by introducing a carefully chosen affine subspace $V \subset \mathbb{R}^{n_0}$, of dimension $\dim(V) = n \ll n_0$, designed such that the discrepancy $\mathbb{E}[\inf_{\mathbf{v} \in V} \|\mathbf{u} - \mathbf{v}\|^2]$ is minimized. Algebraically, this boils to down finding a suitable matrix $\mathbf{V} \in \mathcal{O}_{n_0,n}$ and a suitable vector $\mathbf{q} \in \mathbb{R}^{n_0}$ such that the reconstruction error

$$\mathbb{E} \|\mathbf{u} - [\mathbf{V}\mathbf{V}^\top(\mathbf{u} - \mathbf{q}) + \mathbf{q}]\|^2, \quad (1)$$

is as small as possible.

The main reason behind such construction is that, once $\mathbf{V}$ and $\mathbf{q}$ have been identified, one can use $\mathbf{c} := \mathbf{V}^\top(\mathbf{u} - \mathbf{q})$, an n -dimensional random vector, as a proxy for $\mathbf{u}$. In fact, $\mathbf{V}$ and $\mathbf{q}$ naturally define an encoding and a decoding strategy,

$$\mathbb{R}^{n_0} \ni \mathbf{u} \mapsto \mathbf{V}^\top(\mathbf{u} - \mathbf{q}) \in \mathbb{R}^n \quad \text{and} \quad \mathbb{R}^n \ni \mathbf{c} \mapsto \mathbf{V}\mathbf{c} + \mathbf{q} \in \mathbb{R}^{n_0}, \quad (2)$$

respectively. Both maps are affine and 1-Lipschitz continuous, two properties that contribute to the stability and interpretability of linear reduction methods.

What is most remarkable, however, is that the solution to Eq. (1) is known in closed form. In fact, the latter can be computed using an algorithm that is commonly referred to as Principal Orthogonal Decomposition (POD). At its core, POD relies on the fundamental Eckart-Young-Schmidt theorem, which we state below.

Theorem 2.1 (Eckart-Young-Schmidt). *Let $(H_1, \|\cdot\|_1)$ and $(H_2, \|\cdot\|_2)$ be two separable Hilbert spaces. Let $T \in \mathcal{H}(H_1, H_2)$. There exists a vanishing nonincreasing sequence $s_1 \geq s_2 \geq \dots \geq 0$, an orthonormal basis $\{\xi_i\}_i$ of H_1 and an orthonormal basis $\{v_i\}_i$ of H_2 such that*

$$T = \sum_{i=1}^{\text{rk}(T)} s_i v_i \langle \xi_i, \cdot \rangle_{H_1}. \quad (3)$$

Furthermore, for all $1 \leq n < \text{rk}(T)$, the truncated operator $T_n := \sum_{i=1}^n s_i v_i \langle \xi_i, \cdot \rangle_{H_1}$ is the best n -rank approximation of T and satisfies

$$\|T - T_n\|_{\text{HS}}^2 = \min \{ \|T - L\|_{\text{HS}}^2 : \text{rk}(L) \leq n \} = \sum_{i=n+1}^{\text{rk}(T)} s_i^2,$$

Additionally, s_i^2 and v_i are the eigenvalues and eigenvectors of $TT^* : H_2 \rightarrow H_2$, whereas s_i^2 and ξ_i are the eigenvalues and eigenvectors of $T^*T : H_1 \rightarrow H_1$.

Proof. See, e.g., [31, Theorem VI.17]. □

For a given operator T , the scalars $\{s_i\}_i$ are typically referred to as *singular values*. Similarly, $\{v_i\}_i$ and $\{\xi_i\}_i$ are known as left and right singular vectors, respectively. For this reason, the representation formula in Eq. (3) is often called *singular values decomposition* (SVD), but is also known as *low-rank decomposition*, *polar form* or *Schmidt decomposition*. The connection between (1) and the Eckart-Young-Schmidt theorem lies in the following Proposition, which, in light of its simplicity, we present without proof.

Proposition 2.1. *Given any $\mathbf{u} \in L^2(\mathcal{P}; \mathbb{R}^{n_0})$, define $T_{\mathbf{u}} : L^2(\mathcal{P}) \rightarrow \mathbb{R}^{n_0}$ as*

$$T_{\mathbf{u}} g := \mathbb{E}[\mathbf{u}g].$$

The map $\mathbf{u} \rightarrow T_{\mathbf{u}}$ is a surjective linear isometry from $L^2(\mathcal{P}; \mathbb{R}^{n_0})$ to $\mathcal{H}(L^2(\mathcal{P}), \mathbb{R}^{n_0})$.

Essentially, Proposition 2.1 can be understood as generalization of Riesz's representation theorem to vector-valued functionals, and it can be used to represent Bochner square-integrable random variables as Hilbert-Schmidt operators. By combining the above with the Eckart-Young-Schmidt theorem, it is straightforward to prove the following result, which is the cornerstone underlying the POD algorithm. Even though the proof is classical and can be found in many textbooks, see, e.g., [30, Proposition 6.3] or [22, Theorem 3.8], the notation is often quite different. For this reason, we provide the interested reader with a revisitation of the proof in the Appendix. As a side note, we emphasize that the isometry in Proposition 2.1 is what motivates stating Theorem 2.1 in the general context of separable Hilbert spaces. In fact, although the ambient space, $\mathbb{R}^{n_0}$, is finite-dimensional, the introduction of a probability measure naturally implies an additional layer of complexity, which makes it necessary to discuss low-rank approximation in more general spaces.

Theorem 2.2 (Optimality of POD). *Let $\mathbf{u} \in L^2(\mathcal{P}; \mathbb{R}^{n_0})$. Set $r := \text{rk}(T_{\mathbf{u}})$ and fix any positive integer $n < r \leq n_0$. Consider a decomposition of $\Sigma[\mathbf{u}]$ in diagonal form given as*

$$\Sigma[\mathbf{u}] = \mathbf{X} \mathbf{D} \mathbf{X}^\top,$$

where $\mathbf{D} := \text{diag}(\lambda_1(\boldsymbol{\Sigma}[\mathbf{u}]), \dots, \lambda_r(\boldsymbol{\Sigma}[\mathbf{u}]))$ and $\mathbf{X} \in \mathcal{O}_{n_0, r}$. Define $\mathbf{V}_* \in \mathbb{R}^{n_0 \times n}$ by extracting the first n columns of $\mathbf{X}$ and let $\mathbf{q}_* := \mathbb{E}[\mathbf{u}]$. Then,

$$\begin{aligned} \mathbb{E} \|\mathbf{u} - [\mathbf{V}_* \mathbf{V}_*^\top (\mathbf{u} - \mathbf{q}_*) + \mathbf{q}_*]\|^2 &= \\ &= \min_{\substack{\mathbf{V} \in \mathcal{O}_{n_0, n} \\ \mathbf{q} \in \mathbb{R}^{n_0}}} \mathbb{E} \|\mathbf{u} - [\mathbf{V} \mathbf{V}^\top (\mathbf{u} - \mathbf{q}) + \mathbf{q}]\|^2 = \sum_{i=n+1}^r \lambda_i(\boldsymbol{\Sigma}[\mathbf{u}]). \end{aligned} \quad (4)$$

Proof. See Appendix A. □

Summary and desiderata. The above result, combined with the rich structure provided by linear projections, highlights the following key points:

- (i) the problem of linear reduction can be solved in closed form by the POD algorithm, which involves diagonalizing the covariance matrix $\boldsymbol{\Sigma}[\mathbf{u}]$;
- (ii) the accuracy of the approximation is driven by the decay of the eigenvalues of $\boldsymbol{\Sigma}[\mathbf{u}]$, which ultimately relates to the Eckart-Young-Schmidt theorem;
- (iii) projection methods, including POD, are representation consistent, meaning that they map $\mathbf{u}$ and its reconstruction $\tilde{\mathbf{u}} := \mathbf{V} \mathbf{V}^\top (\mathbf{u} - \mathbf{q}) + \mathbf{q}$ onto the same latent representation. Indeed, being $\mathbf{c} = \mathbf{V}^\top (\mathbf{u} - \mathbf{q})$, one has

$$\tilde{\mathbf{c}} = \mathbf{V}^\top \tilde{\mathbf{u}} = \mathbf{V}^\top [\mathbf{V} \mathbf{V}^\top (\mathbf{u} - \mathbf{q}) + \mathbf{q} - \mathbf{q}] = \mathbf{V}^\top \mathbf{V} \mathbf{V}^\top (\mathbf{u} - \mathbf{q}) = \mathbf{V}^\top (\mathbf{u} - \mathbf{q}) = \mathbf{c};$$

- (iv) POD, and linear projection methods in general, are grounded on a stable decoding mechanism, $\mathbf{c} \mapsto \mathbf{V} \mathbf{c}$. In fact,

$$\|\mathbf{V} \mathbf{c} - \mathbf{V} \mathbf{c}'\| = \|\mathbf{c} - \mathbf{c}'\|,$$

the two norms being respectively in $\mathbb{R}^{n_0}$ and $\mathbb{R}^n$.

Our purpose for the remainder of this work, is to bring similar insights in the context of nonlinear reduction by means of deep symmetric autoencoders.

3. Symmetric autoencoders

Let $n_0 > n_1 \geq \dots \geq n_l > 0$ be positive integers. A symmetric autoencoder with input dimension n_0 , hidden dimensions $n_1, \dots, n_{l-1}$ and *latent dimension* n_l , is a sequence of tuples of the form

$$\boldsymbol{\Psi} = \{(\mathbf{E}_j, \mathbf{D}_j, \mathbf{e}_j, \mathbf{d}_j, \rho)\}_{j=1}^l,$$

where:

- $\rho : \mathbb{R} \rightarrow \mathbb{R}$ is a given bilipschitz activation (cf. Lemma 3.1),
- $\mathbf{E}_j \in \mathbb{R}^{n_j \times n_{j-1}}$ and $\mathbf{D}_j \in \mathbb{R}^{n_{j-1} \times n_j}$ are suitable weight matrices,
- $\mathbf{e}_j \in \mathbb{R}^{n_j}$ and $\mathbf{d}_j \in \mathbb{R}^{n_{j-1}}$ are given bias vectors.

Each symmetric autoencoder defines an encoding procedure $\mathcal{E}(\boldsymbol{\Psi}, \cdot) : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_l}$ via the iterative scheme

$$\mathcal{E}(\boldsymbol{\Psi}, \mathbf{u}) := (e_l \circ \dots \circ e_1)(\mathbf{u}),$$

where $e_j : \mathbf{h} \mapsto \rho(\mathbf{E}_j \mathbf{h} + \mathbf{e}_j)$.

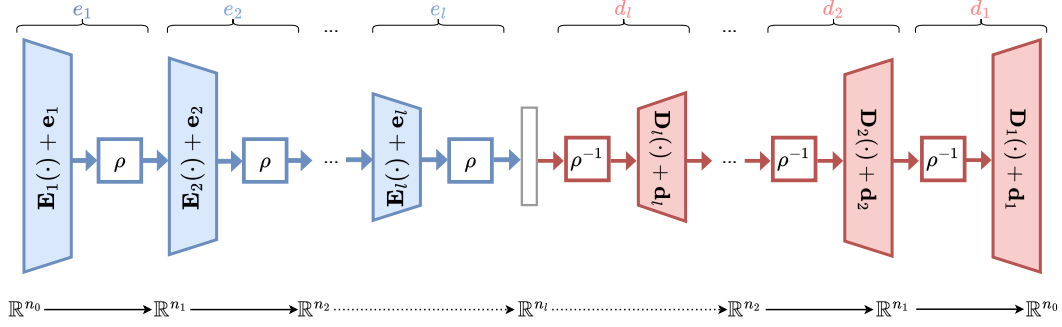


FIGURE 1. Schematic representation of a deep symmetric autoencoder.

Here, the action of ρ over column vectors is intended componentwise. Notice that, in order to ease notation, we do not stress the dependency of each e_j on Ψ , although it is clear that $e_j = e_j(\Psi)$.

Similarly, Ψ also defines a decoding routine, $\mathcal{D}(\Psi, \cdot) : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_0}$, defined via

$$\mathcal{D}(\Psi, \mathbf{c}) := (d_1 \circ \dots \circ d_l)(\mathbf{c}),$$

where $d_j : \mathbf{h} \mapsto \mathbf{D}_j \rho^{-1}(\mathbf{h}) + \mathbf{d}_j$. Together, these are used to define the reconstruction algorithm

$$\mathcal{R}(\Psi, \mathbf{u}) := \mathcal{D}(\Psi, \mathcal{E}(\Psi, \mathbf{u})),$$

which acts as a map $\mathcal{R}(\Psi, \cdot) : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_0}$. We refer the reader to Fig. 1 for a schematic representation: there, it will become apparent why some authors describe symmetric autoencoders as having a “butterfly” architecture [1].

3.1. About the activation function

We highlight the fact that, differently from other works in the literature, we require the activation function to be invertible, so that we can use ρ for the encoder and ρ^{-1} for the decoder, respectively. The same construction can be found in [26]. We argue that this requirement is fundamental to obtain a truly symmetric architecture, as it highlights the specular role of the encoding and decoding mechanisms.

Additionally, in order to guarantee a minimal form of stability, we also require ρ to be bilipschitz. It can be shown, in fact, that this is equivalent to having both ρ and ρ^{-1} to be Lipschitz continuous: see Lemma 3.1 right below.

Lemma 3.1. *Let $\rho : \mathbb{R} \rightarrow \mathbb{R}$ be bilipschitz, meaning that there exists two positive constants, $\eta > 0$ and $L > 0$, such that*

$$\eta|x - y| \leq |\rho(x) - \rho(y)| \leq L|x - y| \quad \forall x, y \in \mathbb{R}. \quad (5)$$

Then, ρ is bijective and its inverse is also bilipschitz. Furthermore, Eq. (5) holds with $\eta = 1/\text{Lip}(\rho^{-1})$ and $L = \text{Lip}(\rho)$.

Proof. See Appendix A. □

This bilipschitz control will be fundamental for our construction, as it ensures a stable behavior both during training and during inference. Indeed, it is straightforward to prove that, for any $\Psi = \{(\mathbf{E}_j, \mathbf{D}_j, \mathbf{e}_j, \mathbf{d}_j, \rho)\}_{j=1}^l$, one has $\text{Lip}(\mathcal{R}(\Psi, \cdot)) \propto (\text{Lip}(\rho)\text{Lip}(\rho^{-1}))^{l-1}$. In this respect, we take the chance to introduce the notion of *sharpness*, a quantity that is specific of bilipschitz activations and that we define as

$$\angle(\rho) = \text{Lip}(\rho)\text{Lip}(\rho^{-1}) - 1 \geq 0. \quad (6)$$

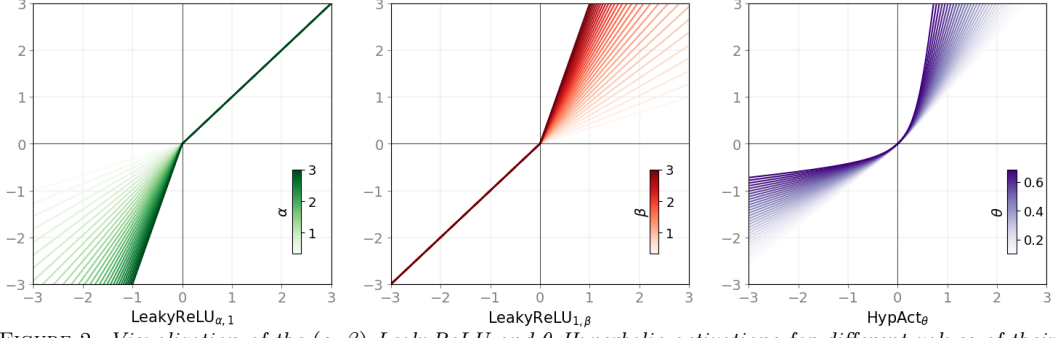


FIGURE 2. Visualization of the (α, β) -LeakyReLU and θ -Hyperbolic activations for different values of their hyperparameters.

Essentially, the sharpness provides a quantitative measure of the instability brought by ρ , and also serves to quantify its degree of nonlinearity: notice, in fact, that $\angle(\rho) \equiv 0$ whenever ρ is linear.

