

Rank Reduction AutoEncoders for Mechanical Design: Advancing Novel and Efficient Data-Driven Topology Optimization

Ismael Ben-Yelun^{a,*}, Mohammed El Fallaki Idrissi^b, Jad Mounayer^c, Sebastian Rodriguez^d, Francisco Chinesta^{b,c,d}

^a*E.T.S. de Ingeniería Aeronáutica y del Espacio, Universidad Politécnica de Madrid, Pza. Cardenal Cisneros 3, Madrid, 28040, Spain*

^b*ENSAM Institute of Technology PIMM, ESI/Keysight Chair CNRS@CREATE, Singapore, 151 Bd de l'Hôpital, Paris, 75013, France*

^c*ENSAM Institute of Technology PIMM, SKF Chair, 151 Bd de l'Hôpital, Paris, 75013, France*

^d*ENSAM Institute of Technology PIMM, RTE Chair, 151 Bd de l'Hôpital, Paris, 75013, France*

Abstract

This work presents a data-driven framework for fast forward and inverse analysis in topology optimization (TO) by combining Rank Reduction Autoencoders (RRAEs) with neural latent-space mappings. The methodology targets the efficient approximation of the relationship between optimized geometries and their corresponding mechanical responses or Quantity of Interest (QoI), with a particular focus on compliance-minimized linear elastic structures. High-dimensional TO results are first compressed using RRAEs, which encode the data into a low-rank approximation via Singular Value Decomposition (SVD), obtained in this sense the most important features that approximate the data. Separate RRAE models are trained for geometry and for different types of QoIs, including scalar metrics, one-dimensional stress fields, and full two-dimensional von Mises stress distributions. The resulting low-dimensional latent coefficients of the latent space are then related through multilayer perceptrons to address both direct problems – predicting structural responses from geometry – and inverse problems—recovering geometries from prescribed performance targets. The proposed approach is demonstrated on a benchmark TO problem based on a half MBB beam, using datasets generated via density-based Solid Isotropic Material with Penalization (SIMP) optimization. Numerical results show that the framework enables accurate and computationally efficient surrogate models, with increasing robustness and fidelity as richer QoIs are considered. The methodology also provides a foundation for generative mechanical design by enabling the synthesis of new geometries and responses through latent-space exploration.

Keywords: AutoEncoders, Topology Optimization, Model Order Reduction, Generative Design, Rank Reduction AutoEncoder

1. Introduction

Topology optimization (TO) has become a central tool in computational mechanics for the automated synthesis of high-performance structures under prescribed loading and boundary conditions. Since the seminal works on homogenization-based methods and density-based approaches, TO has been successfully applied to a wide range of engineering problems, including lightweight structural design [1, 2, 3, 4], compliant mechanisms [5, 6, 7], and multi-physics systems [8, 9, 10]. Among the available formulations, the Solid Isotropic Material with Penalization (SIMP) method [11] remains one of the most widely adopted techniques due to its conceptual simplicity and compatibility with Finite Element solvers [12, 13].

Despite its success, TO remains computationally demanding, particularly when repeated analyses are required, such as in design space exploration, uncertainty quantification, real-time decision

*Corresponding author

Email addresses: i.binsenser@upm.es (Ismael Ben-Yelun), mohammed.el_fallaki_idrissi@ensam.eu (Mohammed El Fallaki Idrissi), jad.MOUNAYER@ensam.eu (Jad Mounayer), sebastian.rodriguez_iturra@ensam.eu (Sebastian Rodriguez), Francisco.CHINESTA@ensam.eu (Francisco Chinesta)

making, or inverse design. Each optimization run involves a large number of Finite Element problem resolutions, and the resulting optimized geometries are typically represented by high-dimensional density fields. This high computational cost has motivated the development of surrogate models and reduced-order representations capable of approximating optimized designs and their mechanical responses at a fraction of the original cost.

Recent advances in machine learning, and deep learning in particular [14], have opened new opportunities for data-driven modeling in TO. Neural networks have been used to accelerate optimization, predict optimized layouts directly from problem parameters, and approximate solution fields such as displacements or stresses. Convolutional neural networks (CNNs) have been especially popular due to their ability to process grid-based density fields. However, purely black-box approaches often suffer from limited interpretability, poor extrapolation, and difficulties in handling inverse problems, where multiple geometries may correspond to the same performance metric. An alternative and increasingly promising strategy consists of combining machine learning with reduced-order modeling techniques. Autoencoders, and more generally nonlinear manifold learning methods, provide a way to compress high-dimensional TO data into low-dimensional latent representations that capture the dominant geometric and physical features of the designs. These latent spaces can then be exploited for fast prediction, interpolation, and inverse analysis. Nevertheless, standard autoencoders often limit expressiveness or introduce redundancy and overfitting.

In this context, Rank Reduction Autoencoders (RRAEs) [15] offer a flexible and physically motivated alternative by imposing that the latent space of an autoencoder should be expressed as low-rank approximation by means of a truncated Singular Value Decomposition (SVD) method [16, 17, 18]. This architecture allows the model to identify the most relevant modes in the latent space while maintaining computational efficiency and robustness. The RRAE has demonstrated strong performance across a wide range of applications. Notable extensions include its variational autoencoder formulation, termed the VRRAE [19]. The RRAE has also been successfully applied to Generative Design (GD) and optimization of homogenized problems in composite materials [20], as well as to GD of parametric surrogate models [21]. More recently, the RRAE has been combined with DeepONets for heat transfer applications [22], employed in the modeling of complex dynamical systems [23] and the field of Structural Health Monitoring (SHM) for the detection of damage in plate structures using Lamb waves [24].

Therefore, the present work proposes a unified data-driven framework based on RRAEs to model the relationship between optimized geometries and their mechanical responses. Separate RRAE models are constructed for geometry and for different QoIs, including scalar stress measures, one-dimensional stress fields, and full two-dimensional von Mises stress distributions. The resulting low-dimensional latent coefficients are linked through neural regressors to address both direct problems—predicting responses from geometries—and inverse problems—reconstructing geometries from prescribed performance targets. It should be emphasized that the inverse problem is inherently ill-posed. Nevertheless, rather than applying a Newton-based approach to the forward problem, this work explores the construction of an inverse regression, fully acknowledging the risks and limitations associated with such a procedure. Through a detailed numerical study on a benchmark TO problem, the impact of the chosen QoI on model accuracy, robustness, and invertibility is systematically investigated.

The main contributions of this work are threefold: (i) the introduction of RRAEs as an efficient and adaptive dimensionality reduction tool for TO data; (ii) a comparative analysis of forward and inverse surrogate modeling performance as a function of the information content of the QoI; and (iii) the demonstration of a latent-space-based framework that naturally enables GD and rapid exploration of optimized structural layouts.

The paper is structured as follows. Section 2 provides an overview of the TO problem considered in this work for the generation of the dataset. Section 3 introduces the main ingredients of the RRAE and how the architecture works. Following Section 4 presents how RRAE can be used to accelerate the topology optimization procedure and its use to perform inverse analysis. The performance of the proposed architecture in approximating as well as performing inverse analysis is demonstrated in Section 5 by considering a half MBB beam to perform TO and by considering as QoIs scalar, vector and 2d von Misses field. Finally, Section 6 provides conclusions and perspectives.

2. Topology optimization

In this section, we review the topology optimization (TO) problem formulation of linear elastic solids using a Finite Element Method (FEM) [12, 25, 13] discretization for the case of compliance minimization problem with volume restriction.

2.1. Problem formulation

Starting in a rectangular domain Ω discretized by n_e square finite elements and n_d Degrees of Freedom (DoF), the equilibrium equation reads:

$$\mathbf{K}(\boldsymbol{\rho})\mathbf{d}(\boldsymbol{\rho}) = \mathbf{p}, \quad (1)$$

where $\mathbf{d} \in \mathbb{R}^{n_d}$, $\mathbf{p} \in \mathbb{R}^{n_d}$, and $\mathbf{K} \in \mathbb{R}^{n_d \times n_d}$ are the displacements vector, the force vector, and the positive definite symmetric global stiffness matrix i.e., already reduced by the boundary conditions respectively [13]. The displacement vector and stiffness matrix depend on the element design variables, stored in the vector $\boldsymbol{\rho} \in \mathbb{R}^{n_e}$, consisting of the relative densities of the elements. By defining the compliance J ,

$$J(\boldsymbol{\rho}) := \mathbf{p} \cdot \mathbf{d}(\boldsymbol{\rho}) = \mathbf{d}(\boldsymbol{\rho}) \cdot \mathbf{K}(\boldsymbol{\rho})\mathbf{d}(\boldsymbol{\rho}), \quad (2)$$

as the objective function to be minimized, the optimization problem is given by:

$$\underset{\boldsymbol{\rho}}{\text{minimize}} \quad J(\boldsymbol{\rho}) = \mathbf{d}(\boldsymbol{\rho}) \cdot \mathbf{K}(\boldsymbol{\rho})\mathbf{d}(\boldsymbol{\rho}) = \sum_{e=1}^{n_e} \mathbf{d}_e(\boldsymbol{\rho}) \cdot \mathbf{K}_e(\boldsymbol{\rho})\mathbf{d}_e(\boldsymbol{\rho}) \quad (3a)$$

$$\text{subject to} \quad \mathbf{K}(\boldsymbol{\rho})\mathbf{d}(\boldsymbol{\rho}) = \mathbf{p}, \quad (3b)$$

$$g(\boldsymbol{\rho}) = \frac{V(\boldsymbol{\rho})}{V_{\max}} - f \leq 0, \quad (3c)$$

$$\mathbf{0} \leq \boldsymbol{\rho} \leq \mathbf{I}_{n_e}. \quad (3d)$$

Considering the area of the e -th element as A_e , the constraint $g(\boldsymbol{\rho})$ ensures that the volume $V(\boldsymbol{\rho}) = \sum_e \rho_e A_e$ does not exceed a prescribed fraction f of the maximum volume allowed in the domain $V_{\max} := V(\boldsymbol{\rho} = \mathbf{I}_{n_e})$. The last constraint enforces the relative densities to acquire values between 0 and 1. Note that the force vector \mathbf{p} is considered fixed throughout all the analyses.

2.2. Penalization and filtering

In order to prevent numerical instabilities such as checker-boarding pattern [26], a stiffness penalization scheme is applied assuming Solid Isotropic Material with Penalization (SIMP) [11]. In this method, the Young's modulus E_e of each element e is related to its density ρ_e through the following penalization function:

$$E_e(\rho_e) = E_{\min} + \rho_e^p (E_0 - E_{\min}), \quad (4)$$

where E_0 is the base material Young's modulus and E_{\min} is a small numerical parameter to avoid the singularity of the stiffness matrix when $\rho_e \approx 0$. The penalization parameter p is selected to enforce the elements to acquire either 0 or 1, and its value is typically set to $p = 3$. Due to the linear regime, the elements present a constant value of Young's modulus, and thus their local stiffness matrix \mathbf{K}_e is linear on such value i.e.,

$$\mathbf{K}_e(\rho_e) = E_e(\rho_e)\mathbf{K}_0, \quad (5)$$

where \mathbf{K}_0 is the local stiffness matrix of a quad element without the influence of the Young's modulus and, hence, not dependant on the densities $\boldsymbol{\rho}$. Then, the global stiffness matrix \mathbf{K} is assembled.