Examples of bilipschitz activations include:

- (i) the (α, β) -LeakyReLU activation, already considered in [35]. Given $\alpha, \beta > 0$, with $\alpha \neq \beta$, the latter is defined as

$$\text{LeakyReLU}_{\alpha, \beta}(x) = \alpha x \mathbf{1}_{(-\infty, 0)}(x) + \beta x \mathbf{1}_{[0, \infty)}(x),$$

where $\mathbf{1}_E$ denotes the characteristic function of the set E ;

- (ii) the θ -Hyperbolic activation, first proposed by Otto et al. [26]. Given $\theta \in (0, \pi/4)$, the latter reads

$$\text{HypAct}_{\theta}(x) = \frac{b_{\theta}}{a_{\theta}} x - \frac{\sqrt{2}}{a_{\theta} \sin \theta} + \frac{1}{a_{\theta}} \sqrt{\left(\frac{2x}{\sin \theta \cos \theta} - \frac{\sqrt{2}}{\cos \theta} \right)^2 + 2a_{\theta}},$$

where $a_{\theta} = \csc^2(\theta) - \sec^2(\theta)$ and $b_{\theta} = \csc^2(\theta) + \sec^2(\theta)$.

We refer the reader to Fig. 2 for a visualization of how the hyperparameters α, β and θ affect the two activations. It is worth mentioning that each activation function comes with its own strenghts and limitations. The (α, β) -LeakyReLU activation, for instance, has the advantage of providing a direct control over the Lipschitz constant of the nonlinearity used for encoding and decoding, respectively. In fact, if $\rho = \text{LeakyReLU}_{\alpha, \beta}$, then $\text{Lip}(\rho) = \max\{\alpha, \beta\}$ and $\text{Lip}(\rho^{-1}) = \max\{1/\alpha, 1/\beta\}$. This nonlinearity, however, is not as smooth as the θ -Hyperbolic activation, which, in contrast, has the benefit of being infinitely differentiable. In turn, the latter does not distinguish as much between encoding and decoding, since $\rho = \text{HypAct}_{\theta} \implies \text{Lip}(\rho) = \text{Lip}(\rho^{-1}) = \tan(\theta + \frac{\pi}{4})$.

3.2. Classes of symmetric autoencoders

Given a decreasing sequence of integers, $n_0 > n_1 \geq \dots \geq n_l > 0$, and a bilipschitz map $\rho: \mathbb{R} \rightarrow \mathbb{R}$, we shall write

$$\text{SAE}_{\rho}(n_0, \dots, n_l) = \prod_{j=1}^l [\mathbb{R}^{n_j \times n_{j-1}} \times \mathbb{R}^{n_{j-1} \times n_j} \times \mathbb{R}^{n_j} \times \mathbb{R}^{n_{j-1}} \times \{\rho\}],$$

for the space of all symmetric autoencoders associated with the given architecture.

We also introduce two additional subclasses, obtained by requiring the above networks to satisfy additional constraints related to their weights and biases. Specifically, we consider the subclass of *symmetric biorthogonal autoencoders*, already introduced in [26],

$$\begin{aligned} \text{SBAE}_\rho(n_0, \dots, n_l) := \{ \Psi \in \text{SAE}_\rho(n_0, \dots, n_l) : \mathbf{E}_j \mathbf{D}_j = \mathbf{I}_{n_j} \text{ and} \\ \mathbf{E}_j \mathbf{d}_j = -\mathbf{e}_j \quad \forall j = 1, \dots, l \}, \end{aligned}$$

and the one of *symmetric orthogonal autoencoders*,

$$\text{SOAE}_\rho(n_0, \dots, n_l) := \{ \Psi \in \text{SBAE}_\rho(n_0, \dots, n_l) : \mathbf{E}_j = \mathbf{D}_j^\top \}.$$

By definition, $\text{SOAE}_\rho(n_0, \dots, n_l) \subseteq \text{SBAE}_\rho(n_0, \dots, n_l) \subseteq \text{SAE}_\rho(n_0, \dots, n_l)$. Notice that, as one moves from the broader to the tighter class of autoencoders, the number of architecture parameters decreases. For instance, in the biorthogonal case, the encoding bias vectors $\mathbf{e}_j$ are univocally determined by $\mathbf{E}_j$ and $\mathbf{d}_j$ (observe that the latter relationship is inspired by linear methods, cf. Eq.(2), where a decoding shift equal to $\mathbf{q}$ results in an encoding shift of $-\mathbf{V}^\top \mathbf{q}$). Similarly, in the orthogonal case, knowing the encoding matrix is enough to derive the decoding matrix and vice versa. In light of this, we shall use the following short-hand notation:

- if $\Psi \in \text{SBAE}_\rho(n_0, \dots, n_l)$ we shall write $\{(\mathbf{E}_j, \mathbf{D}_j, \mathbf{b}_j, \rho)\}_{j=1}^l$ to intend the symmetric autoencoder $\{(\mathbf{E}_j, \mathbf{D}_j, -\mathbf{E}_j \mathbf{b}_j, \mathbf{b}_j, \rho)\}_{j=1}^l$;
- if $\Psi \in \text{SOAE}_\rho(n_0, \dots, n_l)$ we shall write $\{(\mathbf{V}_j, \mathbf{b}_j, \rho)\}_{j=1}^l$ to intend the symmetric autoencoder $\{(\mathbf{V}_j^\top, \mathbf{V}_j, -\mathbf{V}_j^\top \mathbf{b}_j, \mathbf{b}_j, \rho)\}_{j=1}^l$;

With this notation, the encoding layers read $e_j : \mathbf{h} \mapsto \rho(\mathbf{E}_j(\mathbf{h} - \mathbf{b}_j))$ in the biorthogonal case, and $e_j : \mathbf{h} \mapsto \rho(\mathbf{V}_j^\top(\mathbf{h} - \mathbf{b}_j))$ in the orthogonal case. Notice, in particular, how the encoding formula used in the orthogonal case finds a natural connection to the one adopted in the context of linear reduction: cf. Eq. (2). Similar considerations hold for the decoding phase as well.

Notwithstanding, it is worth emphasizing that the constraints fulfilled by $\text{SBAE}_\rho(n_0, \dots, n_l)$ and $\text{SOAE}_\rho(n_0, \dots, n_l)$, respectively, give the two classes several remarkable properties. In particular, one can show that all biorthogonal symmetric autoencoders are representation consistent: a given object and its reconstruction both map onto the same latent representation. As an immediate consequence, all $\Psi \in \text{SBAE}_\rho(n_0, \dots, n_l)$ act as non-linear projectors. Orthogonal symmetric autoencoders, in turn, have the additional benefit of providing a direct control over the Lipschitz constants of the encoder and the decoder modules, which can be extremely useful when model stability becomes a necessity. These facts are made rigorous in the Proposition below.

Proposition 3.1. *Let $n_0 > n_1 \geq \dots \geq n_l$ be positive integers and let $\rho : \mathbb{R} \rightarrow \mathbb{R}$ be bilipschitz. The following hold true.*

- (i) $\Psi \in \text{SBAE}_\rho(n_0, \dots, n_l) \Rightarrow \mathcal{E}(\Psi, \mathcal{D}(\Psi, \mathbf{c})) = \mathbf{c}$ for all $\mathbf{c} \in \mathbb{R}^{n_l}$.
- (ii) $\Psi \in \text{SBAE}_\rho(n_0, \dots, n_l) \Rightarrow \mathcal{R}(\Psi, \mathbf{v}) = \mathbf{v}$ for all $\mathbf{v} \in \mathcal{R}(\Psi, \mathbb{R}^{n_0})$.
- (iii) $\Psi \in \text{SOAE}_\rho(n_0, \dots, n_l) \Rightarrow \text{Lip}(\mathcal{E}(\Psi, \cdot)) \leq \text{Lip}(\rho)^l, \quad \text{Lip}(\mathcal{D}(\Psi, \cdot)) \leq \text{Lip}(\rho^{-1})^l$.

Proof. See Appendix A. □

3.3. Training of symmetric autoencoders

Given some $\mathbf{u} \in L^2(\mathcal{P}; \mathbb{R}^{n_0})$, and a corresponding training set $\mathbf{u}_1, \dots, \mathbf{u}_S \in \mathbb{R}^{n_0}$, consisting of S independent random realizations of $\mathbf{u}$, the practical implementation and training of a

symmetric autoencoder consists in: (i) choosing a suitable architecture, thus identifying a proper hypothesis class $\mathcal{HC} \subseteq \text{SAE}_\rho(n_0, \dots, n_l)$; (ii) training the model by minimizing the empirical loss

$$\mathcal{L}(\Psi) = \frac{1}{S} \sum_{i=1}^S \|\mathbf{u}_i - \mathcal{R}(\Psi, \mathbf{u}_i)\|^2. \quad (7)$$

In the case of unconstrained symmetric autoencoders, $\mathcal{HC} = \text{SAE}_\rho(n_0, \dots, n_l)$, such minimization problem can be addressed using classical optimization routines. Conversely, things become slightly more involved if we consider architectures with (bi-)orthogonality constraints, namely, if either $\mathcal{HC} = \text{SBAE}_\rho(n_0, \dots, n_l)$ or $\mathcal{HC} = \text{SOAE}_\rho(n_0, \dots, n_l)$. In those cases, in fact, one must guarantee that the optimization procedure does not exit the hypothesis manifold $\mathcal{HC}$, meaning the autoencoder must satisfy the (bi-)orthogonality constraints throughout the entire training phase.

One way to overcome this difficulty, and to make ensure that all orthogonality constraints are fulfilled *by design*, is to introduce a suitable parametrization. To this end, let $\pi_{\text{orth}} : \mathbf{A} \mapsto \tilde{\mathbf{A}}$, be any (differentiable) algorithm that, given a rectangular matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$, returns an orthonormal matrix $\tilde{\mathbf{A}} \in \mathcal{O}_{m,n}$ such that $\text{span}(\mathbf{A}) \subseteq \text{span}(\tilde{\mathbf{A}})$, with equality when $\mathbf{A}$ has full rank. An example of such algorithm in the `pytorch` library [28] is the one based on the use of Householder reflections.

SOAE. Given ρ and $n_0, \dots, n_l$, let $\Theta := \prod_{j=1}^l [\mathbb{R}^{n_{j-1} \times n_j} \times \mathbb{R}^{n_{j-1}}]$ denote our parametrization space. For any $\theta = \{(\mathbf{A}_j, \mathbf{b}_j)\}_{j=1}^l \in \Theta$, we denote by Ψ_θ the symmetric orthogonal autoencoder obtained by setting $\Psi_\theta := \{(\pi_{\text{orth}}(\mathbf{A}_j), \mathbf{b}_j, \rho)\}_{j=1}^l$. It is straightforward to see that such parametrization is surjective, meaning that any $\Psi \in \text{SOAE}_\rho(n_0, \dots, n_l)$ can be written as $\Psi = \Psi_\theta$ for some $\theta \in \Theta$. This allows us to reframe the learning problem in terms of the underlying parametrization, namely, using

$$\tilde{\mathcal{L}}(\theta) := \mathcal{L}(\Psi_\theta).$$

The training phase will then consist in finding the optimal $\theta_* \in \Theta$ that minimizes $\tilde{\mathcal{L}}$, and then assembling the corresponding autoencoder.

SBAE. A similar procedure can be carried out in the biorthogonal case, although the parametrization is significantly more involved. For $q \geq r$, one can construct two matrices $\mathbf{E} \in \mathbb{R}^{r \times q}$ and $\mathbf{D} \in \mathbb{R}^{q \times r}$ satisfying $\mathbf{E}\mathbf{D} = \mathbf{I}_r$ by letting

$$\mathbf{E}^\top := \mathbf{X} \begin{bmatrix} \mathbf{Y}\mathbf{S}\mathbf{Z}^\top \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{D} := \mathbf{X} \begin{bmatrix} \mathbf{Y}\mathbf{S}^{-1}\mathbf{Z}^\top \\ \mathbf{Q} \end{bmatrix}, \quad (8)$$

where, having set $d := \min\{r, q - r\}$, the matrices $\mathbf{X} \in \mathcal{O}_{q, r+d}$ and $\mathbf{Y}, \mathbf{Z} \in \mathcal{O}_{r, r}$ are orthonormal, whereas $\mathbf{0} \in \mathbb{R}^{d \times r}$ is the zero matrix, $\mathbf{Q} \in \mathbb{R}^{d \times r}$ is arbitrary and $\mathbf{S} \in \mathbb{R}^{r \times r}$ is a diagonal matrix with strictly positive entries over the diagonal. Notably, this construction is surjective, in the sense that one can easily show that any two biorthogonal matrices can be written as in (8): see Appendix C. Additionally, compared to other parametrizations seen in the literature, see, e.g., [26], the one proposed here is fairly inexpensive as it only requires inverting a diagonal matrix with full rank.

In light of this fact, given ρ and $n_0, \dots, n_l$, we define the optimization space underlying a symmetric biorthogonal autoencoder architecture as

$$\Theta := \prod_{j=1}^l \left[\mathbb{R}^{n_{j-1} \times (n_j + d_j)} \times \mathbb{R}^{n_j \times n_j} \times \mathbb{R}^{n_j \times n_j} \times \mathbb{R}^{d_j \times n_j} \times \mathbb{R}^{n_j} \times \mathbb{R}^{n_{j-1}} \right],$$

where $d_j := \min\{n_j, n_{j-1} - n_j\}$. For any $\theta = \{(\tilde{\mathbf{X}}_j, \tilde{\mathbf{Y}}_j, \tilde{\mathbf{Z}}_j, \mathbf{Q}_j, \mathbf{s}_j, \mathbf{b}_j)\}_{j=1}^l \in \Theta$ the corresponding autoencoder $\Psi_\theta \in \text{SBAE}_\rho(n_0, \dots, n_l)$ can be constructed as

$$\Psi_\theta = \{(\mathbf{E}_j, \mathbf{D}_j, \mathbf{b}_j, \rho)\}_{j=1}^l,$$

where each pair $(\mathbf{E}_j, \mathbf{D}_j)$ is assembled via (8) using the matrices $\mathbf{X}_j := \pi_{\text{orth}}(\tilde{\mathbf{X}}_j)$, $\mathbf{Y}_j := \pi_{\text{orth}}(\tilde{\mathbf{Y}}_j)$, $\mathbf{Z}_j := \pi_{\text{orth}}(\tilde{\mathbf{Z}}_j)$, $\mathbf{Q}_j$ and $\mathbf{S}_j := \text{diag}(\tau(\mathbf{s}_j))$, with $\tau(s) := \log_2(1 + 2^s)$ the softplus transformation, here applied componentwise on vector entries. Now, as in the orthogonal case, the optimization can be carried out in the unconstrained optimization space Θ rather than in $\text{SBAE}_\rho(n_0, \dots, n_l)$.

Remark 3.1. *Both in the case of orthogonal and biorthogonal constraints, the proposed parametrizations are surjective, thus ensuring a proper exploration of the hypothesis class during the training phase. Additionally, this approach guarantees that the constraints are satisfied by design. This is in contrast to other strategies adopted in the literature, where the constraints are only weakly imposed. For example, in [26], the authors employ biorthogonal architectures, but the training is conducted over a broader hypothesis space, $\text{SBAE}_\rho(n_0, \dots, n_l) \subset \mathcal{HC} \subset \text{SAE}_\rho(n_0, \dots, n_l)$, with the biorthogonality requirement only being enforced indirectly by including an additional penalty term in the loss function (7).*

4. Insights from the Eckart-Young-Schmidt perspective

In this Section we shall derive some quantitative error bounds on the reconstruction error of symmetric autoencoders and use those to develop a new initialization strategy. We start by framing everything in an abstract setting, but later transition to a more practical perspective, focusing on the empirical case in which the autoencoder has to be constructed based on a finite amount of data.

4.1. Theoretical considerations

Let $\mathbf{u} \in L^2(\mathcal{P}; \mathbb{R}^{n_0})$. Given $0 < n_l \leq \dots \leq n_1 \leq n_0$ and a bilipschitz $\rho : \mathbb{R} \rightarrow \mathbb{R}$, we are interested in studying the reconstruction error

$$\mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2, \quad (9)$$

where $\Psi \in \text{SAE}_\rho(n_0, \dots, n_l)$. To start, we notice that one has the following chain of inequalities,

$$\begin{aligned} \inf_{\Psi \in \text{SOAE}_\rho} \mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 &\geq \inf_{\Psi \in \text{SBAE}_\rho} \mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 \\ &\geq \inf_{\Psi \in \text{SAE}_\rho} \mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 \geq \sum_{i=n_1+1}^{n_0} \lambda_i(\Sigma[\mathbf{u}]), \end{aligned} \quad (10)$$

where we have dropped the dependency on $(n_0, \dots, n_l)$ to enhance readability. The first two inequalities are obvious, as they stem from the fact that the autoencoder classes SAE_ρ , SBAE_ρ and SOAE_ρ are nested. The last inequality, instead, follows trivially by noting that for each fixed $\Psi = \{(\mathbf{E}_j, \mathbf{D}_j, \mathbf{e}_j, \mathbf{d}_j, \rho)\}_{j=1}^l \in \text{SAE}_\rho(n_0, \dots, n_l)$, regardless of the nonlinearities in the autoencoding process, the outputs of $\mathcal{R}(\Psi, \cdot)$ will always fall inside the n_1 -dimensional affine subspace spanned by the last layer of the decoder module,

$$V_\Psi := \{\mathbf{D}_1 \mathbf{c} + \mathbf{d}_1 : \mathbf{c} \in \mathbb{R}^{n_1}\} \subset \mathbb{R}^{n_0}.$$

Then, as we discussed in Section 2, this immediately implies $\mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 \geq \mathbb{E}[\inf_{\mathbf{v} \in V_\Psi} \|\mathbf{u} - \mathbf{v}\|^2] \geq \sum_{i=n_1+1}^{n_0} \lambda_i(\Sigma[\mathbf{u}])$, thus yielding (10).

This highlights the fact that the first hidden dimension, n_1 , plays a crucial role, as it directly affects the overall expressivity of the architecture. Since (10) already provides a lower-bound to (9), our purpose now is to gain some additional insights by deriving a suitable upper-bound. To this end, leveraging (10), we shall focus for a moment on the smaller class of symmetric orthogonal autoencoders.

In this respect, we start by stating a very useful lemma, upon which the rest of our derivation is based.

Lemma 4.1. *Let $\Psi = \{(\mathbf{V}_j, \mathbf{b}_j, \rho)\}_{j=1}^l \in \text{SOAE}_\rho(n_0, \dots, n_l)$, where $n_0 \geq n_1 \geq n_2 \geq \dots \geq n_l$ for some $l \geq 2$, and $\rho : \mathbb{R} \rightarrow \mathbb{R}$ is bilipschitz. Let $e_j = e_j(\Psi)$ and $d_j := d_j(\Psi)$. Define $E_k := e_k \circ \dots \circ e_1$ and $D_k := d_k \circ \dots \circ d_l$. Let $E_0 := \text{Id}_{n_0}$ and $D_{l+1} := \text{Id}_{n_0}$ act as the identity. Then, for all $k = 1, \dots, l$, one has*

$$\|E_{k-1}(\mathbf{v}) - D_k(E_l(\mathbf{v}))\|^2 \geq \|E_{k-1}(\mathbf{v}) - [\mathbf{V}_k \mathbf{V}_k^\top (E_{k-1}(\mathbf{v}) - \mathbf{b}_k) + \mathbf{b}_k]\|^2 \quad (11)$$

$$+ \text{Lip}(\rho)^{-2} \|E_k(\mathbf{v}) - D_{k+1}(E_l(\mathbf{v}))\|^2,$$

$$\|E_{k-1}(\mathbf{v}) - D_k(E_l(\mathbf{v}))\|^2 \leq \|E_{k-1}(\mathbf{v}) - [\mathbf{V}_k \mathbf{V}_k^\top (E_{k-1}(\mathbf{v}) - \mathbf{b}_k) + \mathbf{b}_k]\|^2 \quad (12)$$

$$+ \text{Lip}(\rho^{-1})^2 \|E_k(\mathbf{v}) - D_{k+1}(E_l(\mathbf{v}))\|^2,$$

for all $\mathbf{v} \in \mathbb{R}^{n_0}$.

Proof. Notice that for all $\mathbf{v} \in \mathbb{R}^{n_0}$

$$\begin{aligned} \|E_{k-1}(\mathbf{v}) - D_k(E_l(\mathbf{v}))\| &\stackrel{\text{def}}{=} \|E_{k-1}(\mathbf{v}) - d_k(D_{k+1}(E_l(\mathbf{v})))\|^2 \\ &\stackrel{\text{def}}{=} \|E_{k-1}(\mathbf{v}) - \mathbf{V}_k \rho^{-1}(D_{k+1}(E_l(\mathbf{v}))) - \mathbf{b}_k\|^2 \\ &\stackrel{\text{orth}}{=} \|[E_{k-1}(\mathbf{v}) - \mathbf{b}_k] - \mathbf{V}_k \mathbf{V}_k^\top [E_{k-1}(\mathbf{v}) - \mathbf{b}_k]\|^2 \\ &\quad + \|\mathbf{V}_k \mathbf{V}_k^\top [E_{k-1}(\mathbf{v}) - \mathbf{b}_k] - \mathbf{V}_k \rho^{-1}(D_{k+1}(E_l(\mathbf{v})))\|^2, \end{aligned} \quad (13)$$

the first two equalities following by definition and the last one by orthogonality. Notice, in fact, that both $\mathbf{V}_k \mathbf{V}_k^\top [E_{k-1}(\mathbf{v}) - \mathbf{b}_k]$ and $\mathbf{V}_k \rho^{-1}(D_{k+1}(E_l(\mathbf{v})))$ belong to $\text{span}(\mathbf{V}_k)$, to which $[E_{k-1}(\mathbf{v}) - \mathbf{b}_k] - \mathbf{V}_k \mathbf{V}_k^\top [E_{k-1}(\mathbf{v}) - \mathbf{b}_k]$, the projection residual, is orthogonal. We now observe that the second term at the end of Eq. (13) can be re-written as

$$\begin{aligned} \|\mathbf{V}_k \mathbf{V}_k^\top [E_{k-1}(\mathbf{v}) - \mathbf{b}_k] - \mathbf{V}_k \rho^{-1}(D_{k+1}(E_l(\mathbf{v})))\|^2 &= \\ &= \|\mathbf{V}_k^\top [E_{k-1}(\mathbf{v}) - \mathbf{b}_k] - \rho^{-1}(D_{k+1}(E_l(\mathbf{v})))\|^2 \\ &= \|\rho^{-1}(\rho(\mathbf{V}_k^\top [E_{k-1}(\mathbf{v}) - \mathbf{b}_k])) - \rho^{-1}(D_{k+1}(E_l(\mathbf{v})))\|^2 \\ &= \|\rho^{-1}(E_k(\mathbf{v})) - \rho^{-1}(D_{k+1}(E_l(\mathbf{v})))\|^2. \end{aligned}$$

Finally, thanks to the bilipschitzness of ρ , cf. Lemma 3.1, we obtain the chain of inequalities

$$\begin{aligned} \text{Lip}(\rho)^{-2} \|E_k(\mathbf{v}) - D_{k+1}(E_l(\mathbf{v}))\|^2 &\leq \|\rho^{-1}(E_k(\mathbf{v})) - \rho^{-1}(D_{k+1}(E_l(\mathbf{v})))\|^2 \\ &\leq \text{Lip}(\rho^{-1})^2 \|E_k(\mathbf{v}) - D_{k+1}(E_l(\mathbf{v}))\|^2. \end{aligned}$$

The above, combined with (13), yields the desired bounds. $\square$

Lemma 4.1 takes advantage of the compositional structure of deep symmetric orthogonal autoencoders to derive suitable error bounds characterizing the projection error at each level $k = 1, \dots, l$. Leveraging Lemma 4.1 in an iterative fashion, we can then easily prove the following general result.

Theorem 4.1. *Let $\mathbf{u} \in L^2(\mathcal{D}; \mathbb{R}^{n_0})$. Let $n_0 \geq n_1 \geq n_2 \geq \dots \geq n_l$, $l \geq 2$, and let $\rho : \mathbb{R} \rightarrow \mathbb{R}$ be bilipschitz. For all $\Psi = \{(\mathbf{V}_j, \mathbf{b}_j, \rho)\}_{j=1}^l \in \text{SOAE}_\rho(n_0, \dots, n_l)$ one has*

$$\mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 \geq \sum_{k=0}^{l-1} \text{Lip}(\rho)^{-2k} \mathbb{E}\|E_k(\mathbf{u}) - [\mathbf{V}_{k+1} \mathbf{V}_{k+1}^\top (E_k(\mathbf{u}) - \mathbf{b}_{k+1}) + \mathbf{b}_{k+1}]\|^2, \quad (14)$$

$$\mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 \leq \sum_{k=0}^{l-1} \text{Lip}(\rho^{-1})^{2k} \mathbb{E}\|E_k(\mathbf{u}) - [\mathbf{V}_{k+1} \mathbf{V}_{k+1}^\top (E_k(\mathbf{u}) - \mathbf{b}_{k+1}) + \mathbf{b}_{k+1}]\|^2, \quad (15)$$

where $E_k := (e_k \circ \dots \circ e_1)$, being $e_j := e_j(\Psi)$ and $E_0 := \text{Id}_{n_0}$.

Proof. We start by proving the lower bound (14). By Lemma 4.1,

$$\begin{aligned} \mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 &= \mathbb{E}\|E_0(\mathbf{u}) - D_1(E_l(\mathbf{u}))\|^2 \geq \mathbb{E}\|E_0(\mathbf{u}) - [\mathbf{V}_1 \mathbf{V}_1^\top (E_0(\mathbf{u}) - \mathbf{b}_1) + \mathbf{b}_1]\|^2 \\ &\quad + \text{Lip}(\rho)^{-2} \mathbb{E}\|E_1(\mathbf{u}) - D_2(E_l(\mathbf{u}))\|^2. \end{aligned}$$

Then, iteratively applying Lemma 4.1 to the last term at the right-hand-side yields

$$\begin{aligned} \mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 &\geq \dots \\ &\geq \sum_{k=0}^{l-1} \text{Lip}(\rho)^{-2k} \mathbb{E}\|E_k(\mathbf{u}) - [\mathbf{V}_{k+1} \mathbf{V}_{k+1}^\top (E_k(\mathbf{u}) - \mathbf{b}_{k+1}) + \mathbf{b}_{k+1}]\|^2 \\ &\quad + \text{Lip}(\rho)^{-2l} \mathbb{E}\|E_l(\mathbf{u}) - D_{l+1}(E_l(\mathbf{u}))\|^2. \end{aligned}$$

Notice, however, that the last term is actually zero as D_{l+1} is the identity. This proves (14). Mirroring the above steps, but using the upper bound result in Lemma 4.1, yields (15) and thus concludes the proof. $\square$

What is most interesting about Theorem 4.1 is that it bounds the reconstruction error in terms of l terms, one per each layer, with the k th term connecting the k th hidden representation in the architecture, $E_k(\mathbf{u})$, with the trainable parameters at level $k+1$, $\mathbf{V}_{k+1}$ and $\mathbf{b}_{k+1}$. Furthermore, each term in Eqs. (14)-(15) can be interpreted as a linear reduction, happening at level k .

In particular, this suggests minimizing the upper-bound in (15) by addressing each term alone, following a sort of greedy approach. These considerations lead to the following Corollary.

Corollary 4.1. *Let $\mathbf{u} \in L^2(\mathcal{P}; \mathbb{R}^{n_0})$, $n_0 \geq n_1 \geq n_2 \geq \dots \geq n_l$ with $l \geq 2$, and let $\rho : \mathbb{R} \rightarrow \mathbb{R}$ be bilipschitz. Set $\tilde{E}_0 := \text{Id}_{n_0}$. Starting with $k = 0$,*

- *define $\tilde{\mathbf{b}}_{k+1} := \mathbb{E}[\tilde{E}_k(\mathbf{u})]$,*
- *construct $\tilde{\mathbf{V}}_{k+1}$ by extracting the n_{k+1} eigenvectors of $\Sigma[\tilde{E}_k(\mathbf{u})]$ associated to the n_{k+1} largest eigenvalues,*
- *define $\tilde{E}_{k+1} : \mathbb{R}^{n_k} \rightarrow \mathbb{R}^{n_{k+1}}$ by setting $\tilde{E}_{k+1}(\mathbf{v}) := \rho(\tilde{\mathbf{V}}_{k+1}^\top (\tilde{E}_k(\mathbf{v}) - \tilde{\mathbf{b}}_{k+1}))$.*

Iterate the previous steps for $k = 1, \dots, l-1$. Then,

$$\begin{aligned} \inf_{\Psi \in \text{SAE}_\rho} \mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 &\leq \inf_{\Psi \in \text{SBAE}_\rho} \mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 \\ &\leq \inf_{\Psi \in \text{SOAE}_\rho} \mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2 \\ &\leq \sum_{k=0}^{l-1} \text{Lip}(\rho^{-1})^{2k} \sum_{i=n_{k+1}+1}^{n_k} \lambda_i(\Sigma[\tilde{E}_k(\mathbf{u})]). \end{aligned} \quad (16)$$

We highlight that Eqs. (10) and (16) are complementary, as they provide lower and upper bounds for the reconstruction error of the optimal SAE, respectively. Despite being looser than the ones provided in Theorem 4.1, such bounds have a practical implication on design and initialization of symmetric autoencoders, as we discuss below.

4.2. From theory to practice: the EYS initialization

Inspired by the considerations made in Section 4.1, particularly by Corollary 4.1, we propose a new initialization strategy for symmetric autoencoders hereon referred to as “Eckart-Young-Schmidt (EYS) initialization”. The idea is to mimic the greedy procedure described in Corollary 4.1, ultimately translating the latter to an empirical framework.

More precisely, let $\mathbf{u} \in L^2(\mathcal{P}, \mathbb{R}^{n_0})$, and assume we are given a collection of S i.i.d. random realizations of $\mathbf{u}$, denoted as $\mathbf{u}_1, \dots, \mathbf{u}_S$. Fix any bilipschitz activation ρ and a suitable skeleton $n_0 \geq n_1 \geq \dots \geq n_l$ for a symmetric autoencoder. The idea is to leverage the data in order to come up with a suitable autoencoder $\Psi \in \mathcal{HC} \subseteq \text{SAE}_\rho(n_0, \dots, n_l)$ that constitutes a first initial guess to be used as a starting point for the training phase. As before, $\mathcal{HC}$ denotes the hypothesis class, which can be either the whole set of symmetric autoencoders, or may be limited to either orthogonal or biorthogonal architectures. Our proposed strategy, which is independent of $\mathcal{HC}$, is detailed in Algorithm 1.

Algorithm 1 EYS initialization (pseudo-code)

Inputs: snapshot matrix $\mathbf{U} \in \mathbb{R}^{n_0 \times S}$ listing S i.i.d. samples of $\mathbf{u}$, bilipschitz activation ρ , architecture skeleton $\{n_j\}_{j=0}^l$

$\mathbf{Z} \leftarrow \mathbf{U}$ #initialize latent vectors

for $j = 1 : l$ **do**

$\mathbf{b}_j \leftarrow \bar{\mathbf{Z}}$ #compute sample mean

$\mathbf{W}_{j, -, -} = \text{SVD}(\mathbf{Z} - \mathbf{b}_j)$ #obtain left eigenvectors

$\mathbf{V}_j \leftarrow \mathbf{W}_j[:, : n_j]$ #retain first n_j vectors

$\mathbf{Z} \leftarrow \rho(\mathbf{V}_j^\top (\mathbf{Z} - \mathbf{b}_j))$ #compute next latent vectors

end for

$\Psi_{\text{EYS}} \leftarrow \{(\mathbf{V}_j, \mathbf{b}_j, \rho)\}_{j=1}^l \in \text{SOAE}_\rho(n_0, \dots, n_l)$ #build autoencoder

return Ψ_{EYS}

We highlight that Algorithm 1 follows the same construction seen in Corollary 4.1, but it leverages the empirical probability distribution $\tilde{\mathbb{P}} := \frac{1}{S} \sum_{i=1}^S \delta_{\mathbf{u}_i}$, rather than the true law $\mathbb{P}$ of the random vector $\mathbf{u} \sim \mathbb{P}$. Here, $\delta_{\mathbf{u}_i}$ denotes the Dirac delta distribution centered at $\mathbf{u}_i$. Consequently, expected values appearing in Corollary 4.1 are replaced by statistical averages, whereas the eigenvectors of the covariance matrices are approximated via empirical SVD (up to centering). In particular, we notice that the EYS initialization ultimately consists in an iterative application of the SVD, progressively going from n_0 to n_l . Relatedly, it is worth noting that this idea can be understood as a data-driven initialization (or pre-training) strategy, in the same spirit of [16, 3, 34].