SIMP methods require filtering techniques, either density filter, sensitivity filter, or both. In order to obtain the filtered densities, a convolution with a cone (or, linear) kernel is applied to the

(physical) densities ρ . We generate a weight matrix w_{ij} by evaluating the centroidal coordinate pair \mathbf{c}_i and \mathbf{c}_j for the elements i and j as follows:

$$w_{ij} = \max \{0, R - \|\mathbf{c}_i - \mathbf{c}_j\|_2\}, \quad (6)$$

where R is the filter length prescribed and $\|\cdot\|_2$ represents the Euclidean distance. With this filtering weights, the filtered density of an element e is expressed as:

$$\hat{\rho}_e = \frac{\sum_i w_{ei} \rho_i}{\sum_i w_{ei}}. \quad (7)$$

2.3. Sensitivity analysis

The TO problem set in (3) is solved using a gradient-based approach, e.g. the Method of Moving Asymptotes (MMA) [27] or an Optimality Criteria (OC) method [28]. Therefore, the sensitivity analysis of the cost function $J(\rho)$ and the volume constraint $g(\rho)$ must be addressed. Note that the equilibrium equation constraint (3b) is not considered in these approaches. Instead, the equilibrium is solved, and a local (convex) approximation around its solution is performed. Then, the convex optimization problem considering (3a) and (3c) is solved at each iteration—see sequential approaches, e.g., [29].

The gradient of the cost function (3a) is:

$$\frac{\partial J(\rho)}{\partial \rho_e} = -\mathbf{d}(\rho) \cdot \frac{\partial \mathbf{K}(\rho)}{\partial \rho_e} \mathbf{d}(\rho), \quad (8)$$

where it has been taken into account the derivative of the equilibrium equation (3b) i.e.,

$$\mathbf{K}(\rho) \frac{\partial \mathbf{d}(\rho)}{\partial \rho_e} = -\frac{\partial \mathbf{K}(\rho)}{\partial \rho_e} \mathbf{d}(\rho). \quad (9)$$

The gradient of the cost function (8) can be expressed in a more convenient way as:

$$\frac{\partial J(\rho)}{\partial \rho_e} = -p(E_0 - E_{\min})^{p-1} \mathbf{d}_e(\rho) \cdot \mathbf{K}_0 \mathbf{d}_e(\rho), \quad (10)$$

where \mathbf{d}_e is the displacement vector only considering the degrees of freedom of the element e , i.e., $\mathbf{d}_e \in \mathbb{R}^8$ in the 2D square element case. It has also been taken into account that:

$$\frac{\partial \mathbf{K}_e(\rho_e)}{\partial \rho_e} = \frac{\partial E_e(\rho_e)}{\partial \rho_e} \mathbf{K}_0 = p(E_0 - E_{\min})^{p-1} \mathbf{K}_0 \quad (11)$$

by computing the (straightforward) derivatives of equations (4) and (5). The gradient of the volume constraint (3c) can be straightforwardly computed as:

$$\frac{\partial g(\rho)}{\partial \rho_e} = A_e. \quad (12)$$

In sensitivity filtering, a filter of sensitivities is computed using the chain rule, replacing the design variable ρ with its filtered counterpart $\hat{\rho}$ i.e.,

$$\frac{\partial J(\hat{\rho})}{\partial \rho_e} = \sum_k \frac{\partial J(\hat{\rho})}{\partial \hat{\rho}_k} \frac{\partial \hat{\rho}_k}{\partial \rho_e}, \quad \frac{\partial g(\hat{\rho})}{\partial \rho_e} = \sum_k \frac{\partial g(\hat{\rho})}{\partial \hat{\rho}_k} \frac{\partial \hat{\rho}_k}{\partial \rho_e}. \quad (13)$$

Differentiating the filtering function (7) yields the term needed to compute the previous equation:

$$\frac{\partial \hat{\rho}_k}{\partial \rho_e} = \frac{w_{ke}}{\sum_i w_{ki}}. \quad (14)$$

2.4. Stress computation

The von Mises stress distribution $\boldsymbol{\sigma}^{vM} \in \mathbb{R}^{n_e}$ is a vector containing the von Mises stress values σ_e^{vM} at every element. This distribution is computed as a post-processing of the optimization problem, to account for the response (i.e., solution) of a structural problem in which the input is the geometry defined by the optimized topology.

First, the stress vector of an element e as a function of the coordinate \underline{x} in the Voigt notation for the 2d case is:

$$\boldsymbol{\sigma}_e(\underline{x}) = (\sigma_{exx}, \sigma_{eyy}, \tau_{exy})^T,$$

and it is computed in the FE analysis as follows:

$$\boldsymbol{\sigma}_e(\underline{x}) = \mathbf{D}_e \mathbf{B}_e(\underline{x}) \mathbf{d}_e, \quad (15)$$

where \mathbf{D}_e is the plane stress constitutive matrix and \mathbf{B}_e the kinematic matrix, both particularized for the element e . Note that the constitutive matrices may present different values since they depend on their element's stiffness $E_e(\rho_e)$. The von Mises stress measure reads:

$$\sigma_e^{vM}(\underline{x}) = (\sigma_{exx}^2 + \sigma_{eyy}^2 - \sigma_{exx}\sigma_{eyy} + 3\tau_{exy}^2)^{\frac{1}{2}}. \quad (16)$$

Lastly, to obtain a unique von Mises measurement per element e , $\sigma_e^{vM}(\underline{x})$ is averaged in the element domain Ω_e performing numerical integration, namely:

$$\sigma_e^{vM} = \int_{\Omega_e} \sigma_e^{vM}(\underline{x}) d\Omega_e = \sum_{p=1}^{n_{ip}} \sigma_e^{vM}(\underline{x}_p) w_p^2, \quad (17)$$

where n_{ip} is the number of integration points in the Gaussian quadrature and w_p the corresponding weights evaluated at such integration points p . Recall that quad 2d elements with $n_{ip} = 4$ integration points are used in this paper.

3. Rank Reduction Autoencoder (RRAE)

In this section, we review the Rank Reduction Autoencoder (RRAE) formulation, introduced by Mounayer et al. [15], and highlight their advantages in comparison with regular autoencoders (AEs). This model is applied to perform dimensionality reduction to our datasets through a non-linear encoding. The input data can be of different nature, e.g., grid/pixel data, or a field defined in a domain, such as geometry or topology, among others. Let consider $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_{n_s}\}$, with $\mathbf{x}_j \in \mathbb{R}^T$ the input data consisting of n_s samples. Note that the input space dimension can be either 1d or 2d i.e., $T = \{D, D \times D\}$, being D the number of embedded features in the domain.

It is known that regular AEs perform the reconstruction through the following non-linear mapping $\tilde{\mathbf{X}} = \mathcal{D}(\mathcal{E}(\mathbf{X}))$ [14], where $\mathcal{E}(\cdot)$ and $\mathcal{D}(\cdot)$ are the encoder and decoder mappings, respectively. However, a low-rank approximation i.e., truncation, of a high-dimensional encoding mapping $\tilde{\mathbf{Y}}$ is introduced in RRAE formulation, to be decoded later, as shown in Figure 1.

Whereas AEs perform directly the map to a lower dimension, RRAEs perform the encoding mapping a sample \mathbf{x}_j into a latent space of dimension L , and the concatenation of samples forms the matrix $\mathbf{Y} \in \mathbb{R}^{L \times n_s}$. Then, a low-rank approximation of this matrix is performed via a truncated Singular Value Decomposition (SVD), i.e., $\mathbf{Y} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{L \times r}$, $\mathbf{V}^T \in \mathbb{R}^{r \times n_s}$, and the singular values are stored in the diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{r \times r}$ sorted from highest to lowest such that $\boldsymbol{\Sigma} = \text{diag}(\sigma_i)$, $i = 1, \dots, r$. Keeping only $k_{\max} \ll r$ modes from the singular values, so-called latent coefficients are defined and can be expressed as a linear combination of a orthogonal basis – since \mathbf{Y} is a real matrix – consisting of the first k_{\max} rows of the matrix \mathbf{U} . This basis is computed during the training process [15].

Therefore, the reconstruction of a latent sample $\tilde{\mathbf{y}}_j$ after applying the truncated SVD might be expressed as:

$$\tilde{\mathbf{y}}_j = \sum_{i=1}^{k_{\max}} (\sigma_i \mathbf{U}_i \mathbf{V}_i^T)_j =: \sum_i^{k_{\max}} \alpha_{i,j} \mathbf{U}_i, \quad j = 1, \dots, n_s, \quad (18)$$

where $\alpha_{i,j} := \sigma_i \mathbf{V}_{i,j}^T$ are the above-mentioned latent coefficients. Thus, in matrix form, $\tilde{\mathbf{Y}}$ reads:

$$\tilde{\mathbf{Y}} = \mathbf{U} \mathbf{A}, \quad (19)$$

with $A_{i,j} = \alpha_{i,j}$, the matrix $\mathbf{A} \in \mathbb{R}^{k_{\max} \times n_s}$ containing these coefficients, and $\mathbf{U} \in \mathbb{R}^{L \times k_{\max}}$, with a slight abuse of notation, a truncated version of the \mathbf{U} matrix from the SVD. Lastly, the reconstructed input data is performed applying the decoder map $\tilde{\mathbf{X}} = \mathcal{D}(\tilde{\mathbf{Y}})$. With that, a reconstruction loss accounting for the data-driven fitting is defined as follows:

$$\mathcal{L}(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{\|\mathbf{X} - \tilde{\mathbf{X}}\|_F}{\|\mathbf{X}\|_F}, \quad (20)$$

where $\|\cdot\|_F$ corresponds to the Frobenius norm. An example of application of this architecture to 2d input data (images, thus CNN-RRAE) is depicted in Figure 1.

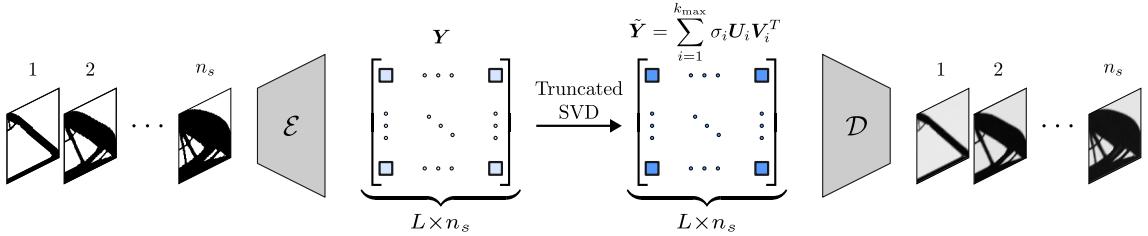


Figure 1: CNN RRAE architecture applied to a topology optimization example, consisting of an encoder \mathcal{E} , performing the SVD and decode \mathcal{D} . $L \gg T$ is the high-rank latent space dimension, k_{\max} is the low-rank latent size, i.e., number of modes retained from the truncated SVD, and n_s is the number of samples in the data.

4. Topology optimization using the RRAE

This section presents the proposed data-driven framework for topology optimization based on Rank Reduction Autoencoders (RRAEs). The approach leverages the ability of RRAEs to compress high-dimensional topology optimization data into low-dimensional latent representations while adaptively identifying the most informative modes through rank truncation. By constructing separate reduced-order models for optimized geometries and for associated quantities of interest, the framework enables efficient learning of the relationships between design variables and structural responses. These latent representations are subsequently coupled through neural regressors to address both forward and inverse problems, providing a unified and computationally efficient surrogate modeling strategy for topology optimization.

Let \mathbf{X} be a set of discretized geometry data, and each of the samples are introduced in a forward PDE problem e.g., a BVP such as a structural domain to solve the equilibrium, forming the set of discretized solutions \mathbf{S} of such underlying physical problem. We will also refer to \mathbf{S} as the Quantity of Interest (QoI) of the problem.