We also point out that, regardless of the chosen hypothesis class $\mathcal{HC}$, the EYS initialization always returns a symmetric orthogonal autoencoder, $\Psi_{\text{EYS}} \in \text{SOAE}_\rho(n_0, \dots, n_l)$, coherently with the fact that

$$\text{SOAE}_\rho(n_0, \dots, n_l) \subseteq \mathcal{HC} \subseteq \text{SAE}_\rho(n_0, \dots, n_l).$$

However, we also remark that during the training phase the autoencoder will be free to explore the whole hypothesis class, potentially exiting the submanifold of SOAE.

In this concern, it is worth spending a few words in order to clarify how the EYS initialization can be incorporated in the training procedure discussed in Section 3.3. Let $\Psi_{\text{EYS}} = \{(\mathbf{V}_j, \mathbf{b}_j, \rho)\}_{j=1}^l$. We distinguish between the three possible choices of the hypothesis class.

- (i) $\mathcal{HC} = \text{SAE}_\rho(n_0, \dots, n_l)$. Here, we just set $\mathbf{E}_j := \mathbf{V}_j^\top$, $\mathbf{D}_j = \mathbf{V}_j$, $\mathbf{e}_j = -\mathbf{V}_j^\top \mathbf{b}_j$, and $\mathbf{d}_j = \mathbf{b}_j$. Then, we optimize $\{(\mathbf{E}_j, \mathbf{D}_j, \mathbf{e}_j, \mathbf{d}_j)\}_{j=1}^l$ freely, as no constraints are involved.

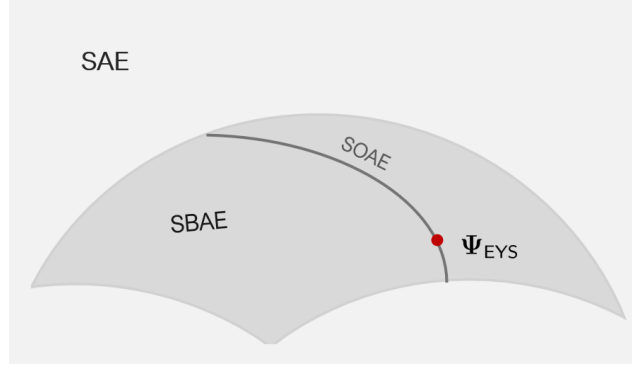


FIGURE 3. Pictorial representation of the EYS initialization and its relationship with the three manifolds, of decreasing dimension, representing the three (nested) classes of symmetric autoencoder architectures: SAE, SBAE and SOAE.

- (ii) $\mathcal{HC} = \text{SBAE}_\rho(n_0, \dots, n_l)$. In this case, one must first translate everything according to the parametrization discussed in Section 3.3, which serves to reframe the training procedure as an unconstrained optimization problem. To this end, we set

$$\tilde{\mathbf{X}}_j := [\mathbf{V}_j, \mathbf{M}_j], \quad \tilde{\mathbf{Y}}_j = \tilde{\mathbf{Z}}_j = \mathbf{I}_{n_j}, \quad \mathbf{Q}_j = \mathbf{0}, \quad \mathbf{s}_j = [0, \dots, 0]^\top,$$

where $\mathbf{I}_{n_j}$ is the $n_j \times n_j$ identity matrix, whereas $\mathbf{M}_j$ is any $n_{j-1} \times d_j$ orthonormal matrix whose span is orthogonal to $\text{span}(\mathbf{V}_j)$. In practice, following the notation in Algorithm 1, this can be constructed by extracting part of the remaining columns in the matrix $\mathbf{W}_j$, namely, $\mathbf{M}_j = \mathbf{W}_j[:, n_j + 1 : n_j + d_j]$.

It is straightforward to prove that with such construction, if we let

$$\boldsymbol{\theta}_{\text{EYS}} := \{(\tilde{\mathbf{X}}_j, \tilde{\mathbf{Y}}_j, \tilde{\mathbf{Z}}_j, \mathbf{Q}_j, \mathbf{s}_j, \mathbf{b}_j)\}_{j=1}^l$$

then $\boldsymbol{\Psi}_{\boldsymbol{\theta}_{\text{EYS}}} = \boldsymbol{\Psi}_{\text{EYS}}$, meaning that the proposed parameter vector is consistent with the proposed initialization. Consequently, the training phase can now be carried out in the optimization space starting from $\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{EYS}}$.

- (iii) $\mathcal{HC} = \text{SOAE}_\rho(n_0, \dots, n_l)$. Here, as in the previous case, one needs to find a suitable $\boldsymbol{\theta}_{\text{EYS}} \in \Theta$ in the optimization space that realizes $\boldsymbol{\Psi}_{\text{EYS}}$ —this time according to the notation for SOAE architectures: see Section 3.3. Here, however, this is much easier to do as one can simply set $\boldsymbol{\theta}_{\text{EYS}} := \{(\mathbf{V}_j, \mathbf{b}_j)\}_{j=1}^l$.

In other words, our proposal is to use the same initialization for all classes of symmetric autoencoders: the choice of the hypothesis class will only impact the architecture during the training phase, as it will restrict the exploration to the corresponding submanifold. We refer to Figure 3 for a pictorial representation. Finally, we remark that, even if the EYS initialization has been developed specifically for symmetric autoencoders, in principle, it can also be applied to generic autoencoder architectures. However, the underlying reasoning would then lose its theoretical significance.

5. Numerical experiments

Within this section, we provide a comprehensive series of numerical experiments aimed at (i) assessing the accuracy and computational performance of deep symmetric autoencoders, in their three flavours (SAE, SBAE, SOAE), (ii) validating the efficacy of our proposed initialization strategy, and (iii) unveiling the effect of design choices on the reconstruction accuracy. The training of the architectures is performed on a NVIDIA L40S 40GB GPU. The code implementation is released and available at <https://github.com/briviosimone/sae-eyes>.

Case study	Parameter	Range	Description
PGA	μ	[0.3, 0.7]	wave position
	ρ	[0.5, 2]	body force
	λ	[0.9, 1.1]	first Lamé coefficient
ELS	μ	[0.5, 1]	second Lamé coefficient
	m	[1.0, 1.5]	mass of the falling object
	δ	[0.05, 0.1]	size of the falling object
	x_0	[0.3, 0.7]	impact location
	θ	$[\frac{\pi}{4}, \frac{3\pi}{4}]$	collision angle
	T_1	[80, 120]	temperature at the inner left circle
ROD	T_2	[80, 120]	temperature at the inner right circle
	T_3	[80, 120]	temperature at the toothed region
	$\log_{10} f$	[1.5, 5]	source term (log scale)
	$\log_{10} \alpha$	[-3.5, -1.5]	irradiation coefficient (log scale)

TABLE 1. Problem parameters for the three case studies considered in the numerical experiments. The variability of such parameters is responsible of the stochasticity of the high-dimensional datum $\mathbf{u}$.

5.1. Datasets description

For our analysis, we shall consider data coming from three different case studies. All problems are formulated within the framework of *discretized functional data*, which we consider propaedeutic to applications involving reduced order modeling of parametrized PDEs. Specifically, the input dimension n_0 arises from the discretization of a spatial domain $\Omega \subset \mathbb{R}^d$, obtained by considering either a uniform grid or an appropriate simplicial mesh. Within this setting, the state vector $\mathbf{u} \in \mathbb{R}^{n_0}$ ultimately serves as a discrete representation of some underlying functional object. That is, being $\{x_i\}_{i=1}^{n_0}$ the nodes used for the spatial discretization, one has $u_i = u(x_i)$ for some $u : \Omega \rightarrow \mathbb{R}$. Specifically, in our case, $\mathbf{u}$ will be the discrete vector representing the solution of some parameter dependent PDE, and the variability of the solution $\mathbf{u}$ will be obtained by letting the parameters of the PDE vary within a suitable range: see Table 1 and Fig. 4.

Consequently, we emphasize that all the data is synthetic and is generated using either analytic formulae or numerical solvers. We exploit this controlled environment to ensure a fair comparison of the different architectures across the different case studies. For each numerical experiment, we collect 800 realizations of the state vector $\mathbf{u}$, generated by repeatedly sampling different values of the problem parameters. Each parameter is sampled independently and uniformly over its respective physical range (see Table 1). After computing the $S_{\text{tot}} = 800$ simulations, we split them between training (1/4 of the total), validation (1/8) and test set (5/8). To guarantee a fair comparison, we conduct all the experiments following a standardized training routine. Specifically, we employ 1500 epochs (with a patience of 500 for early stopping), using Adam [18] with $lr = 10^{-3}$ as learning rate, $B = 8$ as batch size, and min-max normalization for all of our experiments. The simulations in the test set, instead, are used to compute the empirical mean-squared error, $\text{MSE}_{\text{test}} = \frac{1}{S_{\text{test}}} \sum_{j=1}^{S_{\text{test}}} \|\mathbf{u}_j^{\text{test}} - \mathcal{R}(\Psi, \mathbf{u}_j^{\text{test}})\|^2$, which serves as a Monte Carlo approximation of $\mathbb{E}\|\mathbf{u} - \mathcal{R}(\Psi, \mathbf{u})\|^2$.

We provide a brief description of the three case studies right below.

Parameterized Gaussian (PGA). As the first test case, we select a classical experiment setup stemming from the reduced order modeling community, namely, the one-dimensional

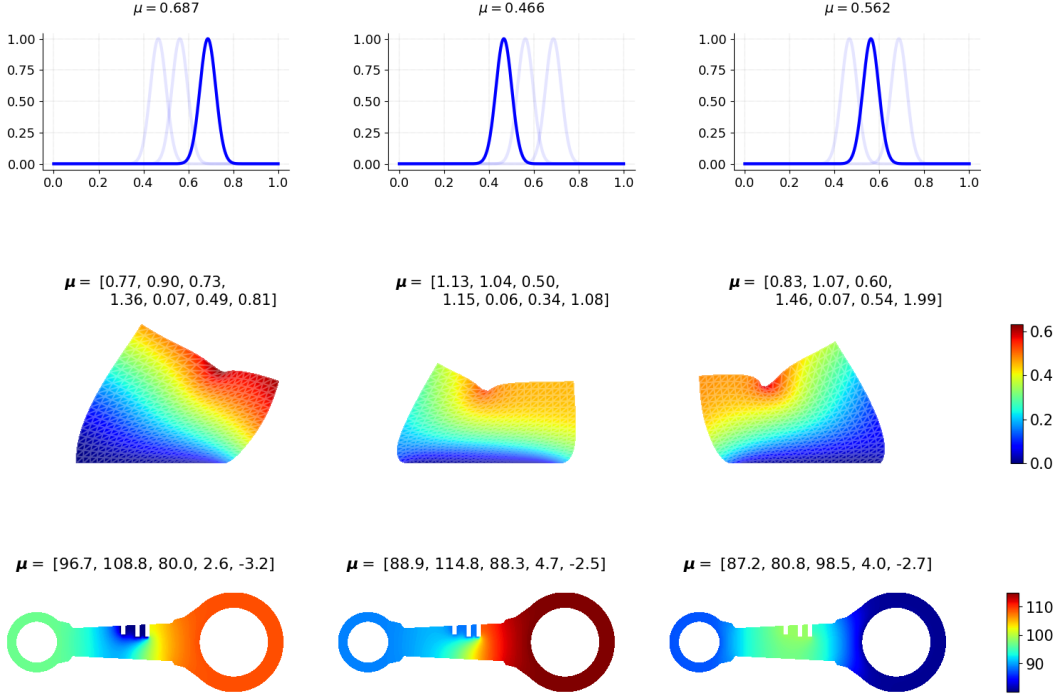


FIGURE 4. Random samples of $\mathbf{u}$ for the three test cases considered in this work: PGA (top), ELS (middle), and ROD (bottom). We denote with $\boldsymbol{\mu}$ the collection of problem parameters associated to each simulation.

parameterized Gaussian function,

$$u(x; \mu) = \exp(-400(x - \mu)^2), \quad x \in [0, 1],$$

having $\mu \in [0.3, 0.7]$ as a unique parameter. The latter can be thought as the solution at terminal time $T = 1$ of the advection equation $\partial_t u + \mu \partial_x u = 0$ with a fixed initial profile. For the sake of the discretization we use a uniform grid of stepsize $h = 1/513$, resulting in a problem dimension of $n_0 = 514$.

Linear elasticity (ELS). For this case study, we consider a PDE model describing the collision between a falling object and a block of elastic material,

$$\begin{cases} -\nabla \cdot (\lambda \nabla \cdot \mathbf{u} + \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^\top)) = \mathbf{f} & \text{in } \Omega \\ \mathbf{u} = \mathbf{0} & \text{on } \Gamma_{\text{bottom}} \\ \lambda (\nabla \cdot \mathbf{u}) \cdot \mathbf{n} + \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^\top) \cdot \mathbf{n} = \mathbf{T} & \text{on } \Gamma_{\text{top}} \\ \lambda (\nabla \cdot \mathbf{u}) \cdot \mathbf{n} + \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^\top) \cdot \mathbf{n} = \mathbf{0} & \text{on } \partial\Omega \setminus (\Gamma_{\text{bottom}} \cup \Gamma_{\text{top}}). \end{cases} \quad (17)$$

Here, $\Omega = (0, 1)^2$ is the spatial domain, Γ_{bottom} and Γ_{top} are the bottom and top edge, while $\mathbf{T}(x, y) = \mathbf{1}_{[x_0 - \delta, x_0 + \delta]}(x) \cdot [\cos \theta, -\sin \theta]^\top m$ and $\mathbf{f} = [0, -\rho]^\top$ are the imposed traction at the top edge and the forcing term, respectively. The problem depends on 7 parameters, modeling the mechanical properties of the absorbing block and the falling object: we refer to Table 1 for a short description about their meaning.

We approximate the solution to (17) using continuous piecewise-linear finite elements ($\mathbb{P}^1$ -FEM) over a mesh consisting of 441 vertices. Notice that, since the solution to (17) is vector-valued, the actual problem dimension is $n_0 = 2 \times 441 = 882$. We remark that, under the parametric ranges considered in Table 1, the displacement fields obtained by solving (17) end up violating the small deformation hypothesis, thus resulting in solutions with

poor physical significance. Still, because our problem formulation is effectively adimensional, physical significance can be recovered by suitably rescaling the parameters that contribute to the right hand side of the variational problem. Here, we deliberately refrain from doing so in order to make the resulting deformations more clearly visible.

Nonlinear diffusion in a rod geometry (ROD). To generate the third dataset, we solve repeatedly a nonlinear PDE describing the temperature distribution in a non-trivial rod geometry, for different configurations of the governing parameters: see Table 1. The PDE reads as follows,

$$\begin{cases} -\nabla \cdot (\sigma(u)\nabla u) = f - \alpha u^4 & \text{in } \Omega \\ \sigma(u)\nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \setminus (\cup_{i=1}^3 \Gamma_i) \\ u = T_i & \text{on } \Gamma_i, \end{cases} \quad (18)$$

where $\sigma(u) := 10^3 + e^{u/8}$ is the diffusion coefficient, whereas $\Gamma_1, \Gamma_2, \Gamma_3$ are the inner left circle, inner right circle, and the “toothed region”, respectively: see Fig. 4. We approximate the solution to (18) using $\mathbb{P}^1$ -FEM with $n_0 = 4347$ degrees of freedom, distributed across a nonstructured triangular mesh.

5.2. Initialization

In the following, we validate the efficacy of our initialization strategy by comparing it with the standard ones proposed in the literature. In the case of unconstrained architectures, we consider the so-called “He initialization” [15] as our main competitor. Specifically, we use the original algorithm when ρ is the (α, β) -LeakyReLU. Conversely, we rely a suitable generalization of the latter when considering other activations. We mention, in fact, that it is possible to extend the underlying reasoning put forth in [15] to other bilipschitz activations: we defer such discussion to the Appendix B. In the case of autoencoders with (bi)-orthogonality constraints, instead, we consider the initialization proposed by Otto et al. [26] as a benchmark. The latter consists in randomly sampling the weight matrices from the orthogonal group, while initializing all bias vectors to zero.