By applying an already trained RRAE, a geometry \mathbf{x}_j can be expressed in the latent space from the couple $(\boldsymbol{\alpha}_j, \mathbf{U})$ in the following way:

$$\mathbf{y}_j = \mathcal{E}^g(\mathbf{x}_j) \quad \rightarrow \quad \mathbf{y}_j = \mathbf{U} \boldsymbol{\alpha}_j \quad \Rightarrow \quad \boldsymbol{\alpha}_j = \mathbf{U}^T \mathbf{y}_j, \quad (21)$$

where \mathbf{y}_j is the associated embedded geometry and $\mathcal{E}^g(\cdot)$ is the encoder mapping of the geometries. The full set of samples might be expressed as $\mathbf{A} = \mathbf{U}^T \mathbf{Y}$, where $\mathbf{Y} = \mathbf{U} \mathbf{A} = \mathcal{E}^g(\mathbf{X})$. Then, the reconstruction of a sample $\tilde{\mathbf{x}}_j$ is performed as:

$$\tilde{\mathbf{x}}_j = \mathcal{D}^g(\mathbf{U} \boldsymbol{\alpha}_j), \quad (22)$$

where $\mathcal{D}^g(\cdot)$ is the decoder mapping of the geometry RRAE.

Analogously, by separately training an RRAE for the QoI, a solution \mathbf{s}_j can be expressed in its latent space given the so-defined couple $(\boldsymbol{\beta}_j, \mathbf{V})$ of coefficient and basis such that:

$$\mathbf{z}_j = \mathcal{E}^s(\mathbf{s}_j) \quad \rightarrow \quad \mathbf{z}_j = \mathbf{V} \boldsymbol{\beta}_j \quad \Rightarrow \quad \boldsymbol{\beta}_j = \mathbf{V}^T \mathbf{z}_j, \quad (23)$$

where \mathbf{z}_j is the embedded solution and $\mathcal{E}^s(\cdot)$ the encoder mapping for solutions. Similarly, the reconstruction of the solution \mathbf{s}_j is:

$$\tilde{\mathbf{s}}_j = \mathcal{D}^s(\mathbf{V}\boldsymbol{\beta}_j), \quad (24)$$

where $\mathcal{D}^s(\cdot)$ is the decoder mapping for the solutions.

In other words, low-rank approximations for the geometries in terms of latent coefficients $\boldsymbol{\alpha}_j$ and basis \mathbf{U} , and for the solutions i.e., QoIs, in terms of latent coefficients $\boldsymbol{\beta}_j$ and basis \mathbf{V} are found. In order to join both latent spaces, mapping functions relating $\boldsymbol{\alpha}_j$ to $\boldsymbol{\beta}_j$ and vice versa have to be obtained. Therefore, MLPs are used as interpolators to build these functions. Specifically, we will refer to $\mathcal{NN}_*^d : \boldsymbol{\alpha}_j \rightarrow \boldsymbol{\beta}_j$ as the neural network addressing the ‘direct’ problem i.e., from geometry to solution, and $\mathcal{NN}_*^i : \boldsymbol{\beta}_j \rightarrow \boldsymbol{\alpha}_j$ as the neural network connecting the ‘inverse’ problem i.e., from solution to geometry.

Considering the joint of these two architectures, two novel frameworks are developed: fast solution inference for a (potentially new) geometry – so-called ‘direct’ problem – and geometry discovery from a target performance—so-called ‘inverse’ problem. In the first case, the steps to apply the direct problem given a geometry \mathbf{x}_j are:

- Geometry embedding: $\mathbf{y}_j = \mathcal{E}^g(\mathbf{x}_j)$
- Projection in the (geometry) latent space: $\boldsymbol{\alpha}_j = \mathbf{U}^T \mathbf{y}_j$
- Evaluate the associated coefficients $\boldsymbol{\beta}_j$ using the trained regression: $\boldsymbol{\beta}_j = \mathcal{NN}_*^d(\boldsymbol{\alpha}_j)$
- Solution evaluation: $\tilde{\mathbf{s}}_j = \mathcal{D}^s(\mathbf{V}\boldsymbol{\beta}_j)$

In the second case, the steps to address the inverse problem are analogous. Starting from a solution \mathbf{s}_j ,

- Solution embedding: $\mathbf{z}_j = \mathcal{E}^s(\mathbf{s}_j)$
- Projection in the (solution) latent space: $\boldsymbol{\beta}_j = \mathbf{V}^T \mathbf{z}_j$
- Evaluate the latent coefficient using (another) trained regression: $\boldsymbol{\alpha}_j = \mathcal{NN}_*^i(\boldsymbol{\beta}_j)$
- Geometry retrieval: $\tilde{\mathbf{x}}_j = \mathcal{D}^g(\mathbf{U}\boldsymbol{\alpha}_j)$

We refer to these two ways as the following mappings:

$$\tilde{\mathbf{S}} = \text{geo2sol}_*(\mathbf{X}), \quad \tilde{\mathbf{X}} = \text{sol2geo}_*(\mathbf{S}),$$

While keeping the geometry RRAE fixed, in this paper we propose three different QoIs regarding the solutions, namely:

- **Scalar.** This QoI does not require a low-rank approximation, therefore the mapping directly relates the (geometry) latent coefficients with the scalar value. In this case, the scalar is a function of the von Mises stress distribution i.e., $f(\boldsymbol{\sigma}^{vM})$, as highlighted in Figure 2a.
- **1d field in a fixed domain.** This QoI comprises the von Mises stress values of the elements belonging in the diagonal of the domain Ω , since this is the main load path. Since a curve is obtained as the QoI – hence the 1d field – an MLP-RRAE is used as the low-rank model.
- **2d field in a fixed domain.** The QoI is the whole von Mises distribution in the domain, $\boldsymbol{\sigma}^{vM}$ expressed in a matrix form to be understood as an image by the CNN-RRAE. This is depicted in Figure 2b.

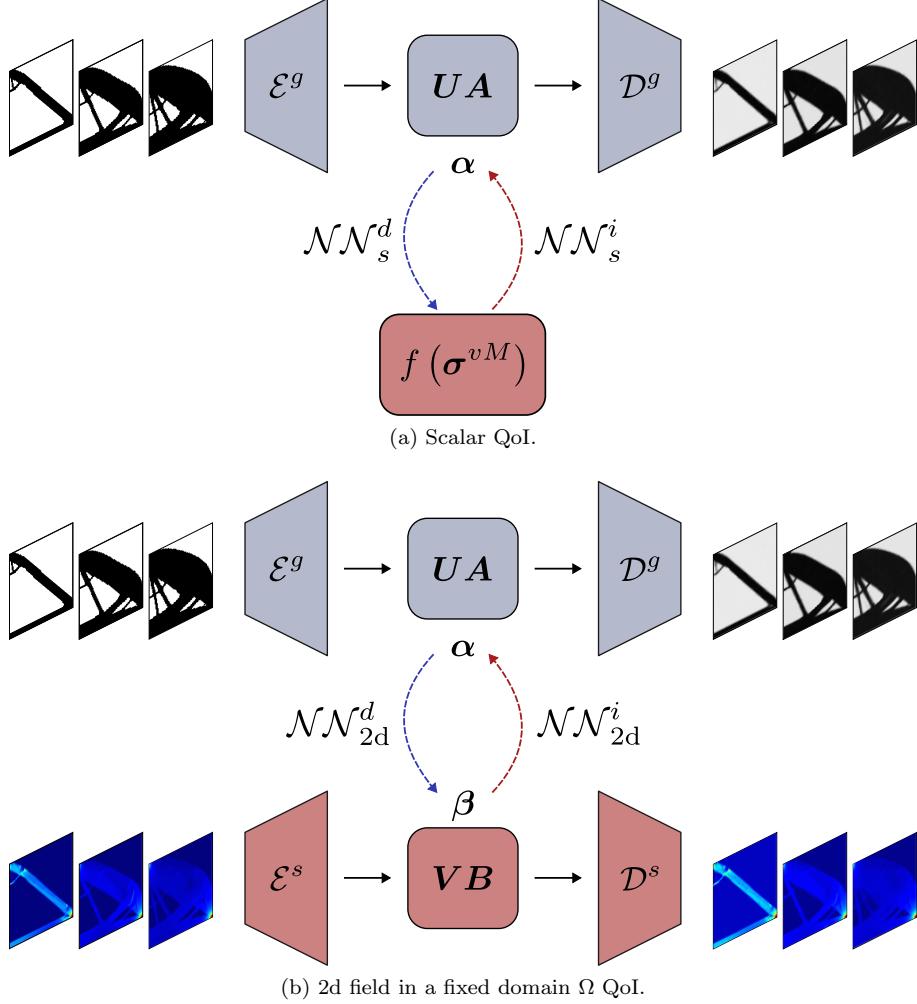


Figure 2: Pipeline to address the direct `geo2sol_*` or inverse `sol2geo_*` problems, for (a) scalar and (b) 2d field QoIs. There are three models in each case that are trained *separately*: geometry RRAE (UA), solution RRAE (VB), and the mappings between both latent coefficients, either $\mathcal{NN}_*^d : \alpha \rightarrow \beta$ or $\mathcal{NN}_*^i : \beta \rightarrow \alpha$.

Lastly, another advantage that this methodology might present is its use in GD. New geometries or solutions might be generated by choosing new points in the latent space, i.e., $\hat{\alpha}$ or $\hat{\beta}$ and then decoding in the corresponding problem—geometry or solution, respectively. Particularly, these new points might be expressed as interpolations of the training set latent coefficients, i.e., within the convex hull of the latent coefficients space in order to avoid extrapolations. Although this is a potentially powerful tool for GD with applications such as dataset enriching, discovery of new topologies, etc., this proposal is out of the scope of this paper.

5. Numerical results

We now proceed to demonstrate the utility and efficiency of the proposed methodology using a TO example and different QoIs. The main example consists of performing a Design of Experiments (DoE) by sweeping the volume constraint in a compliance minimization problem, as the half MBB beam displayed in Figure 3.

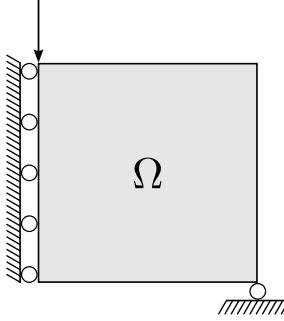


Figure 3: Domain Ω and boundary conditions of the half MBB beam for the TO problem.

The domain Ω is discretized into 80×80 quad FEs of linear elastic base material with unit Young's modulus $E_0 = 1$ and Poisson's ratio $\nu = 0.3$. For the TO problem, a SIMP scheme with a penalization parameter of $p = 3$ and a sensitivity filter as the one described in Section 2, with a filter radius $R = 1.5$ of the element's size is applied. A minimum Young's modulus value of $E_{\min} = 10^{-9}$ is introduced in the elements in order to avoid obtaining singular stiffness matrices \mathbf{K} . Regarding the volume constraint f , this value has been varied by generating a vector of $N = 100$ elements equispaced linearly between 0.1 and 0.9. Therefore, a total of N TO problems are solved using the Optimality Criteria (OC) method provided in the `top88.m` open source script by DTU [30] in order to generate the dataset to be used in this proposed methodology. Note that the boundary conditions (BC) are fixed in the analysis, wherein the force applied is a unit load. There is a possibility to parameterize the BCs e.g., by displacing the load application and/or support node along the boundary of the domain $\partial\Omega$, however, this idea is out of the scope of this paper.

Once the set of optimal topologies $\{\rho_1^*, \dots, \rho_N^*\}$ is obtained, they are post-processed and the corresponding set of von Mises stress distributions $\{\sigma_1^{vM,*}, \dots, \sigma_N^{vM,*}\}$ are computed according to Section 2.4. The set of optimal topologies constitutes the discretized geometries dataset \mathbf{X} , in which a sample \mathbf{x}_j is defined by:

$$\mathbf{x}_j := \rho_j^*,$$

properly rearranged in a matrix i.e., replicating the 2d mesh. On the other hand, the von Mises stress distribution set constitutes the discretized solutions dataset \mathbf{S} , and each sample s_j is assigned as a function of the $\sigma_j^{vM,*}$ distribution depending on the type of QoI considered {scalar, 1d, 2d}.

Thus, the training data \mathbf{X} and \mathbf{S} are generated to train the geometries and solutions RRAEs separately. Afterwards, we train models that represent a mutual fitting between both latent spaces. In addition to these datasets, a test dataset is generated to assess the accuracy of the methodology in new data samples. To this end, another DoE of $N_{\text{test}} = 20$ volume fractions linearly equispaced between 0.1 and 0.9 is performed. Note that there are no common volume fraction values with respect to the training data but 0.1 and 0.9.

In the following subsections, three types of QoIs are analyzed: scalar, 1d field in a fixed domain, and 2d field in a fixed domain. In each of them, the direct and inverse problems are addressed.

5.1. Scalar quantity of interest

The first case corresponds to the relation between the geometries latent coefficients and an equivalent or global response of the structure. This global magnitude is the maximum von Mises stress value in the domain, which may be of particular interest for industrial applications since it is often included in TO problems as a constraint. Thus, the samples s_j of the solutions dataset \mathbf{S} are assigned as follows:

$$s_j := \max(\sigma_j^{vM,*}).$$

These scalar problems are the most challenging of the three. The more compressed the solution is – a global magnitude contains less information than the full resolution field – the less bijective the problem is. Hence, the inverse problem becomes more ill-posed.

The next step is to build the geometries RRAE model. As the input are the discretized relative densities ρ_j^* of the elements, rearranged in matrix form to maintain the mesh layout and neighborhood i.e., an image, a convolutional autoencoder architecture is used (CNN-RRAE). The

chosen latent size is $L = 500$, to further compress via SVD retaining $k_{\max} = 2$ modes. Theoretically, an accurate reconstruction can be performed with only one mode – since it is only the volume fraction the parameter varying across the dataset – however, we keep 2 modes in order to obtain predictions with higher accuracy. The rest of hyper-parameters are displayed in Table 1.

Module	Layer	Input shape	Output shape	Activation	Kernel
Encoder	Conv. Layer	$(n_b, 1, 80, 80)$	$(n_b, 32, 40, 40)$	ReLU	(3×3)
	Conv. Layer	$(n_b, 32, 40, 40)$	$(n_b, 64, 20, 20)$	ReLU	(3×3)
	Conv. Layer	$(n_b, 64, 20, 20)$	$(n_b, 128, 10, 10)$	ReLU	(3×3)
	Dense	$(n_b, 12800)$	$(n_b, 500)$	ReLU	–
Decoder	Dense	$(n_b, 500)$	$(n_b, 3200)$	ReLU	–
	Conv. Trans. Layer	$(n_b, 32, 10, 10)$	$(n_b, 128, 20, 20)$	ReLU	(3×3)
	Conv. Trans. Layer	$(n_b, 128, 20, 20)$	$(n_b, 64, 40, 40)$	ReLU	(3×3)
	Conv. Trans. Layer	$(n_b, 64, 40, 40)$	$(n_b, 32, 80, 80)$	ReLU	(3×3)
Final Conv. Layer		$(n_b, 32, 80, 80)$	$(n_b, 1, 80, 80)$	Linear	(1×1)

Table 1: Hyper-parameters of the CNN-RRAE model for the geometries, where n_b is the number of samples in a batch. The parameters common to all convolutional layers in both encoder and decoder are padding = 1, stride = 2 and dilation = 1, except for the final convolutional layer, which has no padding and stride = 1 in order to recover the initial 80×80 shape—acting as a ‘trainable’ pooling for the channels.

Since it is already defined in $[0, 1]$, geometry data \mathbf{x}_j is fed to the CNN-RRAE model without any normalization. The training parameters are shown in Table 2. There are three training stages, in which the learning rate is decreased to a tenth of the value of the previous stage so smoother steps are performed at later epochs.

Parameter	Choice
Optimizer	AdaBelief
Learning rate	$[10^{-3}, 10^{-4}, 10^{-5}]$
Epochs	[3500, 3500, 3500]
Batch size	[20, 20, 20]
Loss	Equation (20)

Table 2: CNN-RRAE geometries model. Optimization parameters of the training.

After the training, the reconstruction loss of the geometries RRAE is $\mathcal{L}_{\text{train}} = 4.32\%$ in the training set, and its value in the test counterpart is $\mathcal{L}_{\text{test}} = 6.32\%$. Examples for both train and test reconstructions for three arbitrary samples are depicted in Figure 4.

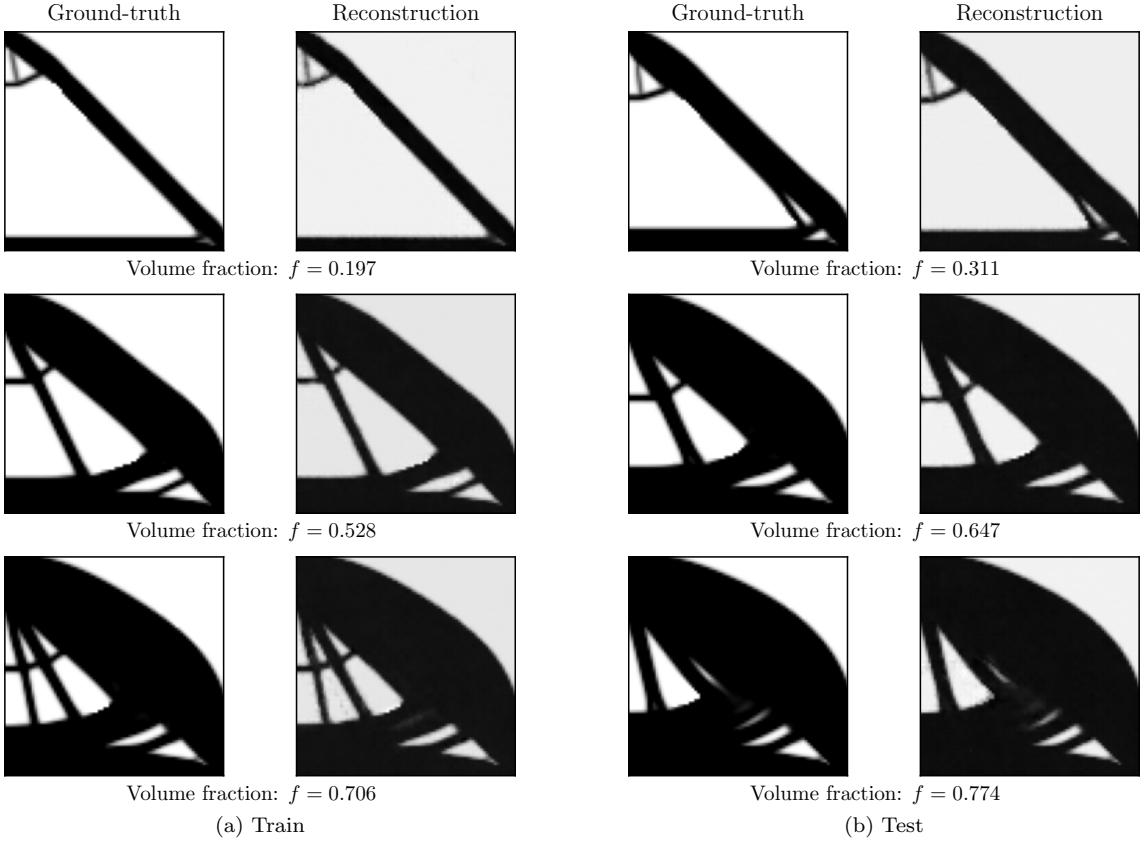


Figure 4: CNN-RRAE geometries model. Example reconstructions in (a) train and (b) test sets.

5.1.1. Direct problem

Once obtained the geometries latent coefficients \mathbf{A} , we propose a mapping relating such coefficients with the scalar QoI i.e., the maximum von Mises stress σ_{\max}^{vM} in the domain. In order to build the whole pipeline within a DL framework, an MLP is trained to this end. Thus, the sought mapping can be expressed as $\mathcal{NN}_s^d : \boldsymbol{\alpha}_j \rightarrow s_j$, where the superscript d stands for direct problem. In order to only highlight the hyper-parameters of the RRAE models – those of interest in this paper – the hyper-parameters of the MLP architecture are displayed in Table A.1 in [Appendix A.1](#). Both inputs and outputs are normalized with a standard scaler i.e., each feature being subtracted its mean and divided by its standard deviation. An 80-20% split is performed to generate the validation set. Then, the training is carried out with the optimization parameters highlighted in Table A.2, also in [Appendix A.1](#).

Performing the training, the loss value in the training set is $MSE_{\text{train}} = 1.4 \cdot 10^{-1}$, which represents a slightly high value for this kind of standards. This may be also observed in the coefficient of determination, obtaining $R^2_{\text{train}} = 0.814$ and $R^2_{\text{test}} = 0.930$. The fitting curve comparing the ground-truth σ_{\max}^{vM} against the \mathcal{NN}_s^d predicted values is depicted in Figure A.1, in [Appendix A.2](#). All the fitting curves from the \mathcal{NN}_s^d and \mathcal{NN}_s^i MLP models are depicted in such appendix to avoid the proliferation of plots.

The next step within the scalar QoI direct problem is to join the CNN-RRAE geometries and MLP \mathcal{NN}_s^d models to create a pipeline. Such pipeline is referred to as geo2sol_s , performing the following steps (also described in Section 4): (1) takes a geometry \mathbf{x}_j as input, (2) encodes such geometry through the first part of the CNN-RRAE geometry model to obtain the corresponding latent coefficient $\boldsymbol{\alpha}_j$, and (3) maps such latent coefficient through \mathcal{NN}_s^d to predict the scalar QoI \tilde{s}_j , i.e., the maximum von Mises stress σ_{\max}^{vM} in the original input geometry. Thus, following this procedure allows solutions to be obtained without having to go through the FE solver, and therefore achieving a surrogate model enabling computational efficiency.

To evaluate the accuracy of predictions in train and test datasets, the coefficient of determina-

tion between the ground-truth σ_{\max}^{vM} values and the geo2sol_s predicted ones is computed, yielding the following:

$$R_{\text{train}}^2 = 0.867, \quad R_{\text{test}}^2 = 0.987.$$

Since the predictions in the \mathcal{NN}_s^d are not sufficiently accurate, this is reflected in the geo2sol_s predictions, especially for the train set. As it was mentioned, this case presents worse performance due to non-bijectivity of the geo2sol_s mapping. The train and test curves displaying the true values against the predicted ones with this direct methodology for the scalar QoI are depicted in Figure 5.

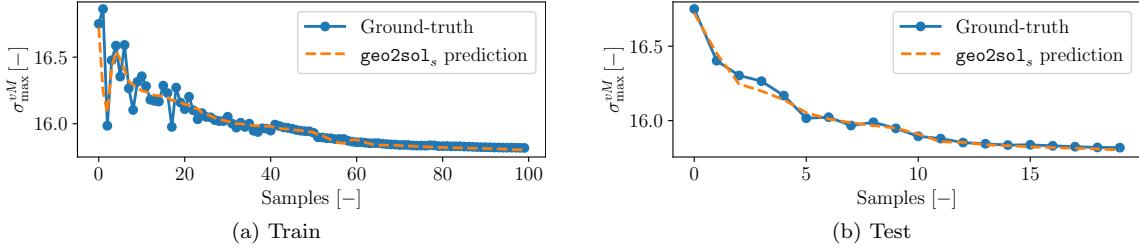


Figure 5: geo2sol_s predictions in (a) train and (b) test.