In order to assess the effectiveness of the EYS initialization, we investigate its impact on: (i) the validation MSE before training, to see whether it provides a good initial guess; (ii) the training dynamics, to see whether it affects the final accuracy and the speed of convergence. We summarize our findings below.

Before training. For the sake of brevity, we limited this preliminary analysis to the PGA case study: the ELS and ROD test cases will be addressed in the next paragraph. We report our findings in Figure 5, where we explore how different SAE architectures of variable latent dimension n_l and depth l perform before training, comparing their MSE under the EYS and the He initialization, respectively. The analysis, here reported relatively to the HypAct $_{\theta}$ activation, was repeated for different values of the θ -parameter, corresponding to different sharpness values of the nonlinearity $\rho = \rho_{\theta}$.

We observe that, as expected, the latent dimension does not influence the initial MSE value in the “standard” initialization, regardless of the sharpness of the activation function. Conversely, being data-driven, the EYS initialization shows a strong dependency between the starting MSE and the latent dimension. In particular, for larger values of n_l the EYS initialization performs significantly better —by several orders of magnitude— when compared to the literature standard. Notably, this improvement is found consistently for all sharpness values.

The MSE of a SAE initialized with the EYS initialization, however, tends to deteriorate as the architecture gets deeper and deeper, especially when combined with activation

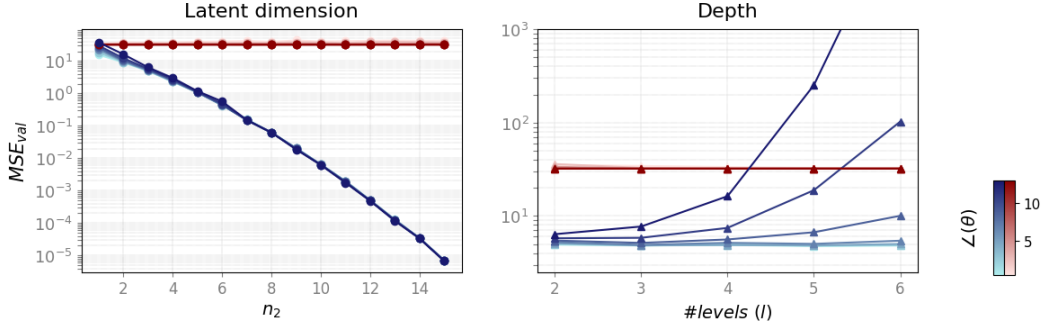


FIGURE 5. Comparison between the EYS (blue) and the He initialization (red) for the PGA case study. Models are compared in terms of the *initial* MSE over the validation set. Results refer to SAEs with HypAct_θ activation of varying sharpness $\angle = \angle(\theta)$. Left: two-level architecture $\{n_0, n_1, n_2\}$ with $n_1 = 20$ and varying latent dimension n_2 . Right: SAEs of variable depth but fixed latent dimension, obtained following the pattern $\{n_0, 65, 3\}, \{n_0, 65, 5, 3\}, \dots, \{n_0, 65, 33, 17, 9, 5, 3\}$. NB: to account for the stochasticity of the He initialization, we repeatedly initialized the architectures 100 times and only kept the trial reporting the best performance.

functions entailing a larger sharpness. Nonetheless, this is consistent with our estimates in Theorem 4.1, which ultimately suggest an exponential dependency of the error with respect to l , with a growth rate depending on $\angle(\rho_\theta) = \angle(\theta)$. In contrast, the He initialization shows no dependency on the depth of the architecture, coherently with the rationale underlying its construction [15].

We mention that similar results were obtained for the (α, β) -LeakyReLU, as well as for the other autoencoder architectures (SBAE, SOAE). We direct the interested reader to the Supplementary Materials.

During and after training. To further validate the efficacy of our strategy, we ought to demonstrate the impact of our proposed initialization on the training dynamics, this time for a fixed architecture design (cf. Table 2). In order to gain additional insights, we further enrich our analysis by including a classic autoencoder (AE) as a model baseline. We construct the latter using same skeleton adopted for the other architectures, but fixing the activation function to be the same for the encoder and the decoder. More precisely, we set $\rho_{AE}^{dec} = \rho_{AE}^{enc} = \rho_{SAE}^{-1}$, so that the decoders of AE and its symmetric counterpart share the exact same structure.

Our findings are summarized in Fig. 6. There, we see that the EYS procedure generally outperforms its standard counterpart, ensuring both a faster convergence and a smaller loss at the end of the training phase. This is particularly true in the ELS and ROD test cases, where all autoencoder architectures, SAE, SBAE and SOAE, show a substantial improvement. Things, however, are not as evident in the PGA example, where, even though the EYS-initialized architectures start the training with lower loss values when compared to their standard-initialized counterparts, it seems that this advantage does not make a difference in term of training convergence. We hypothesize that is caused by the intrinsic nature of the PGA test case, which is known to be particularly troublesome for linear reduction methods, upon which the EYS initialization is based. We mention that, while Fig. 6 only shows the results for the (α, β) -LeakyReLU with a specific choice of α and $\beta = 5/4$, analogous results were found for a different value of α , and for the HypAct_θ activation. We refrain from reporting here such results in order to minimize redundancy, and we refer the interested reader to the Supplementary Materials.

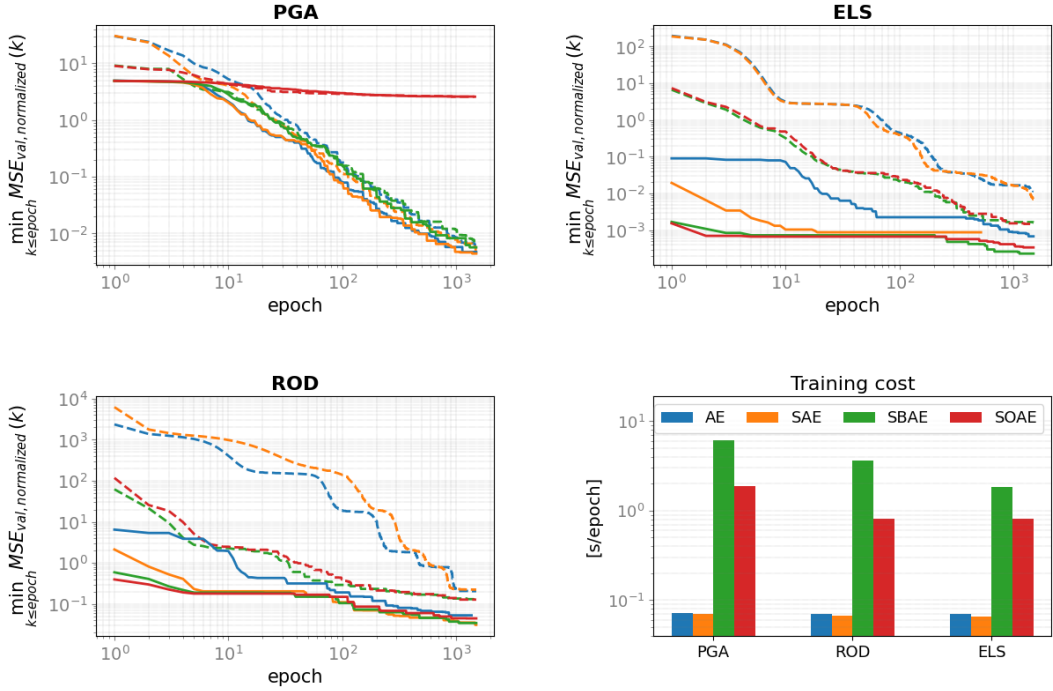


FIGURE 6. Visualization of the training dynamics and computational performance of AE, SAE, SBAE, and SOAE, for all the test cases. We compare our proposed initialization (solid line) with the literature standard (dashed line). The reported result refer to the LeakyReLU $_{\alpha,5/4}$ activation, with $\angle(\alpha) = 0.5$. The validation MSE is computed on normalized data. We refer the reader to Table 2 for the specifics on the employed architectures.

Test case	PGA	ELS	ROD
Skeleton $\{n_j\}_{j=0}^l$	{514, 64, 15, 3}	{882, 15, 10, 8}	{4347, 15, 7, 5}

TABLE 2. Architecture skeletons for the comparative study. We use the criteria proposed in [12] to set the latent dimension, while we use the error bounds developed in this paper to heuristically set the other levels' complexity.

5.3. Comparative study

Hereinafter, we compare SAE, SBAE, SOAE in terms of training efficiency and reconstruction accuracy, varying critical parameters such as the activation function and its sharpness value. Our findings are summarized in Table 3 and Fig. 6.

As a first notice, we observe that, coherently with our previous analysis, the EYS initialization proves superior for nearly all autoencoder architectures. Surprisingly, this is also true for the classical AE. This could be explained by noting that, after initialization, the decoders of AE and SAE are effectively identical. Still, after training, SAE typically achieves better performances, supporting the use of different activation functions for encoding and decoding: see Table 3. Furthermore, during training, SAE converges much faster than AE, while also entailing a smaller computational cost, Fig. 6. Such cost reduction can be explained by the fact that, at the bottleneck, AE has an additional activation compared to SAE.

What is most interesting, however, is that SBAE often outperforms SAE, meaning that the biorthogonality constraints not only do not hinder the expressivity of the architecture, but they can even prove beneficial. Since $\text{SBAE}_\rho(n_0, \dots, n_l) \subseteq \text{SAE}_\rho(n_0, \dots, n_l)$, this may indicate that the loss landscape is better behaved on the submanifold of SBAE than on the larger space of SAE, resulting in a more regular optimization. On the other hand,

these benefits come at a larger computational cost: SBAE, in fact, is the most expensive architecture to be trained, as seen in Fig. 6.

An interesting compromise is given by SOAE, which, although still more onerous than a classical unconstrained autoencoder, is effectively cheaper than a SBAE whilst still offering competitive performances. However, we also notice that the tight architecture restrictions characterizing SOAEs may occasionally prevent these architectures from generating accurate reconstructions: see PGA test case, Fig. 6. Before coming to the conclusions, it is worth making a final note about the robustness of the results presented so far. Specifically, it is important to assess whether these findings remain consistent when models are trained under different conditions—such as varying sample sizes (e.g., in low-data regimes), alternative architectural configurations (e.g., different numbers of neurons or layers), or different optimization hyperparameters (e.g., learning rates). To this end, the interested reader can find an extensive series of numerical experiments, corroborating the previous analysis across these diverse settings, in the Supplementary Materials, Sections SM3 and SM4.

			MSE	MRE
HypAct $_{\theta}$	$\angle(\theta) = 0.5$	AE	$(1.09 \pm 0.16) \times 10^2$	$(1.36 \pm 0.07) \times 10^{-3}$
		SAE	$(1.25 \pm 0.15) \times 10^2$	$(1.52 \pm 0.07) \times 10^{-3}$
		SBAE	$(5.28 \pm 0.76) \times 10^1$	$(9.54 \pm 0.50) \times 10^{-4}$
		SOAE	$(1.54 \pm 0.22) \times 10^2$	$(1.62 \pm 0.09) \times 10^{-3}$
	$\angle(\theta) = 3.0$	AE	$(2.65 \pm 0.52) \times 10^2$	$(2.09 \pm 0.13) \times 10^{-3}$
		SAE	$(1.28 \pm 0.14) \times 10^2$	$(1.56 \pm 0.06) \times 10^{-3}$
		SBAE	$(3.55 \pm 0.72) \times 10^1$	$(7.34 \pm 0.47) \times 10^{-4}$
		SOAE	$(1.27 \pm 0.20) \times 10^2$	$(1.46 \pm 0.08) \times 10^{-3}$
LeakyReLU $_{\alpha, 5/4}$	$\angle(\alpha) = 0.5$	AE	$(1.14 \pm 0.14) \times 10^2$	$(1.44 \pm 0.07) \times 10^{-3}$
		SAE	$(6.22 \pm 1.15) \times 10^1$	$(9.98 \pm 0.59) \times 10^{-4}$
		SBAE	$(7.73 \pm 1.32) \times 10^1$	$(1.13 \pm 0.06) \times 10^{-3}$
		SOAE	$(1.03 \pm 0.18) \times 10^2$	$(1.26 \pm 0.08) \times 10^{-3}$
	$\angle(\alpha) = 3.0$	AE	$(9.01 \pm 1.91) \times 10^2$	$(3.69 \pm 0.27) \times 10^{-3}$
		SAE	$(1.25 \pm 0.13) \times 10^2$	$(1.54 \pm 0.06) \times 10^{-3}$
		SBAE	$(5.19 \pm 0.73) \times 10^1$	$(9.46 \pm 0.50) \times 10^{-4}$
		SOAE	$(1.68 \pm 0.22) \times 10^2$	$(1.73 \pm 0.09) \times 10^{-3}$

TABLE 3. Average test reconstruction accuracy (with 95% confidence interval) of AE, SAE, SBAE, and SOAE for the ROD test case, when considering the EYS initialization, but varying the activation function and the sharpness parameter.

Conclusions

Within the present work, we devoted our efforts to the study of deep symmetric autoencoders. Specifically, we put forth

- *a comprehensive mathematical framework* for their classification, analysis, and implementation. In particular, we first distinguished between constrained and unconstrained architectures. Under this scope, we characterized their numerical properties, focusing on model stability and representation consistency. To complete the picture, we devised new practical strategies for their parameterization and optimization;
- *a novel initialization strategy*, termed EYS, directly stemming from newfound error bounds that connect deep symmetric autoencoders back to linear reduction methods. To the best of our knowledge, this is the first *data-driven* strategy that leverages

low-rank structures for the initialization of nonlinear symmetric autoencoders with bilipschitz activations.

In our experiments, the EYS initialization reported promising results, suggesting a potentially broader impact than the one explored in this work. Indeed, we believe that plenty of exciting research directions may originate from here, both from a theoretical and a practical standpoint. For instance, future research may focus on deepen our theoretical understanding of the relation between the EYS optimization and the optimization dynamics, possibly investigating a connection with [25]. Another interesting question is whether one could craft a suitable extension of the proposed initialization to be used for more complex and competitive architectures, such as convolutional autoencoders [27] or graph-based architectures [29]. Finally, one could exploit the proposed framework for the construction of neural surrogates for the simulation of complex physical systems, viz. DL-ROMs [13].

On top of this, future research could also be devoted at addressing some of the limitations of this work, such as the exploration of applications beyond model order reduction, like, e.g., image compression [4], the integration of probabilistic methods, such as randomized SVD, to handle very high-dimensional problems [14] and improve the scaling of the EYS initialization, or the derivation of suitable theoretical results capable of bridging model expressivity and training complexity, as in [2, 9].

Acknowledgments

SB acknowledges the support of European Union - NextGenerationEU within the Italian PNRR program (M4C2, Investment 3.3) for the PhD Scholarship “Physics-informed data augmentation for machine learning applications”. SB and NF acknowledge the support of the project *Reduced Order Modeling and Deep Learning for the real-time approximation of PDEs (DREAM)*, grant no. FIS00003154, funded by the Italian Science Fund (FIS) and by Ministero dell’Università e della Ricerca (MUR). Part of the simulations discussed in this work were performed on the HPC Cluster of the Department of Mathematics of Politecnico di Milano which was funded by MUR grant “Dipartimento di Eccellenza” 2023-2027. The authors are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM).