5.1.2. Inverse problem

In this subsection, an analogous inverse procedure is performed. In this case, we propose a way of obtaining geometries from the scalar QoI. In order to do so, the first step is to build the inverse mapping $\mathcal{NN}_s^i : s_j \rightarrow \boldsymbol{\alpha}_j$, where the superscript i stands for inverse. This model is an MLP with the same architecture and training parameters as its direct counterpart – Tables A.1 and A.2 – with the input and output dimensions permuted since $s_j \in \mathbb{R}^1$ and $\boldsymbol{\alpha}_j \in \mathbb{R}^2$. Normalizing both inputs and outputs, and training the model, a loss value of $\text{MSE}_{\text{train}} = 6.4 \cdot 10^{-2}$ is obtained. The coefficients of determination between ground-truth and \mathcal{NN}_s^i predicted latent coefficients values $\boldsymbol{\alpha} = [\alpha_0, \alpha_1]$ are $R_{\text{train}}^2 = 0.943$ and $R_{\text{test}}^2 = 0.945$. The corresponding curves are depicted in Figure A.2 (Appendix A.2).

Having fitted \mathcal{NN}_s^i , the pipeline sol2geo_s (inverse to the previous pipeline) can be generated. In this case, a certain value of σ_{\max}^{vM} i.e., the scalar QoI s_j , is requested, so it is fed to the pipeline as an input. The value s_j is mapped into the corresponding geometry latent coefficient $\boldsymbol{\alpha}_j$ through \mathcal{NN}_s^i . Finally, a geometry $\tilde{\mathbf{x}}_j$ is retrieved by passing $\boldsymbol{\alpha}_j$ through the decoder of the CNN-RRAE geometries model. Since the output is a mesh that can be interpreted as an image with each pixel having a density value $\rho_j^* \in [0, 1]$, the reconstruction error between the ground-truth geometry \mathbf{x}_j and the sol2geo_s predicted geometry $\tilde{\mathbf{x}}_j$ is performed. This in turn implies the computation of the L^2 norm of the difference of the flattened matrices, as highlighted in Equation (20), yielding the following loss values:

$$\mathcal{L}_{\text{train}} = 20.59\%, \quad \mathcal{L}_{\text{test}} = 20.22\%.$$

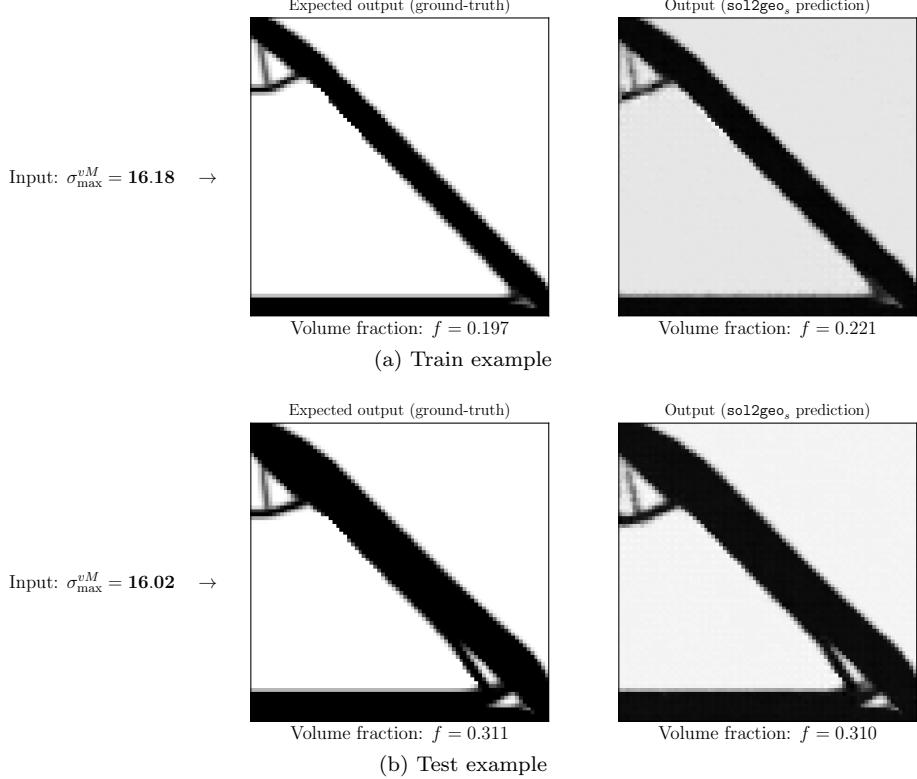


Figure 6: sol2geo_s example predictions in (a) train and (b) test.

Figure 6 displays examples of sol2geo_s train and test predictions when requesting a σ_{\max}^{vM} value, against the corresponding ground-truth geometries. As can be noticed in the errors or in the figure, it is important to emphasize that actually in this scalar case architecture the model is learning in an overfitted way, i.e., memorizing, a ‘bijective’ type of mapping, which is not a good practice for this type of models. This highlights that more information is needed on the solution side for this methodology to be successful, as we shall show in the next two cases in this paper. Additionally, when the requested value is out of the bounds of the training set, this pipeline starts to produce geometries that lack physical meaning, which is another drawback in this first case.

5.2. 1d field quantity of interest in a fixed domain

This second analysis showcases the relation between the geometries latent coefficients (from the previous model) and the latent coefficients of the response of the structure in a 1d field. In this case, the von Mises stress values of the elements of the diagonal (top-left to bottom-right elements in Ω). They are of special interest since they comprise the main load path—connecting to nodes affected by the BC. Thus, the samples s_j of the solutions dataset S are assigned as follows:

$$s_j := \text{diag}(\sigma_j^{vM,*}).$$

The RRAE model regarding the geometries is the same as the previous Section 5.1. The next step is to build the solutions RRAE model. The solutions input is dense i.e., a matrix, therefore an MLP-RRAE architecture is selected. The latent size is $L = 500$, and the truncated SVD compression is performed keeping $k_{\max} = 1$. The rest of hyper-parameters are displayed in Table 3.

Module	Layer	Input shape	Output shape	Activation
Encoder	Dense	$(n_b, 80)$	$(n_b, 64)$	ReLU
	Dense	$(n_b, 64)$	$(n_b, 500)$	Linear
	Dense #1	$(n_b, 500)$	$(n_b, 64)$	ReLU
	Dense #2	$(n_b, 64)$	$(n_b, 64)$	ReLU
Decoder			\vdots	
	Dense #6	$(n_b, 64)$	$(n_b, 64)$	ReLU
	Final Layer	$(n_b, 64)$	$(n_b, 80)$	Linear

Table 3: Hyper-parameters of the MLP-RRAE model for the von Mises stress 1d field solutions, where n_b is the number of samples in a batch. In the decoder, the properties of dense layers #2 to #6 are the same, hence they are omitted for the sake of simplicity.

The training parameters are the same as those displayed in Table 2. Once fitted, the reconstruction loss of the solutions RRAE is $\mathcal{L}_{\text{train}} = 3.14\%$ in the training set, and its value in the test counterpart is $\mathcal{L}_{\text{test}} = 3.98\%$. As an illustration, three examples of train and test solutions are depicted in Figure 7.

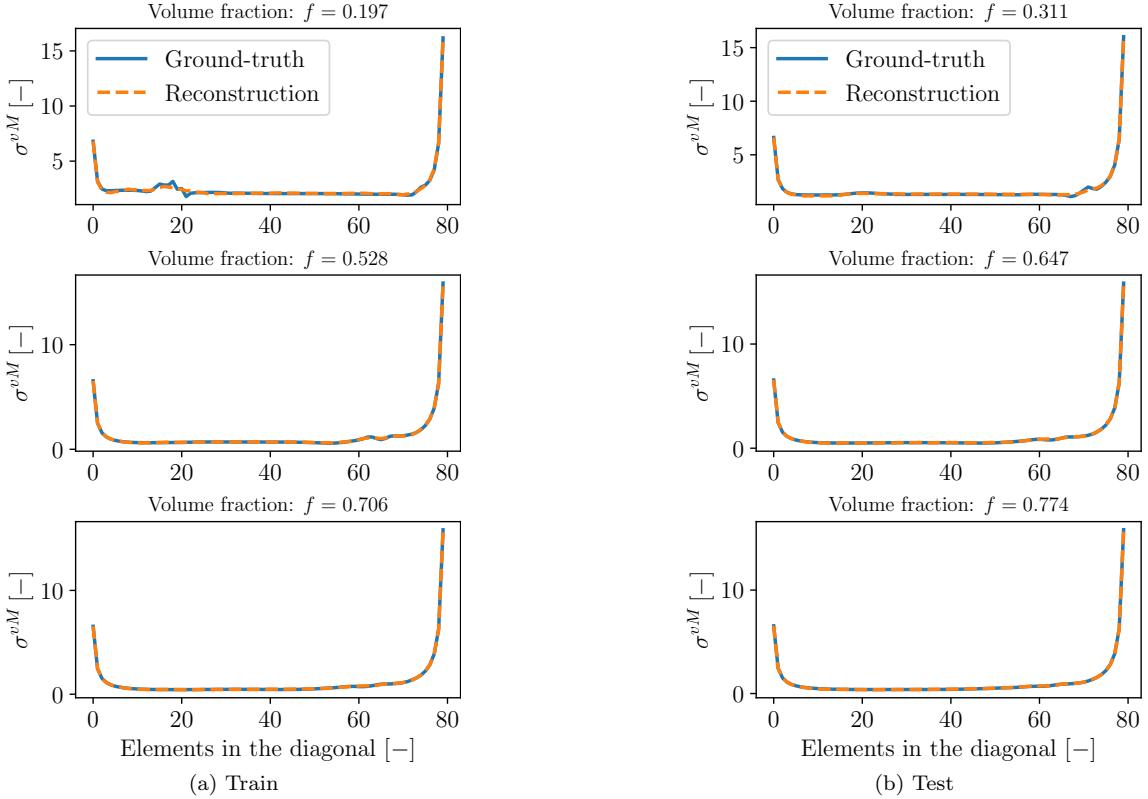


Figure 7: MLP-RRAE 1d solutions model (von Mises stress in the diagonal). Example reconstructions in (a) train and (b) test sets.

5.2.1. Direct problem

Regarding the methodology for this 1d field QoI, the next step consists in finding a mapping relating geometries latent coefficients \mathbf{A} and solutions latent coefficients \mathbf{B} . Again, an MLP is trained to this end, and the sought mapping can be expressed as $\mathcal{NN}_{1d}^d : \boldsymbol{\alpha}_j \rightarrow \boldsymbol{\beta}_j$, where the superscript d stands for direct problem. Recall that $\boldsymbol{\alpha}_j \in \mathbb{R}^2$ and $\boldsymbol{\beta}_j \in \mathbb{R}^1$ due to the chosen k_{\max} (hyper-)parameters in previous sections.

Analogously, the hyper-parameters of the MLP architecture are displayed in Table A.1, the optimization parameters are highlighted in Table A.2, and both input and output are normalized before performing the training. The loss value in the train is $\text{MSE}_{\text{train}} = 2.2 \cdot 10^{-3}$. The coefficient of

determination R^2 is another metric to assess the goodness of the fit, having obtained $R_{\text{train}}^2 = 0.998$ and $R_{\text{test}}^2 = 0.989$. The prediction of every latent coefficient β_0 through \mathcal{NN}_{1d}^d compared with its ground-truth value is depicted in Figure A.3. Note the smoothness of latent coefficient, achieved thanks to the use of MLP-RRAE of the solutions. In other words, the dimensionality reduction keeps the physical meaning of varying the volume fraction f , making the relation between latent spaces an easier task with respect to the previous scalar counterpart.

Now, we have three trained models: a CNN-RRAE for the geometries, an MLP-RRAE for 1d-solutions, and an MLP \mathcal{NN}_{1d}^d relating latent coefficients of the both. Joining all them properly by following the procedure outlined in Section 4, we can generate the pipeline geo2sol_{1d} , whose input is a geometry \mathbf{x}_j , and its output is $\tilde{\mathbf{s}}_j$, the von Mises stress distribution across the diagonal (top-left to bottom-right elements) of such geometry i.e., the 1d field highlighted in this section. To assess the accuracy of predictions, the coefficient of determination R^2 between ground-truth and predicted 1d curves is computed per sample, and then averaged across the dataset. These coefficients are:

$$R_{\text{train}}^2 = 0.997, \quad R_{\text{test}}^2 = 0.997.$$

Figure 8 displays samples of successful geo2sol_{1d} predictions in train and test sets.

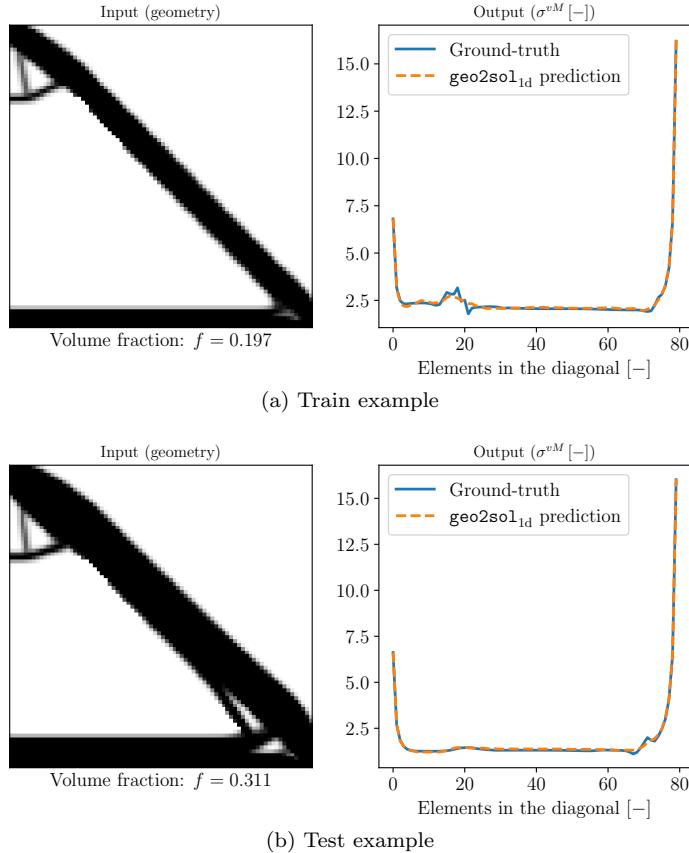


Figure 8: geo2sol_{1d} example predictions in (a) train and (b) test.

5.2.2. Inverse problem

This procedure is analogous to the previous subsection, but generating an inverse mapping relating latent coefficients i.e., from \mathbf{B} to \mathbf{A} . The mapping \mathcal{NN}_{1d}^i , where now i stands for inverse, is again an MLP that inputs $\beta_j \in \mathbb{R}^1$ and outputs $\alpha_j \in \mathbb{R}^2$. Their hyper-parameters are the same as the previous (Table A.1), with the input and output dimensions permuted, but with the same hidden layer configurations. The same training parameters (Table A.2) are likewise used. The loss value is $\text{MSE}_{\text{train}} = 1.1 \cdot 10^{-3}$, and the coefficients of determination, $R_{\text{train}}^2 = 0.999$ and $R_{\text{test}}^2 = 0.999$. The two curves for the prediction of coefficients $\alpha = [\alpha_0, \alpha_1]$ are depicted in Figure A.4.

With \mathcal{NN}_{1d}^i , the **sol2geo_{1d}** pipeline can be built, whose input is the von Mises stress curve (across the diagonal of the domain Ω), predicting the geometry which generates such curve—hence the inverse problem is addressed. The encoder of MLP-RRAE of solutions maps the input curve s_j into a (single) latent coefficient $\beta_{0,j}$. Then, such coefficient is mapped through \mathcal{NN}_{1d}^i to generate the latent geometry coefficients α_j , to be further mapped through the CNN-RRAE decoder, thus generating a geometry \tilde{x}_j as the output. Evaluating the reconstruction loss, Equation (20), between ground-truth and **sol2geo_{1d}** predicted geometries yields the following:

$$\mathcal{L}_{\text{train}} = 6.38\%, \quad \mathcal{L}_{\text{test}} = 7.63\%.$$

Figure 9 displays samples of successful **sol2geo_{1d}** predictions in train and test sets.

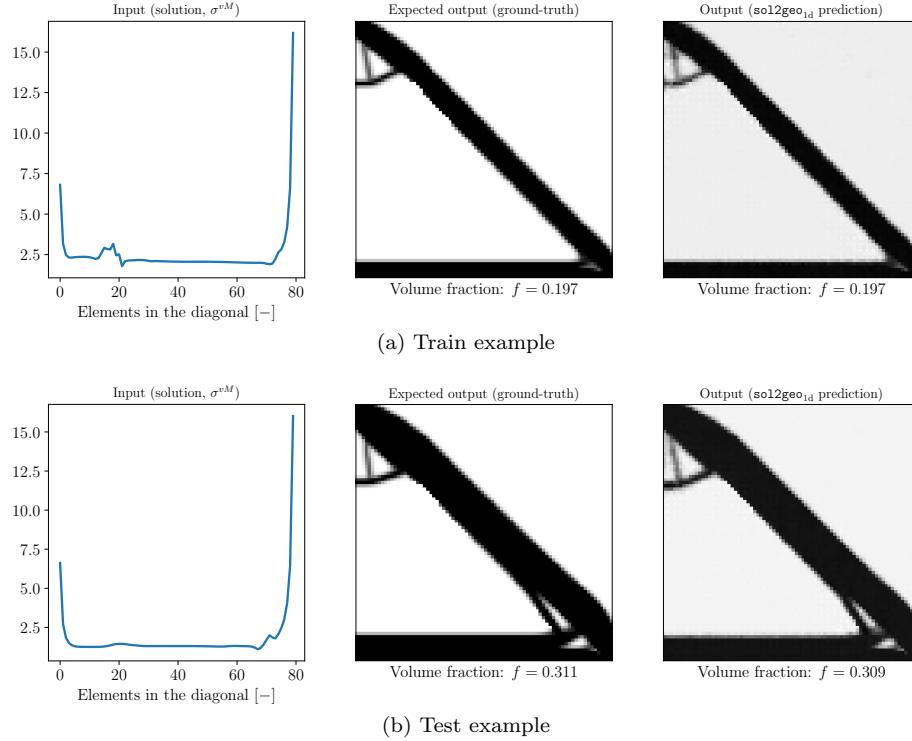


Figure 9: **sol2geo_{1d}** example predictions in (a) train and (b) test. Note the closeness of volume fractions in ground-truth and predicted geometries – exact in the train – despite not having prescribed such constraint in the pipeline.

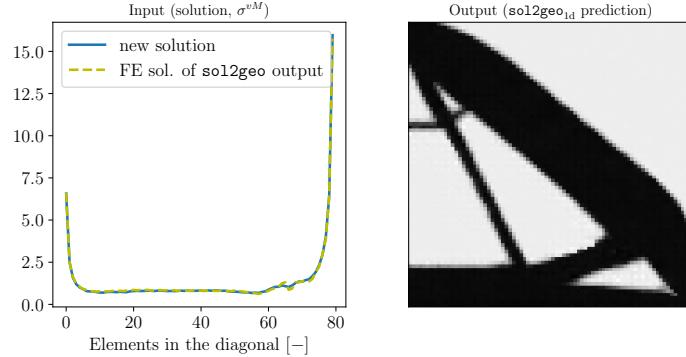


Figure 10: Example to assess the validity of solutions from generated geometries. A new solution (blue, left) is fed to the **sol2geo_{1d}** pipeline, yielding a geometry (right). Then, this geometry is solved via FE, and the σ^{vM} of the diagonal elements i.e., the high-fidelity solution, is represented against the input (dashed yellow, left).

Lastly in this section, when predicting geometries, one can ask whether these geometries are physically reasonable. This is what we intend to evaluate to conclude this section. To do so, a

new solution is going to be tested by comparing it with the actual solution of the (new) generated geometry. To be new, the solution is generated as an interpolation in the latent space of solutions, therefore, the geometry produced in the `sol2geo1d` pipeline will be new—never seen before by the model. This geometry is solved via FE, so a high-fidelity curve related to this geometry is obtained. Finally, this is compared with the input, i.e. the new solution (result of interpolation). This idea is displayed in Figure 10.

5.3. 2d field quantity of interest in a fixed domain

The last case in this paper showcases the direct and inverse problems when the solutions (QoI) are a 2d field defined in the fixed domain Ω , that is:

$$\mathbf{s}_j := \boldsymbol{\sigma}_j^{vM,*},$$

properly rearranged in a matrix i.e., replicating the 2d mesh. In this case, full resolution $\boldsymbol{\sigma}^{vM}$ solutions are predicted using this methodology.

On the one hand, the already trained CNN-RRAE highlighted in Section 5.1 is used as the geometries model. On the other hand, a new CNN-RRAE for solutions is built and trained. The architecture is the same (Table 1), but using a $k_{\max} = 1$ in order to compress in a unique latent coefficient. Training the model with Table 2 parameters, the reconstruction loss of the solutions RRAE is $\mathcal{L}_{\text{train}} = 12.23\%$ in the training set, and its value in the test counterpart is $\mathcal{L}_{\text{test}} = 13.87\%$. Examples for both train and test reconstructions are displayed in Figure 11.