A. Technical proofs

Proof of Theorem 2.2. Recall that for a generic random vector $\mathbf{z} \in \mathbb{R}^m$ one has

$$\mathbb{E}\|\mathbf{z} - \mathbf{d}\|^2 \geq \mathbb{E}\|\mathbf{z} - \mathbb{E}[\mathbf{z}]\|^2 = \text{Tr}(\boldsymbol{\Sigma}[\mathbf{z}]),$$

for all deterministic vectors $\mathbf{d} \in \mathbb{R}^m$. Then, it is straightforward to prove that $\mathbf{q}_* = \mathbb{E}[\mathbf{u}]$ provides an optimal shifting. Indeed, for all $\mathbf{V} \in \mathcal{O}_{n_0, n}$ and all $\mathbf{q} \in \mathbb{R}^{n_0}$ one has

$$\mathbb{E}\|\mathbf{u} - [\mathbf{V}\mathbf{V}^\top(\mathbf{u} - \mathbf{q}) + \mathbf{q}]\|^2 = \mathbb{E}\|(\mathbf{I} - \mathbf{V}\mathbf{V}^\top)\mathbf{u} - (\mathbf{I} - \mathbf{V}\mathbf{V}^\top)\mathbf{q}\|^2.$$

Since $\mathbb{E}[(\mathbf{I} - \mathbf{V}\mathbf{V}^\top)\mathbf{u}] = (\mathbf{I} - \mathbf{V}\mathbf{V}^\top)\mathbb{E}[\mathbf{u}]$, up to setting $\mathbf{z} := (\mathbf{I} - \mathbf{V}\mathbf{V}^\top)\mathbf{u}$, it is evident that $\mathbf{q}_* = \mathbb{E}[\mathbf{u}]$ is optimal, regardless of the matrix $\mathbf{V}$. Next, we notice that $\tilde{\mathbf{u}} := \mathbf{u} - \mathbb{E}[\mathbf{u}]$ and $\mathbf{u}$ share the same covariance matrix, $\boldsymbol{\Sigma}[\tilde{\mathbf{u}}] = \boldsymbol{\Sigma}[\mathbf{u}]$. Therefore, without loss of generality, we shall continue the proof assuming that $\mathbb{E}[\mathbf{u}] \equiv 0$.

Let now $T_{\mathbf{u}} = \sum_{i=1}^r s_i \mathbf{v}_i \langle \xi_i, \cdot \rangle_{L^2(\mathcal{D})}$ be the SVD decomposition of $T_{\mathbf{u}}$, given as Theorem 2.1. Notice that, up to considering $\boldsymbol{\Sigma}[\mathbf{u}]$ as an operator from $\mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_0}$, one has $\boldsymbol{\Sigma}[\mathbf{u}] = T_{\mathbf{u}} T_{\mathbf{u}}^*$. In fact, for all $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{n_0}$,

$$\begin{aligned} \mathbf{a}^\top T_{\mathbf{u}} T_{\mathbf{u}}^* \mathbf{b} &= \mathbf{a}^\top \mathbb{E}[\mathbf{u} T_{\mathbf{u}}^* \mathbf{b}] = \mathbb{E}[\mathbf{a}^\top \mathbf{u} T_{\mathbf{u}}^* \mathbf{b}] = \langle \mathbf{a}^\top \mathbf{u}, T_{\mathbf{u}}^* \mathbf{b} \rangle_{L^2(\mathcal{D})} = \\ &= \langle T_{\mathbf{u}} \mathbf{a}^\top \mathbf{u}, \mathbf{b} \rangle_{\mathbb{R}^{n_0}} = \mathbb{E}[\mathbf{u} \mathbf{a}^\top \mathbf{u}]^\top \mathbf{b} = \mathbb{E}[(\mathbf{u}^\top \mathbf{a})(\mathbf{u}^\top \mathbf{b})] = \mathbf{a}^\top \boldsymbol{\Sigma}[\mathbf{u}] \mathbf{b}. \end{aligned}$$

It follows that, up to multiplicities, $s_i^2 = \lambda_i(\Sigma[\mathbf{u}])$ and $\mathbf{X} = [\mathbf{v}_1^*, \dots, \mathbf{v}_r^*]$, as the eigenvalues and eigenvectors of $\Sigma[\mathbf{u}]$ are nothing but the squared singular values and left singular vectors of $T_{\mathbf{u}}$.

As a next step, given any $\mathbf{V} \in \mathcal{O}_{n_0, n}$, let $P_{\mathbf{V}} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_0}$ be the projector acting as $P_{\mathbf{V}} : \mathbf{y} \mapsto \mathbf{V}\mathbf{V}^\top \mathbf{y}$. Notice that, by linearity, $T_{P_{\mathbf{V}}\mathbf{u}} = P_{\mathbf{V}}T_{\mathbf{u}}$. In particular, $\text{rk}(T_{P_{\mathbf{V}}\mathbf{u}}) \leq n$, and, by Proposition 2.1 and Theorem 2.1,

$$\mathbb{E}\|\mathbf{u} - \mathbf{V}\mathbf{V}^\top \mathbf{u}\|^2 = \|T_{\mathbf{u}} - T_{P_{\mathbf{V}}\mathbf{u}}\|_{\text{HS}}^2 \geq \sum_{i=n+1}^r \lambda_i(\Sigma[\mathbf{u}]).$$

At the same time, if $\mathbf{V} = \mathbf{V}_*$, we get $T_{P_{\mathbf{V}_*}\mathbf{u}} = P_{\mathbf{V}_*}T_{\mathbf{u}} = P_{\mathbf{V}_*} \sum_{i=1}^r s_i \mathbf{v}_i \langle \xi_i, \cdot \rangle_{L^2(\mathcal{D})} = \sum_{i=1}^r s_i P_{\mathbf{V}_*} \mathbf{v}_i \langle \xi_i, \cdot \rangle_{L^2(\mathcal{D})} = \sum_{i=1}^n s_i \mathbf{v}_i \langle \xi_i, \cdot \rangle_{L^2(\mathcal{D})}$, i.e.: $T_{P_{\mathbf{V}_*}\mathbf{u}}$ is the n -rank truncated SVD of $T_{\mathbf{u}}$. Thus, $\mathbb{E}\|\mathbf{u} - \mathbf{V}_* \mathbf{V}_*^\top \mathbf{u}\|^2 = \|T_{\mathbf{u}} - T_{P_{\mathbf{V}_*}\mathbf{u}}\|_{\text{HS}}^2 = \sum_{i=n+1}^r \lambda_i(\Sigma[\mathbf{u}])$. $\square$

Proof of Lemma 3.1. We first prove that if ρ is bilipschitz, then it is bijective. We have $|\rho(x) - \rho(y)| \geq \eta|x - y| > 0$ whenever $x \neq y$, meaning that ρ is injective. Additionally, by setting $y = 0$ and letting $x \rightarrow \pm\infty$, it is clear that $|\rho(x)| \rightarrow \infty$. Since ρ is continuous and injective, it is straightforward to conclude that either $\rho(\pm\infty) = \pm\infty$ or $\rho(\pm\infty) = \mp\infty$. In both cases, ρ ends up being surjective. Concerning the bilipschitz property of ρ^{-1} , instead, notice that we can re-write Eq. (5) as

$$\eta|\rho^{-1}(\rho(x)) - \rho^{-1}(\rho(y))| \leq |\rho(x) - \rho(y)| \leq L|\rho^{-1}(\rho(x)) - \rho^{-1}(\rho(y))| \quad \forall x, y \in \mathbb{R}.$$

As ρ is surjective, setting $w = \rho(x)$ and $z = \rho(y)$ yields $|\rho^{-1}(w) - \rho^{-1}(z)| \leq \eta^{-1}|w - z|$ and $|\rho^{-1}(w) - \rho^{-1}(z)| \geq L^{-1}|w - z|$ for all $w, z \in \mathbb{R}$, thus concluding the proof. $\square$

Proof of Proposition 3.1.

(i) Let $\Psi = \{(\mathbf{E}_j, \mathbf{D}_j, \mathbf{b}_j, \rho)\}_{j=1}^l \in \text{SBAE}_\rho(n_0, \dots, n_l)$. Notice that for each $j = 1, \dots, n_l$ one has $e_j \circ d_j = \text{Id}_{n_j}$. In fact, given any $\mathbf{h} \in \mathbb{R}^{n_j}$,

$$e_j(d_j(\mathbf{h})) = \rho(\mathbf{E}_j((\mathbf{D}_j \rho^{-1}(\mathbf{h}) + \mathbf{b}) - \mathbf{b})) = \rho(\mathbf{E}_j \mathbf{D}_j \rho^{-1}(\mathbf{h})) = \rho(\rho^{-1}(\mathbf{h})) = \mathbf{h},$$

due biorthogonality. Using associativity, it follows that $\mathcal{E}(\Psi, \mathcal{D}(\Psi, \mathbf{c})) = (e_l \circ \dots \circ e_1 \circ d_1 \circ \dots \circ d_l)(\mathbf{c}) = \mathbf{c}$ for all $\mathbf{c} \in \mathbb{R}^{n_l}$.

(ii) This is a direct consequence of (i), as it can be obtained by setting $\mathbf{c} = \mathcal{E}(\Psi, \mathbf{v})$ and applying $\mathcal{D}(\Psi, \cdot)$ to both sides of the identity.

(iii) Let $\Psi = \{(\mathbf{V}_j, \mathbf{b}_j, \rho)\}_{j=1}^l \in \text{SOAE}_\rho(n_0, \dots, n_l)$. Notice that, for every $j = 1, \dots, n_l$, the maps $\mathbf{c} \mapsto \mathbf{V}_j \mathbf{c} + \mathbf{b}_j$ and $\mathbf{h} \mapsto \mathbf{V}_j^\top (\mathbf{h} - \mathbf{b}_j)$ are both affine and 1-Lipschitz. By composition, it follows that $\text{Lip}(e_j) \leq \text{Lip}(\rho)$ and $\text{Lip}(d_j) \leq \text{Lip}(\rho^{-1})$. Since $\text{Lip}(\mathcal{E}(\Psi, \cdot)) \leq \prod_{j=1}^l \text{Lip}(e_j)$ and $\text{Lip}(\mathcal{D}(\Psi, \cdot)) \leq \prod_{j=1}^l \text{Lip}(d_j)$, the conclusion follows. $\square$

B. Complements on the He initialization

Hereby, we derive an extension of the so-called *He initialization* by generalizing the construction proposed in [15] to more complex bilipschitz activations. To this end, let f be a bilipschitz function with $f(0) = 0$. The generic forward pass of a layer mapping $\mathbb{R}^n \rightarrow \mathbb{R}^q$ reads $\mathbf{x} = \mathbf{W}f(\mathbf{z}) + \mathbf{b}$, with $\mathbf{x}, \mathbf{b} \in \mathbb{R}^q$, $\mathbf{W} \in \mathbb{R}^{q \times n}$ and $\mathbf{z} \in \mathbb{R}^n$. Following [15], we shall set $\mathbf{b} \equiv \mathbf{0}$ and assume that

- (i) $\mathbf{z} = [z_1, \dots, z_n]^\top$ is a random vector with i.i.d. components $z_j \stackrel{\text{i.i.d.}}{\sim} \omega$;
- (ii) ω is centered and has finite second moment;
- (iii) $\mathbf{W} = (w_{ij})_{i,j}$ is drawn at random according to $w_{ij} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$.

The idea is to pick σ such that the variance of $\mathbf{z}$ transfers to $\mathbf{x}$ in a controlled manner. In this respect, we notice that if $\mathbf{x} = [x_1, \dots, x_q]^\top$ then, for all $i = 1, \dots, q$,

$$\text{Var}(x_i) = \text{Var}\left(\sum_{j=1}^n w_{ij}f(z_j)\right) = \sum_{j=1}^n \text{Var}(w_{ij}f(z_j)) = n\sigma^2\mathbb{E}|f(z_1)|^2, \quad (19)$$

due independence and identical distribution. Indeed,

$$\begin{aligned} \text{Var}(w_{ij}f(z_j)) &= \mathbb{E}[w_{ij}^2 f(z_j)^2] - \mathbb{E}[w_{ij}f(z_j)]^2 = \\ &= \mathbb{E}|w_{ij}|^2 \mathbb{E}|f(z_j)|^2 - \mathbb{E}[w_{ij}]^2 \mathbb{E}[f(z_j)]^2 = \sigma^2 \mathbb{E}|f(z_j)|^2, \end{aligned}$$

since $\mathbb{E}[w_{i,j}] = 0$. Thus, we are left to study $\mathbb{E}|f(z_1)|^2 = \int_{-\infty}^{+\infty} |f(z)|^2 \omega(z) dz$, which can be bounded from above and below, using the stability properties of f . Specifically, using the fact that $f(0) = 0$, and $\text{Lip}(f^{-1})^{-1}|z| \leq |f(z)| \leq \text{Lip}(f)|z|$ for all $z \in \mathbb{R}$, one has

$$\text{Var}(z_1) \text{Lip}(f^{-1})^{-2} \leq \int_{-\infty}^{+\infty} |f(z)|^2 \omega(z) dz \leq \text{Var}(z_1) \text{Lip}(f)^2.$$

Combining the above with (19) yields

$$n\sigma^2 \text{Var}(z_1) \text{Lip}(f^{-1})^{-2} \leq \text{Var}(x_i) \leq n\sigma^2 \text{Var}(z_1) \text{Lip}(f)^2. \quad (20)$$

In the wake of [15], we select σ such that $\text{Var}(x_i) \approx \text{Var}(z_1)$. Heuristically, we achieve so by taking the harmonic mean between the lower and upper bound values of σ derived from (20), which results in $\sigma^2 = 2[n(\text{Lip}(f^{-1})^{-2} + \text{Lip}(f)^2)]^{-1}$.

Note that, in the case of the LeakyReLU, this choice coincides exactly with the initialization proposed in [15].

C. Complements on biorthogonal matrices

Let $q, r \in \mathbb{N}_+$ with $r \leq q$. Set $d := \min\{r, q - r\}$. As discussed in Section 3, we can parametrize a pair of biorthogonal matrices $\Xi, \Theta \in \mathbb{R}^{q \times r}$ using a list of five matrices $\mathbf{X}, \mathbf{U}, \mathbf{S}, \mathbf{Z}, \mathbf{Q}$ where: $\mathbf{X} \in \mathbb{R}^{q \times (r+d)}$ is orthonormal; $\mathbf{U}, \mathbf{Z} \in \mathbb{R}^{r \times r}$ are orthonormal; $\mathbf{S} \in \mathbb{R}^{r \times r}$ is diagonal with strictly positive entries; $\mathbf{Q} \in \mathbb{R}^{d \times r}$ is arbitrary. We recall that such parametrization is given by

$$\Xi := \mathbf{X} \begin{bmatrix} \mathbf{USZ}^\top \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \Theta := \mathbf{X} \begin{bmatrix} \mathbf{US}^{-1}\mathbf{Z}^\top \\ \mathbf{Q} \end{bmatrix}.$$

To see that the biorthogonality constraint is fulfilled, simply notice that

$$\begin{aligned}\Xi^\top \Theta &= [\mathbf{Z}\mathbf{S}\mathbf{U}^\top \quad \mathbf{0}] \mathbf{X}^\top \mathbf{X} \begin{bmatrix} \mathbf{U}\mathbf{S}^{-1}\mathbf{Z}^\top \\ \mathbf{Q} \end{bmatrix} = [\mathbf{Z}\mathbf{S}\mathbf{U}^\top \quad \mathbf{0}] \begin{bmatrix} \mathbf{U}\mathbf{S}^{-1}\mathbf{Z}^\top \\ \mathbf{Q} \end{bmatrix} = \\ &= \mathbf{Z}\mathbf{S}\mathbf{U}^\top \mathbf{U}\mathbf{S}^{-1}\mathbf{Z}^\top = \mathbf{I}_r.\end{aligned}$$

In fact, since $\mathbf{Z}$ is a square matrix, one has $\mathbf{Z}^\top \mathbf{Z} = \mathbf{I}_r \implies \mathbf{Z}\mathbf{Z}^\top = \mathbf{I}_r$.

Such parametrization is surjective, meaning that any pair of biorthogonal matrices can be written in such a way. To see this, assume Ξ and Θ are given. Let $\mathbf{X}_1 \in \mathbb{R}^{q \times r}$ be an orthonormal matrix spanning the same r -dimensional subspace of Ξ . Construct $\mathbf{X} \in \mathbb{R}^{q \times (r+d)}$ by adding d columns to $\mathbf{X}_1$ such that $\mathbf{X}$ is orthonormal and

$$\text{span}(\mathbf{X}) \supseteq \text{span}(\Xi) \cup \text{span}(\Theta).$$

Evidently, this is always possible. Indeed, if $d = r$, we simply notice that

$$\dim(\text{span}(\Xi) \cup \text{span}(\Theta)) \leq r + r = r + d.$$

Conversely, if $d = q - r$, then $d + r = q$, meaning that $\text{span}(\mathbf{X}) = \mathbb{R}^q$; the inclusion is then trivial as $\text{span}(\Xi), \text{span}(\Theta) \subseteq \mathbb{R}^q$. With such construction, one then has

$$\Xi := \mathbf{X} \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \Theta := \mathbf{X} \begin{bmatrix} \mathbf{C} \\ \mathbf{Q} \end{bmatrix},$$

for some invertible matrices $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{r \times r}$ and $\mathbf{Q} \in \mathbb{R}^{d \times r}$. Since $\mathbf{I}_r = \Xi^\top \Theta = \mathbf{B}^\top \mathbf{C}$, it follows that $\mathbf{C} = (\mathbf{B}^\top)^{-1}$. Introducing an SVD decomposition for $\mathbf{B}$ yields the desired representation.