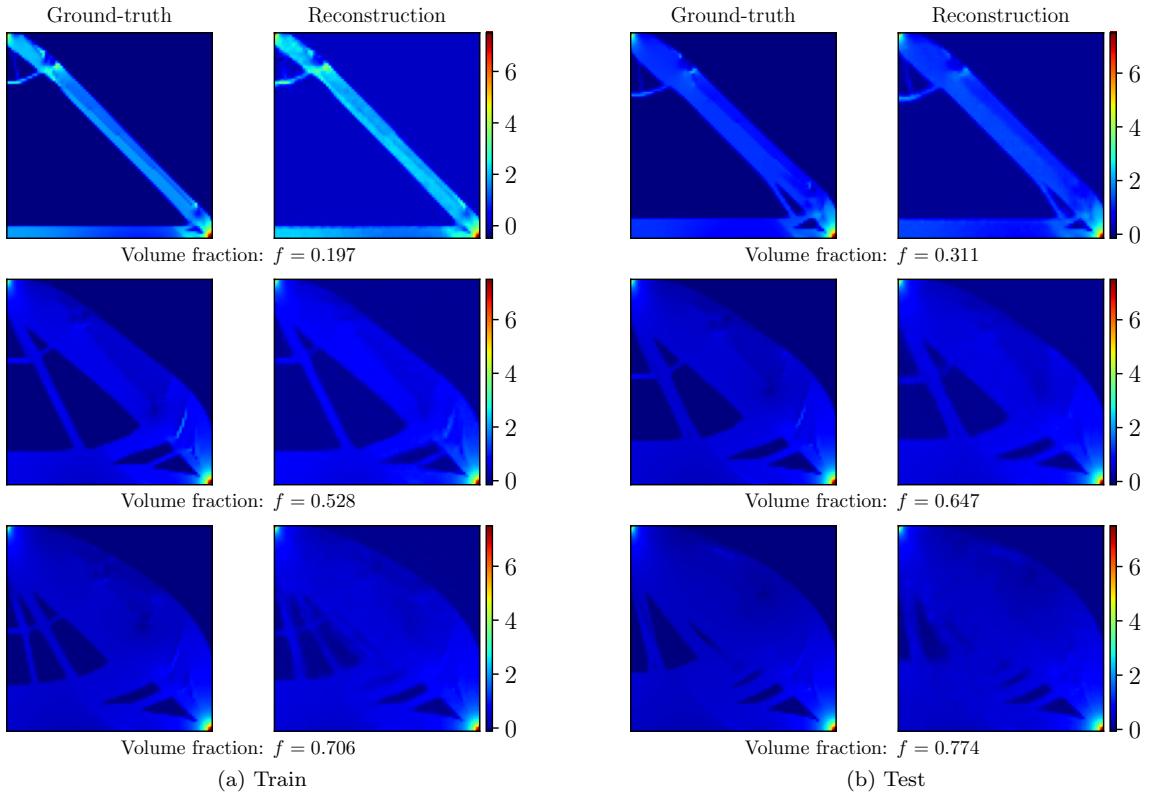


Figure 11: CNN-RRAE 2d solutions (von Mises stress distribution) model. Example reconstructions in (a) train and (b) test sets. For the sake of visualization, the maximum value in the plot is 7.5.

5.3.1. Direct problem

The direct problem is addressed by building a mapping \mathcal{NN}_{2d}^d relating the geometry latent coefficients \mathbf{A} and the new 2d-field solution latent coefficients \mathbf{B} —the same symbol is used, with a slight abuse of notation. Since the architecture is preserved for all the mappings, the same hyper-parameters are used (Table A.1 and Table A.2).

The loss value is $\text{MSE}_{\text{train}} = 1.5 \cdot 10^{-4}$, and coefficients of determination are $R_{\text{train}}^2 = 1.000$, $R_{\text{test}}^2 = 0.999$. Note that since the solution part has more information in this 2d case i.e., it is not compressed like the 1d or scalar case, it is easier to relate the latent coefficients to the solution part, and hence the metrics are better. The latent coefficient β_0 prediction results of \mathcal{NN}_{2d}^d are depicted in Figure A.5.

In this subsection, the pipeline geo2sol_{2d} is built: CNN-RRAE geometries model encoding, map the latent coefficients through \mathcal{NN}_{2d}^d , and decode through the CNN-RRAE 2d solutions model. Once built, the evaluation of the reconstruction loss, Equation (20), between ground-truth an geo2sol_{2d} predicted geometries yields the following:

$$\mathcal{L}_{\text{train}} = 10.39\%, \quad \mathcal{L}_{\text{test}} = 11.59\%.$$

Two examples (train, test) are shown in Figure 12. Note the accuracy of the prediction in how close the maximum stress values σ_{\max}^{vM} are in both ground-truth and geo2sol_{2d} predictions. This may serve as an alternative to the direct scalar QoI case (geo2sol_s).

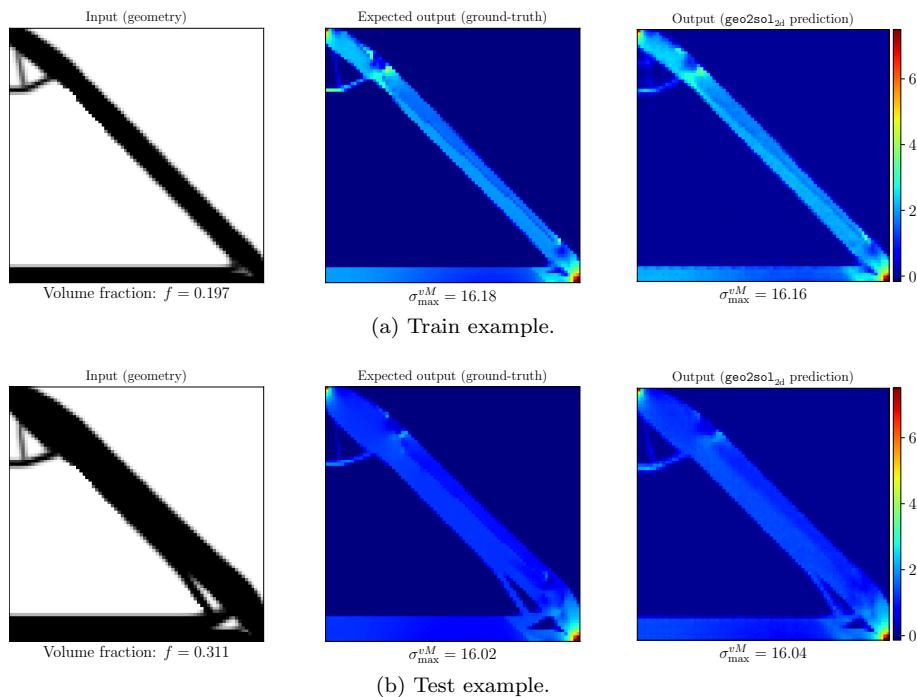


Figure 12: geo2sol_{2d} example predictions in (a) train and (b) test. Note how close are the maximum stress values in ground-truth and predicted stress fields. The maximum value in the plot is 7.5 again.

5.3.2. Inverse problem

The last mapping \mathcal{NN}_{2d}^i is built to map \mathbf{B} 2d solution latent coefficients into \mathbf{A} geometry latent coefficients. Using the same MLP architecture and training parameters as the previous cases, the loss value obtained is $\text{MSE}_{\text{train}} = 6.3 \cdot 10^{-4}$, and the coefficients of determination $R_{\text{train}}^2 = 1.000$, $R_{\text{test}}^2 = 0.999$. The fitting curves are depicted in Figure A.6.

Now, the inverse mapping sol2geo_{2d} with respect to the previous subsection is built. This implies taking the stress distribution in the mesh as input \mathbf{s}_j , encode it via the CNN-RRAE 2d solutions model, map the corresponding $\beta_{0,j}$ coefficient through \mathcal{NN}_{2d}^i into $\mathbf{\alpha}_j$, and decode it with the CNN-RRAE geometries model to obtain the predicted geometry $\hat{\mathbf{x}}_j$. Evaluating the reconstruction loss in the pipeline, between ground-truth an sol2geo_{2d} predicted geometries, yields the following:

$$\mathcal{L}_{\text{train}} = 5.74\%, \quad \mathcal{L}_{\text{test}} = 7.80\%.$$

Examples of train and test showcasing this pipeline are depicted in Figure 13.

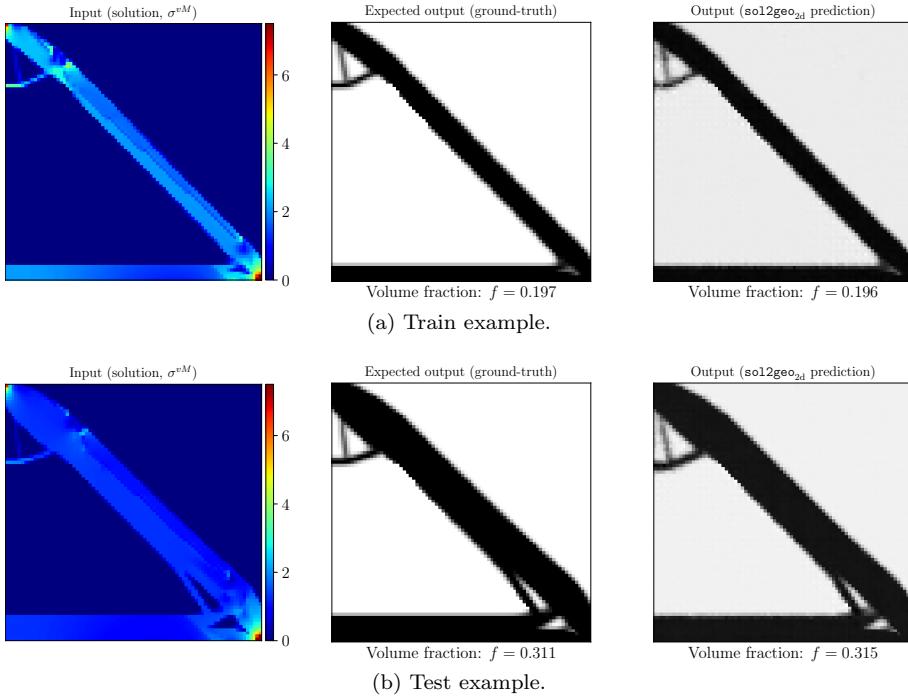


Figure 13: `sol2geo2d` example predictions in (a) train and (b) test. Note how close are the volume fractions in ground-truth and predicted geometries. The maximum value in the plot is 7.5 again.

6. Conclusions and Perspectives

This work has introduced a data-driven framework that combines Rank Reduction Autoencoders (RRAEs) with neural latent-space mappings to address both forward and inverse problems in topology optimization (TO), both of them predicted via regression. By considering a low-rank approximation of the latent space, the proposed methodology enables efficient dimensionality reduction while preserving the dominant physical trends governing optimized geometries and their mechanical responses.

The numerical results demonstrate that the effectiveness of the framework strongly depends on the richness of the selected quantity of interest (QoI). When the response is compressed into a scalar metric, such as the maximum von Mises stress, the resulting geometry-solution mapping becomes inherently non-bijective or ill-conditioned. This limitation is reflected in reduced predictive accuracy, increased sensitivity to extrapolation, and overfitting tendencies in the inverse problem. These observations highlight the intrinsic difficulty of recovering detailed geometric information from highly localized data. In contrast, when the QoI is defined as a one-dimensional or two-dimensional stress field, the proposed approach achieves a substantial improvement in both forward and inverse predictions. The use of an RRAE for the solution fields yields smooth, physically meaningful latent coefficients that correlate well with variations in the design parameters. As a result, the learned mappings between geometry and solution latent spaces exhibit high accuracy and robustness, enabling reliable reconstruction of geometries and responses without enforcing explicit constraints such as volume fraction preservation. The best overall performance is obtained when the full two-dimensional von Mises stress field is considered as the QoI. In this case, the increased informational content of the solution leads to highly accurate latent-space regressions and consistent predictions in both direct and inverse settings. Despite the higher complexity of the data, the RRAE-based compression remains effective, and the reconstructed stress fields and geometries closely match their high-fidelity counterparts. These results suggest that the proposed framework can serve as an efficient surrogate for Finite Element analyses in TO workflows.

Beyond surrogate modeling, the methodology naturally supports Generative Design (GD) capabilities through latent-space interpolation due to the use of the SVD model-order reduction, enabling the synthesis of new geometries and responses that remain physically plausible. Although this aspect was only preliminarily explored, it highlights the potential of RRAEs as a unifying

tool for reduced-order modeling, inverse design, and data-driven exploration or GD of optimized structures.

Future work will focus on extending the framework to multi-parameter design spaces, incorporating varying boundary conditions and loading scenarios, and enforcing physics-informed constraints within the latent mappings. Additionally, integrating the proposed approach into optimization loops and uncertainty-aware design pipelines represents a promising direction for advancing data-driven generative mechanical design.