As a final note, we mention that this parametrization is redundant, as the manifold of biorthogonal matrices is $2qr - r^2$ -dimensional [26], whereas the proposed parametrization features

$$q(r+d) - \frac{1}{2}(r+d)(r+d+1) + r^2 - r + r + dr = \frac{1}{2}(r+d)(2q+r-d-1)$$

parameters. In particular, if $d = r$, then our approach entails a total of $r(r-1) \geq 0$ “extra” parameters. Conversely, the latter are $(\frac{q}{2} - r)^2 + \frac{q}{2}(\frac{q}{2} - 1) \geq 0$ whenever $d = q - r$.

D. Supplementary material

D.1. Complementary results on the EYS initialization

The following Fig. 7 refers to the study conducted on the PGA test case concerning the performances of the EYS initialization *before training*. Results refer to different architectures and activations of varying sharpness, as indicated below; other specifics are the same as the ones of Fig. 5 in the main text.

Results for SBAE and SOAE are reported together as, for these architectures, both the EYS initialization and the literature standard coincide. We recall, in fact, that we considered the approach proposed in [26] as our benchmark: there, the authors propose to initialize SBAE architectures with $\mathbf{E}_j = \mathbf{D}_j^\top$. Consequently, right after initialization, the SBAE module is equivalent to a SOAE. This is also true in terms of the underlying parametrization, as we use a procedure analogous to the one discussed in Section 4.2 (ii) to initialize the matrices $\tilde{\mathbf{X}}_j, \tilde{\mathbf{Y}}_j, \tilde{\mathbf{Z}}_j, \mathbf{Q}_j$ and the vector $\mathbf{s}_j$.

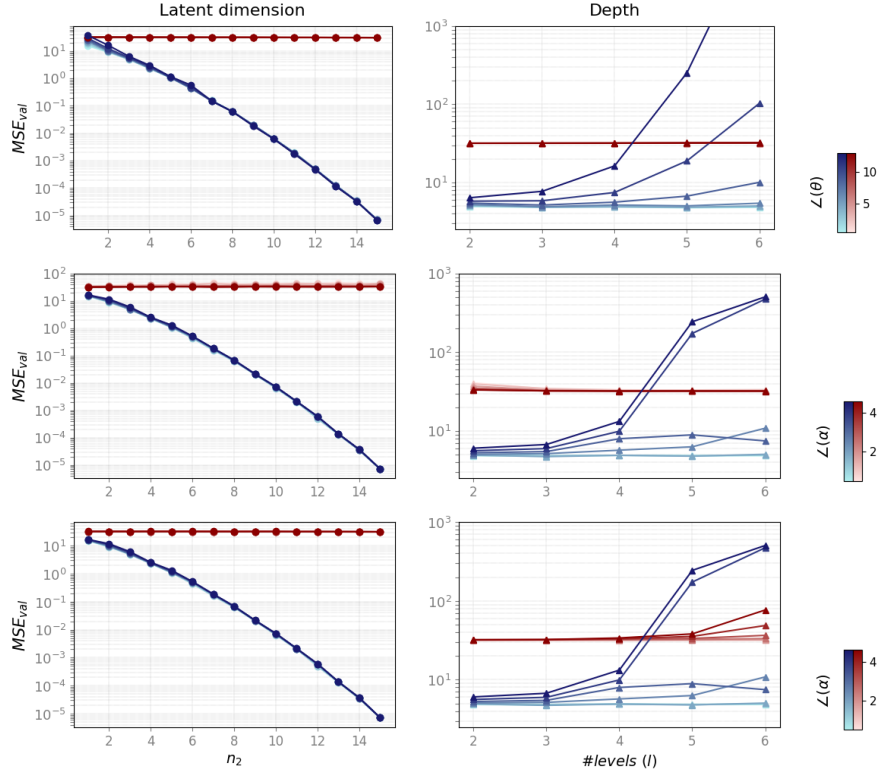


FIGURE 7. Comparison between the EYS (blue) and the standard random initialization (red) for the PGA case study. Top: SOAEs with HypAct_θ . Middle: SAEs with $\text{LeakyReLU}_{\alpha,5/4}$. Bottom: SOAEs with $\text{LeakyReLU}_{\alpha,5/4}$.

D.2. Complementary results on the comparative study

We report below the complementary results for the comparative study presented in Section 5. Specifically, Tables 4-5 read as Table 3 in the paper but refer to the PGA and ELS case studies, respectively. Conversely, Figures 8, 9, and 10 are analogous to Fig. 6 in the paper but discuss different choices of ρ and its sharpness value.

		MSE		MRE
HypAct $_\theta$	$\angle(\theta) = 0.5$	AE	$(7.55 \pm 1.90) \times 10^{-3}$	$(1.30 \pm 0.07) \times 10^{-2}$
		SAE	$(7.13 \pm 1.23) \times 10^{-3}$	$(1.33 \pm 0.06) \times 10^{-2}$
		SBAE	$(1.21 \pm 0.20) \times 10^{-2}$	$(1.75 \pm 0.07) \times 10^{-2}$
		SOAE	$(3.27 \pm 0.23) \times 10^0$	$(3.02 \pm 0.09) \times 10^{-1}$
	$\angle(\theta) = 3.0$	AE	$(1.81 \pm 0.41) \times 10^{-3}$	$(6.53 \pm 0.32) \times 10^{-3}$
		SAE	$(1.79 \pm 0.35) \times 10^{-3}$	$(6.60 \pm 0.31) \times 10^{-3}$
		SBAE	$(5.23 \pm 0.92) \times 10^{-3}$	$(1.17 \pm 0.05) \times 10^{-2}$
		SOAE	$(1.57 \pm 0.28) \times 10^{-2}$	$(1.99 \pm 0.09) \times 10^{-2}$
LeakyReLU $_{\alpha,5/4}$	$\angle(\alpha) = 0.5$	AE	$(7.64 \pm 2.62) \times 10^{-3}$	$(1.22 \pm 0.08) \times 10^{-2}$
		SAE	$(5.84 \pm 1.56) \times 10^{-3}$	$(1.14 \pm 0.06) \times 10^{-2}$
		SBAE	$(7.59 \pm 1.81) \times 10^{-3}$	$(1.35 \pm 0.07) \times 10^{-2}$
		SOAE	$(2.78 \pm 0.19) \times 10^0$	$(2.79 \pm 0.08) \times 10^{-1}$
	$\angle(\alpha) = 3.0$	AE	$(4.42 \pm 0.75) \times 10^{-3}$	$(1.05 \pm 0.05) \times 10^{-2}$
		SAE	$(2.67 \pm 0.36) \times 10^{-3}$	$(8.18 \pm 0.35) \times 10^{-3}$
		SBAE	$(7.43 \pm 0.88) \times 10^{-3}$	$(1.42 \pm 0.05) \times 10^{-2}$
		SOAE	$(2.09 \pm 0.26) \times 10^{-2}$	$(2.35 \pm 0.09) \times 10^{-2}$

TABLE 4. PGA test case, with EYS initialization (see Section D.2).

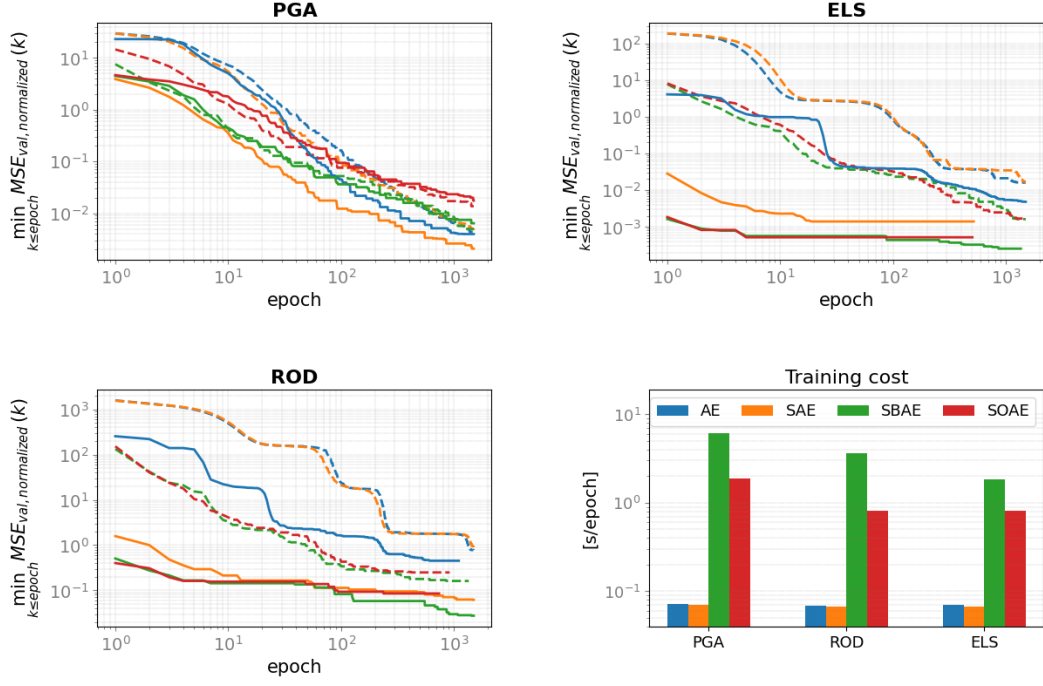


FIGURE 8. Visualization of the training dynamics, with $\text{LeakyReLU}_{\alpha,5/4}$, with α such that $\angle(\alpha) = 3.0$.

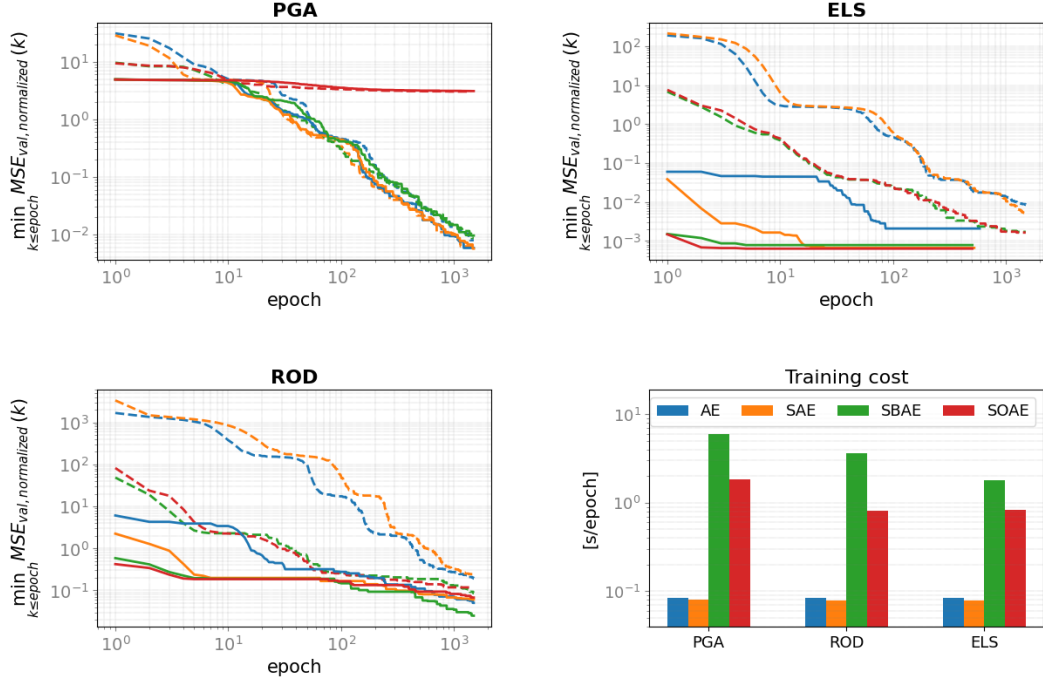


FIGURE 9. Visualization of the training dynamics, with $\text{HypAct}(\theta)$, with θ such that $\angle(\theta) = 0.5$.

D.3. Exploration of the hyperparameters' space

To complement the analysis reported in Section 5 of the main text, we report below some additional results concerning the impact of network hyperparameters on the reconstruction accuracy of EYS-initialized and standard-initialized *deep symmetric autoencoders*. These include design hyperparameters—number of layers, neurons, etc.—and training hyperparameters—learning rate, decay schedulers, etc. To limit the otherwise excessive computational demand, this analysis was limited to SAE architectures, which are by far the most efficient to train. As anticipated, aside from the initialization procedure, in our hyperparameter sweep we vary:

- the activation function in $\{\text{LeakyReLU}_{\alpha,5}, \text{HypAct}_{\theta}\}$ with α, θ such that $\angle(\cdot) \in \{0.5, 3\}$;
- the learning rate, $lr \in \{3 \times 10^{-4}, 6 \times 10^{-4}, 1 \times 10^{-3}\}$;
- the learning rate scheduler in $\{\text{None}, \text{CosineAnnealing}, \text{OneCycle}\}$, with the `OneCycle` scheduler featuring a maximum learning rate equal to 10^{-3} .
- the architecture skeletons, according to problem-dependent set of choices, see Table 6.

We emphasize that each run is repeated 3 times to increase the robustness of the analysis. Our findings are summarized in 11, where we report and compare the validation performance of all trained architectures. The hyperparameters' space exploration shows that the EYS initialization method consistently outperforms its standard counterpart (cf. Fig. 11, left column), quite independently of the choice of the design and training hyperparameters, thereby demonstrating the robustness of the proposed initialization strategy. At the same time, there also clear differences in between the three test cases: while EYS is basically always advantageous when considering the ROD and ELS examples, the improvement rate drops to 84% in the PGA test case. Still, we notice that this is consistent with our original findings, which suggested that the advantages of the EYS initialization seem to be tied to the linear reducibility of the problem at hand: cf. §5.2 in the main text.

D.4. Sensitivity analysis with respect to the training sample size

Within this Section, we aim to further study the robustness of our analysis with respect to varying training-validation sample sizes. To this end, our data generation and training setup is set as follows,

			MSE	MRE
HypAct $_{\theta}$	$\angle(\theta) = 0.5$	AE	$(7.86 \pm 0.48) \times 10^{-3}$	$(1.42 \pm 0.06) \times 10^{-2}$
		SAE	$(2.67 \pm 0.21) \times 10^{-3}$	$(7.93 \pm 0.25) \times 10^{-3}$
		SBAE	$(3.10 \pm 0.28) \times 10^{-3}$	$(8.26 \pm 0.29) \times 10^{-3}$
		SOAE	$(2.62 \pm 0.22) \times 10^{-3}$	$(7.70 \pm 0.26) \times 10^{-3}$
	$\angle(\theta) = 3.0$	AE	$(1.24 \pm 0.47) \times 10^{-2}$	$(1.46 \pm 0.06) \times 10^{-2}$
		SAE	$(2.95 \pm 0.23) \times 10^{-3}$	$(8.15 \pm 0.27) \times 10^{-3}$
		SBAE	$(2.00 \pm 0.22) \times 10^{-3}$	$(6.44 \pm 0.23) \times 10^{-3}$
		SOAE	$(2.32 \pm 0.21) \times 10^{-3}$	$(7.10 \pm 0.25) \times 10^{-3}$
LeakyReLU $_{\alpha,5/4}$	$\angle(\alpha) = 0.5$	AE	$(2.83 \pm 0.20) \times 10^{-3}$	$(8.18 \pm 0.33) \times 10^{-3}$
		SAE	$(3.48 \pm 0.25) \times 10^{-3}$	$(8.97 \pm 0.26) \times 10^{-3}$
		SBAE	$(8.80 \pm 0.65) \times 10^{-4}$	$(4.61 \pm 0.16) \times 10^{-3}$
		SOAE	$(1.36 \pm 0.10) \times 10^{-3}$	$(5.87 \pm 0.26) \times 10^{-3}$
	$\angle(\alpha) = 3.0$	AE	$(1.89 \pm 0.56) \times 10^{-2}$	$(1.78 \pm 0.08) \times 10^{-2}$
		SAE	$(5.43 \pm 0.42) \times 10^{-3}$	$(1.11 \pm 0.03) \times 10^{-2}$
		SBAE	$(9.03 \pm 1.10) \times 10^{-4}$	$(4.38 \pm 0.16) \times 10^{-3}$
		SOAE	$(2.16 \pm 0.19) \times 10^{-3}$	$(6.89 \pm 0.23) \times 10^{-3}$

TABLE 5. ELS test case, with EYS initialization (see Section D.2).