Appendix A. MLP models details

Appendix A.1. \mathcal{NN}_{1d}^d and \mathcal{NN}_{1d}^i architecture and optimization hyper-parameters

Module	Layer	Input shape	Output shape	Activation
Input layer	Dense	$(n_b, 2)$	$(n_b, 8)$	ELU
Hidden layer #1	Dense	$(n_b, 8)$	$(n_b, 16)$	ELU
Hidden layer #2	Dense	$(n_b, 16)$	$(n_b, 8)$	ELU
Hidden layer #3	Dense	$(n_b, 8)$	$(n_b, 4)$	ELU
Output layer	Dense	$(n_b, 4)$	$(n_b, 1)$	Linear

Table A.1: Architecture of \mathcal{NN}_{1d}^d (and \mathcal{NN}_{1d}^i , permuting input and output shapes), the MLP model relating latent coefficients from geometry to solution in the 1d QoI case.

Parameter	Choice
Optimizer	nadam
Learning rate	10^{-3}
Epochs	3000
Batch size	100
Loss	MSE

Table A.2: \mathcal{NN}_{1d}^d and \mathcal{NN}_{1d}^i MLP models. Optimization parameters of the training.

Appendix A.2. \mathcal{NN}_s^d and \mathcal{NN}_s^i fitting curves

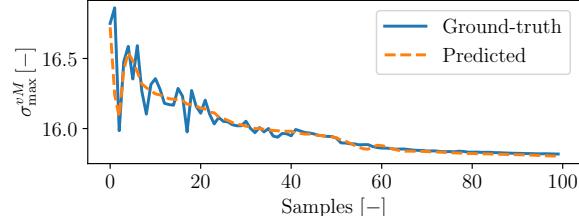


Figure A.1: \mathcal{NN}_s^d MLP model. Prediction of the scalar QoI σ_{\max}^M .

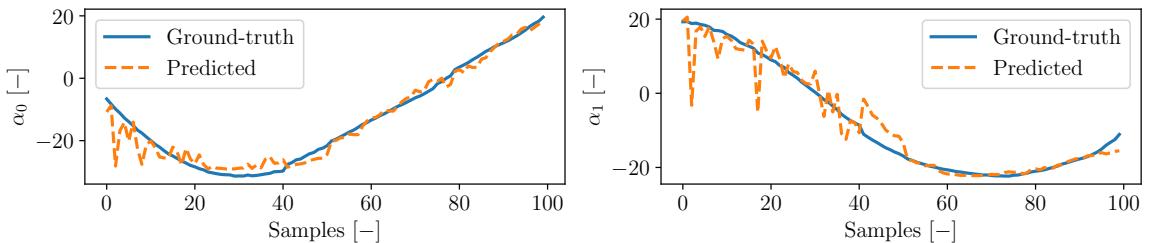


Figure A.2: \mathcal{NN}_s^i MLP model. Prediction of the geometries latent coefficients α_0 and α_1 .

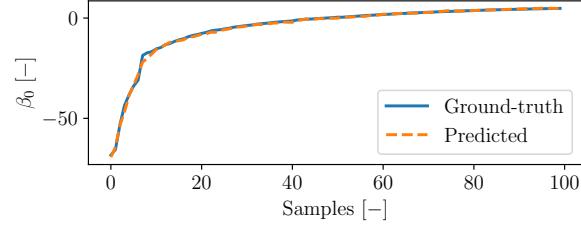


Figure A.3: \mathcal{NN}_{1d}^d MLP model. Prediction of the solutions latent coefficients β_0 .

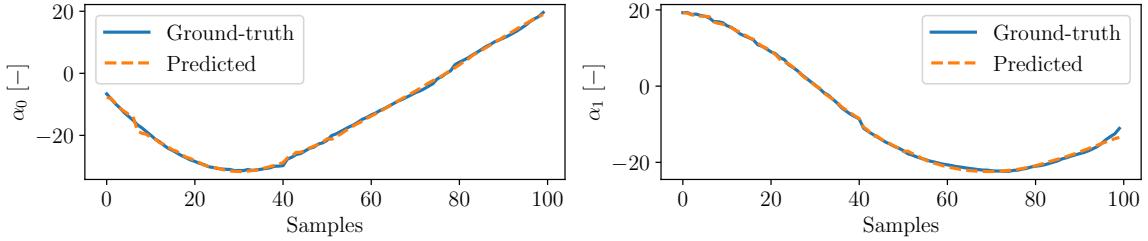


Figure A.4: \mathcal{NN}_{1d}^i MLP model. Prediction of the geometries latent coefficients α_0 and α_1 .

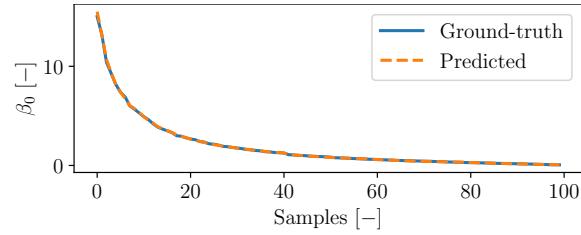


Figure A.5: \mathcal{NN}_{2d}^d MLP model. Prediction of the solutions latent coefficients β_0 .

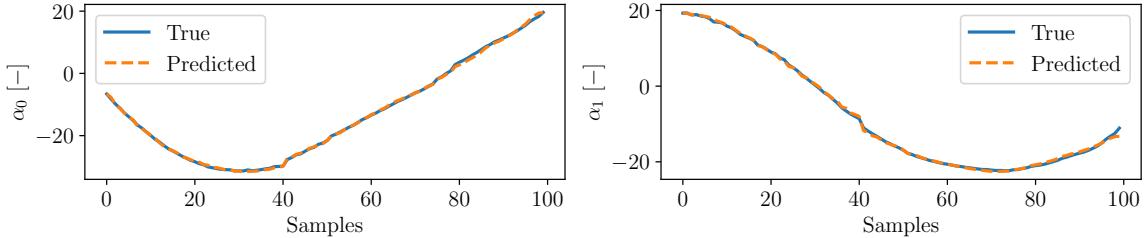


Figure A.6: \mathcal{NN}_{2d}^i MLP model. Prediction of the geometries latent coefficients α_0 and α_1 .

References

- [1] Daozhong Li and Il Yong Kim. Multi-material topology optimization for practical lightweight design. *Structural and Multidisciplinary Optimization*, 58(3):1081–1094, 2018.
- [2] Stefan Junk, Benjamin Klerch, and Ulrich Hochberg. Structural optimization in lightweight design for additive manufacturing. *Procedia CIRP*, 84:277–282, 2019.
- [3] Hongjun Xue, Haiyang Yu, Xiaoyan Zhang, and Qi Quan. A novel method for structural lightweight design with topology optimization. *Energies*, 14(14):4367, 2021.
- [4] Yue Zhang, Yingchun Shan, Xiantong Liu, and Tian He. An integrated multi-objective topology optimization method for automobile wheels made of lightweight materials. *Structural and Multidisciplinary Optimization*, 64(3):1585–1605, 2021.

- [5] Xianmin Zhang and Benliang Zhu. *Topology optimization of compliant mechanisms*. Springer, 2018.
- [6] Ole Sigmund. On the design of compliant mechanisms using topology optimization. *Journal of Structural Mechanics*, 25(4):493–524, 1997.
- [7] Benliang Zhu, Xianmin Zhang, Hongchuan Zhang, Junwen Liang, Haoyan Zang, Hai Li, and Rixin Wang. Design of compliant mechanisms using continuum topology optimization: A review. *Mechanism and Machine Theory*, 143:103622, 2020.
- [8] Soonok Park and Jeonghoon Yoo. Structural optimization of a multi-physics problem considering thermal and magnetic effects. *IEEE transactions on magnetics*, 48(11):3883–3886, 2012.
- [9] David J Munk, Timoleon Kipouros, Gareth A Vio, Geoffrey T Parks, and Grant P Steven. Multiobjective and multi-physics topology optimization using an updated smart normal constraint bi-directional evolutionary structural optimization method. *Structural and Multidisciplinary Optimization*, 57(2):665–688, 2018.
- [10] Sourav Das and Alok Sutradhar. Multi-physics topology optimization of functionally graded controllable porous structures: Application to heat dissipating problems. *Materials & Design*, 193:108775, 2020.
- [11] Martin P Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1:193–202, 1989.
- [12] Junuthula Narasimha Reddy. An introduction to the finite element method. *New York*, 27(14), 1993.
- [13] Junuthula Narasimha Reddy. *An Introduction to Nonlinear Finite Element Analysis: with applications to heat transfer, fluid mechanics, and solid mechanics*. Oxford university press, 2015.
- [14] Yoshua Bengio, Ian Goodfellow, Aaron Courville, et al. *Deep learning*, volume 1. MIT press Cambridge, MA, USA, 2017.
- [15] Jad Mounayer, Sebastian Rodriguez, Chady Ghnatos, Charbel Farhat, and Francisco Chinesta. Rank reduction autoencoders. *arXiv preprint arXiv:2405.13980*, 2024.
- [16] Virginia Klema and Alan Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, 25(2):164–176, 1980.
- [17] Jan J Gerbrands. On the relationships between svd, klt and pca. *Pattern recognition*, 14(1-6):375–381, 1981.
- [18] NK Mani, EJ Haug, and KE Atkinson. Application of singular value decomposition for analysis of mechanical system dynamics. 1985.
- [19] Jad Mounayer, Alicia Tierz, Jerome Tomezyk, Chady Ghnatos, and Francisco Chinesta. Variational rank reduction autoencoder. *arXiv preprint arXiv:2505.09458*, 2025.
- [20] Mohammed El Fallaki Idrissi, Ismael Ben-Yelun, Jad Mounayer, Sebastian Rodriguez, Chady Ghnatos, and Francisco Chinesta. A new framework for generative design, real-time prediction, and inverse design optimization: Application to microstructure. 2025.
- [21] Mohammed El Fallaki Idrissi, Jad Mounayer, Sebastian Rodriguez, Fodil Meraghni, and Francisco Chinesta. Generative parametric design (gpd): A framework for real-time geometry generation and on-the-fly multiparametric approximation. *arXiv preprint arXiv:2512.11748*, 2025.
- [22] Alicia Tierz, Jad Mounayer, Beatriz Moya, and Francisco Chinesta. Variational rank reduction autoencoders for generative thermal design. *Results in Engineering*, page 108418, 2025.

- [23] Jad Mounayer, Sebastian Rodriguez, Jerome Tomezyk, Chady Ghnatos, and Francisco Chinesta. Rraedy: Adaptive latent linearization of nonlinear dynamical systems. *arXiv preprint arXiv:2512.07542*, 2025.
- [24] S Rodriguez, M Rebillat, N Mechbal, A Ammar, and F Chinesta. Damage detection algorithm based on an innovative nonlinear model-order reduction technique: The rank reduction autoencoder (rrae) conditioned to learn damage features. 2026.
- [25] Junuthula Narasimha Reddy. *An introduction to the finite element method*, volume 3. McGraw-Hill New York, 2005.
- [26] Ole Sigmund and Joakim Petersson. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural optimization*, 16:68–75, 1998.
- [27] Krister Svanberg. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, 24(2):359–373, 1987.
- [28] Martin P Bendsøe. *Optimization of structural topology, shape, and material*, volume 414. Springer, 1995.
- [29] Peter W Christensen and Anders Klarbring. *An introduction to structural optimization*, volume 153. Springer Science & Business Media, 2008.
- [30] Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S Lazarov, and Ole Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43:1–16, 2011.