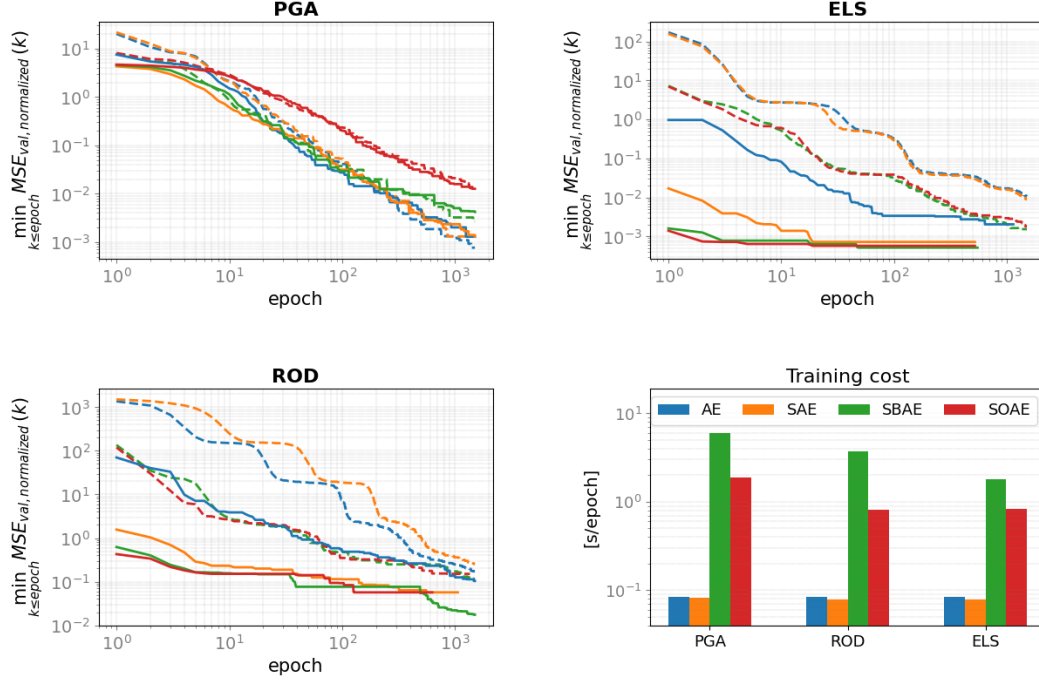


FIGURE 10. Visualization of the training dynamics, with $\text{HypAct}(\theta)$, with θ such that $\angle(\theta) = 3.0$.

PGA	ROD	ELS
[514, 64, 3]	[4347, 15, 5]	[882, 8, 5]
[514, 64, 15, 3]	[4347, 15, 7, 5]	[882, 15, 10, 8]
[514, 64, 15, 7, 3]	[4347, 20, 7]	[882, 15, 10]
[514, 32, 15, 3]	[4347, 30, 20, 10, 5]	[882, 10, 8]
[514, 16, 10, 3]	[4347, 30, 10, 5]	[882, 30, 15, 10, 8]
[514, 64, 15, 5]	[4347, 20, 10, 7]	[882, 30, 10]
[514, 64, 18, 7]	[4347, 30, 20, 15, 10, 5]	[882, 15, 9, 6]

TABLE 6. Architecture skeletons for the hyperparameter sweep of Section D.4.

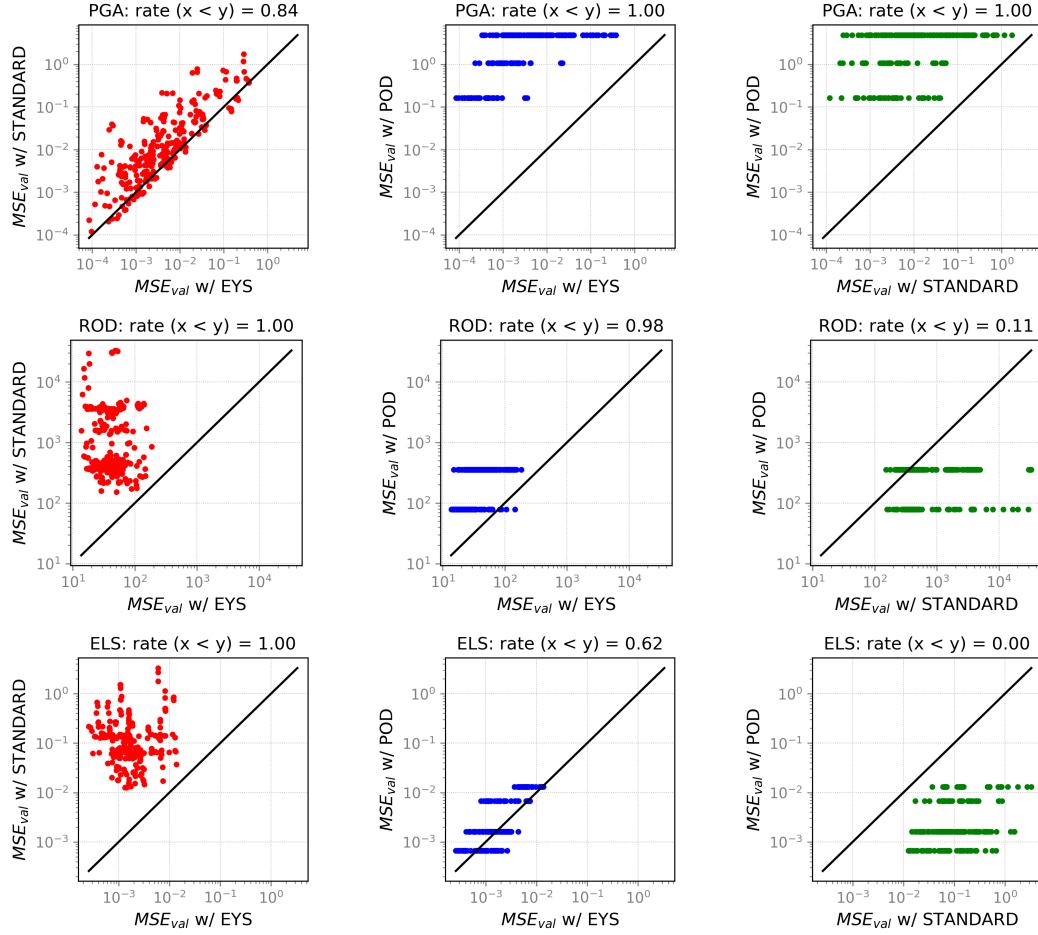


FIGURE 11. For any choice of the design and training hyperparameters, we compare the validation MSE of the best (among 3 runs) EYS- and standard- initialized architectures after training (first column). The POD baseline having the same latent dimension as the trained SAE is reported for reference (second and third columns).

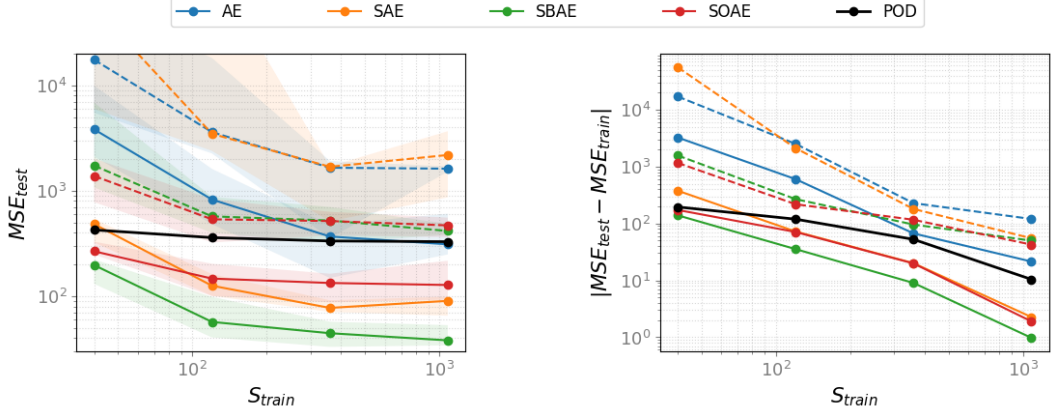


FIGURE 12. We report the results of the analysis of the sample size sensitivity study, using continuous and dashed lines to indicate architectures initialized with EYS and the standard strategy, respectively. The POD baseline is added for reference. We report the median value (among the 5 different runs), along with min-max bands (whenever they do not degrade the visualization).

- we select different sample sizes, $S_{\text{train+val}} \in \{50, 150, 450, 1350\}$, for model training and validation. The training-validation split is set to 80:20. As usual, we use the training data for the computation and minimization the loss function, whereas we rely on the validation set for checkpoint selection and early stopping;
- we perform 5 independent runs, obtained by re-generating the training and the validation data (each run being done independently), for all architectures and all sample sizes mentioned above;
- we employ a fixed test set consisting of 500 independent simulations.

To limit the otherwise excessive computational burden of the proposed analysis, for this numerical experiment, we consider only the ROD test case, using LeakyReLU_{5,5/4} as activation function. We emphasize that the other hyperparameters coincide with the ones reported in Numerical Experiments Section of the main text. We showcase the results of our analysis in Fig. 12, which shows that:

- (i) when initialized with the EYS strategy, all symmetric autoencoders show a mild decay of the test error MSE_{test} , suggesting that properly initialized symmetric autoencoders can readily benefit from an increasing amount of data samples;
- (ii) the test error of standard-initialized symmetric autoencoders seems to saturate earlier and with a larger saturation value, when compared to their EYS-initialized counterparts, corroborating again the importance of model initialization;
- (iii) especially when the number of samples is quite small, the EYS-initialized architectures outperform their standard-initialized counterparts in terms of generalization gap, here defined as $|MSE_{\text{train}} - MSE_{\text{test}}|$, thereby hinting that proper initialization strategies can be effective in enhancing generalization performance of symmetric autoencoders.

References

- [1] Najmeh Abiri, Björn Linse, Patrik Edén, and Mattias Ohlsson. Establishing strong imputation performance of a denoising autoencoder in a wide range of missing data problems. *Neurocomputing*, 365:137–146, 2019.

- [2] Ben Adcock and Nick Dexter. The gap between theory and practice in function approximation with deep neural networks. *SIAM Journal on Mathematics of Data Science*, 3(2):624–655, 2021.
- [3] Muhammad Al-Digeil, Yuri Grinberg, Daniele Melati, Jens H Schmid, Pavel Cheben, Siegfried Janz, and Dan-Xia Xu. Pca-enhanced autoencoders for nonlinear dimensionality reduction in low data regimes. In *Canadian AI*, 2023.
- [4] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations*, 2017.
- [5] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000.
- [6] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau. Autoencoder-based network anomaly detection. In *2018 Wireless telecommunications symposium (WTS)*, pages 1–5. IEEE, 2018.
- [7] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In Dekai Wu, Marine Carpuat, Xavier Carreras, and Eva Maria Vecchi, editors, *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics.
- [8] John B Conway. *A course in functional analysis*, volume 96. Springer, New York, 2007.
- [9] Maarten V de Hoop, Nikola B Kovachki, Nicholas H Nelsen, and Andrew M Stuart. Convergence rates for learning linear operators from noisy data. *SIAM/ASA Journal on Uncertainty Quantification*, 11(2):480–513, 2023.
- [10] Richard M Dudley. *Real analysis and probability*. Chapman and Hall/CRC, 2018.
- [11] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, Sep 1936.
- [12] Nicola Franco, Andrea Manzoni, and Paolo Zunino. A deep learning approach to Reduced Order Modelling of parameter dependent partial differential equations. *Mathematics of Computation*, 92(340):483–524, 2023.
- [13] Stefania Fresca, Luca Dede’, and Andrea Manzoni. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. *Journal of Scientific Computing*, 87:1–36, 2021.
- [14] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [16] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [17] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and Helmholtz free energy. In J. Cowan, G. Tesauero, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1993.

- [18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [19] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, November 2019.
- [20] M.A. Kramer. Autoassociative neural networks. *Computers & Chemical Engineering*, 16(4):313–328, 1992.
- [21] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.
- [22] Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error estimates for deepnets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022.
- [23] M. Loeve. *Probability Theory II*. Graduate Texts in Mathematics. Springer, 1978.
- [24] Nikolaj T. Mücke, Sander M. Bohté, and Cornelis W. Oosterlee. Reduced order modeling for parameterized time-dependent PDEs using spatially and memory aware deep learning. *Journal of Computational Science*, 53:101408, 2021.
- [25] Thanh V. Nguyen, Raymond K. W. Wong, and Chinmay Hegde. Benefits of jointly training autoencoders: An improved neural tangent kernel analysis. *IEEE Transactions on Information Theory*, 67(7):4669–4692, 2021.
- [26] Samuel E. Otto, Gregory R. Macchio, and Clarence W. Rowley. Learning nonlinear projections for reduced-order modeling of dynamical systems using constrained autoencoders. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(11):113130, 11 2023.
- [27] Pranshu Pant, Ruchit Doshi, Pranav Bahl, and Amir Barati Farimani. Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations. *Physics of Fluids*, 33(10):107101, 10 2021.
- [28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [29] Federico Pichi, Beatriz Moya, and Jan S Hesthaven. A graph convolutional autoencoder approach to model order reduction for parametrized PDEs. *Journal of Computational Physics*, 501:112762, 2024.
- [30] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, Cham, 2015.
- [31] Michael Reed and Barry Simon. *Methods of modern mathematical physics. vol. 1. Functional analysis*. Academic Press, San Diego, 1980.
- [32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986.
- [33] Erhard Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. *Mathematische Annalen*, 63(4):433–476, Dec 1907.

- [34] Mathias Seuret, Michele Alberti, Marcus Liwicki, and Rolf Ingold. PCA-Initialized deep neural networks applied to document image analysis. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 877–882, 2017.
- [35] Yunfei Teng and Anna Choromanska. Invertible Autoencoder for Domain Adaptation. *Computation*, 7(2), 2019.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 73/2026** Corti, M.
A weighted polygonal discontinuous Galerkin method for hierarchically coupled reaction–diffusion systems with cubic interactions
- 72/2026** Lorenzon, G.; Regazzoni, F.
Optimal Transport Dropout for Structured Predictive Uncertainty
- 71/2026** Radisic, I.; Regazzoni, F.; Dede', L.; Quarteroni, A.
A mathematical model for irreversible damage of the collagen scaffold in the myocardium
- 70/2026** Micheletti, S.
Longitudinal versus quasi-P waves in weakly anisotropic elasticity: operator comparison and long-time dephasing
- 68/2026** Patanè, G.; Greven, S.; Menafoglio, A.
Random mixtures in Bayes Hilbert spaces
- 69/2026** Zecchi, A. A.; Ferro, N.; Perotto, S.
Complexity-controlled QUBO formulation for topology optimization via anisotropic mesh adaptation
- 67/2026** Ciardulli, S.; Fontana, N.; Vantini, S.; Ieva F.
Generalized propensity score weighting for functional causal inference framework
- 66/2026** Antonietti, P.F.; Bonizzoni, F.; Corti, M.; De March, N.; Di Noto, S.; Regazzoni, F.
A stability-preserving polytopal discontinuous Galerkin method for the Fisher-Kolmogorov model with applications to neurodegenerative diseases
- 65/2026** De Sanctis, M.F.; Arnone, E.; Ieva, F.; Sangalli, L.M.
Modeling group heterogeneity in spatio-temporal data via physics-informed regression
- 64/2026** Radisic, I.; Tirotta, R.; Zingaro, A.; Pagani, S.; Dede', L.
Physics-Informed Neural Networks for the High-Resolution Reconstruction of Flow Measurement Indicators in Fluid Dynamics