Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 405 (2023) 115857
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Distance-preserving manifold denoising for data-driven mechanics

Bahador Bahmani, WaiChing Sun*

Department of Civil Engineering and Engineering Mechanics, Columbia University, 614 SW Mudd, Mail Code: 4709, New York, NY 10027, United States of America

Received 18 July 2022; received in revised form 21 November 2022; accepted 15 December 2022
Available online 9 January 2023

Abstract

This article introduces an isometric manifold embedding data-driven paradigm designed to enable model-free simulations with noisy data sampled from a constitutive manifold. The proposed data-driven approach iterates between a global optimization problem that seeks admissible solutions for the balance principle and a local optimization problem that finds the closest point projection of the Euclidean space that isometrically embeds a nonlinear constitutive manifold. To de-noise the database, a geometric autoencoder is introduced such that the encoder first learns to create an approximated embedding that maps the underlying low-dimensional structure of the high-dimensional constitutive manifold onto a flattened manifold with less curvature. We then obtain the noise-free constitutive responses by projecting data onto a denoised latent space that is completely flat by assuming that the noise and the underlying constitutive signal are orthogonal to each other. Consequently, a projection from the conservative manifold onto this de-noised constitutive latent space enables us to complete the local optimization step of the data-driven paradigm. Finally, to decode the data expressed in the latent space without reintroducing noise, we impose a set of isometry constraints while training the autoencoder such that the nonlinear mapping from the latent space to the reconstructed constituent manifold is distance-preserving. Numerical examples are used to both validate the implementation and demonstrate the accuracy, robustness, and limitations of the proposed paradigm.
© 2022 Elsevier B.V. All rights reserved.

Keywords: Data-driven mechanics; Manifold de-noising; Geodesic; Constitutive manifold; Autoencoder; Isometry

1. Introduction

The era of big data comes with the great promise that the abundance of material data may enable a data-driven/model-free approach to revolutionize how mechanistic simulations are run and used [1,2]. If constitutive data for solids may one day become abundant and openly shared, the data-driven/model-free approach introduced in [2] may lead to a new ecosystem for simulation-based engineering in which the time-consuming and expensive constitutive modeling can be replaced by a distance-minimizing algorithm. As such, the burden of calibrating phenomenological models [2] and the lack of physical underpinning of internal variables [3–5], and even the need for verification and validation can all be bypassed in the model-free paradigm [6–8].

However, engineering industries and businesses that employ mechanistic simulations are not necessarily operated with access to big data. In many cases, engineering analysis may require access to proprietary, export-controlled,

* Corresponding author.
E-mail address: wsun@columbia.edu (W. Sun).

https://doi.org/10.1016/j.cma.2022.115857
0045-7825/© 2022 Elsevier B.V. All rights reserved.

and even classified data, often in a limited data regime. A low-cost mechanism to acquire large amounts of data with sufficient fidelity through interactions with users is not available. For instance, constitutive data of solids are often proprietary [9] and sometimes even classified or under export controls. In many cases, they could be the primary and permanent assets where constitutive models and solvers often come to better utilize those assets [10]. As such, a model-free approach that requires a lot of data to function is not always practical. Alternative approaches that can operate in a reasonably *robust* manner in both data-rich and data-limited regimes are crucial for real-world applications.

1.1. Relevant work on data-driven paradigm with noisy data

There are a number of published contributions dedicated to handling noisy data for the model-free/data-driven paradigm. For instance, Kirchdoerfer and Ortiz [11] extend the distance-minimization paradigm in [2] to a probabilistic formulation through the lens of maximum-entropy estimation. This formulation, which is referred to as the max-ent Data-Driven Computing, improves the robustness of the projection step by introducing a probabilistic aggregation mechanism for a cluster of data points in the phase space. Since the projection is conducted via the cluster but not from a particular chosen data point, data-driven computing is less sensitive to outliers. In [12], a local embedding approach is introduced where a set of neighbor data points closest to the conservation manifold is identified and used to construct a locally convex hull on the fly. Consequently, this local approach does not require the construction of a global embedding such as those used in [13], while a regularization parameter can be used to reduce the impacts of outliers. In both cases, the robustness toward noisy data is nevertheless at the expense of additional computational costs of the simulated-annealing strategies adopted during the simulations.

Eggersmann et al. [14], on the other hand, incorporate local second-order information regarding the local tangent operator at each material data point calculated by an instance-based method known as tensor-voting. This method is combined with the max-entropy approach to enhance the classical distance-minimization method in dealing with noisy and limited data. Compared to first-order approximations, this method can track the geometrical structure of data with higher precision. However, it may become less effective when given very limited or unevenly distributed data as the voting process relies on the local aggregation of information, which may not be suitable for data points of different proximity or when the data points are far apart from each other. Kanno [15] introduces a kernel method to reconstruct a smooth constitutive manifold that may filter out noise through global interpolation. However, kernel-based methods may become less computationally favorable in the big data regime as they scale cubically with respect to the data size. Ayensa-Jiménez et al. [16] define a statistically consistent distance measure by incorporating higher-order statistical moments into the data-driven computation to address the prediction uncertainty for noisy databases. Finally, He et al. [17] extend this idea by embedding the data in a lower-dimensional latent space obtained via an autoencoder. In this work, the encoder provides the nonlinear dimensionality reduction that maps the high-dimensional data into a low-dimensional space, necessarily filtering out the noise while preserving the nonlinearity of the data.

1.2. Scope of this work

In this research, our goal is to introduce a geometric framework that enables the manifold embedding data-driven paradigm to perform robustly with noisy data. To achieve this goal, we design a geometric autoencoder. While the encoder allows us to first identify the underlying de-noised smooth constitutive manifold through embedding the nonlinear constitutive data onto a pre-designated flat plane, a locally isometric (distance-preserving) decoder is used to prevent overfitting and the re-introduction of noise through the reconstruction procedure. (see Fig. 1).

This setup helps us overcome the non-discriminative distance structure of the high-dimensional noises [18,19] that may otherwise make the decomposition of noise and data difficult. This distance-preserving treatment allows us to preserve the geometric structure of a plausible constitutive manifold without directly dealing with the complexity due to the nonlinearity of the high-dimensional signal and noise. With the isometric embedding established, the data-driven simulation can then be conducted by an iterative scheme that minimizes the distance between the conservation and constitutive manifolds.

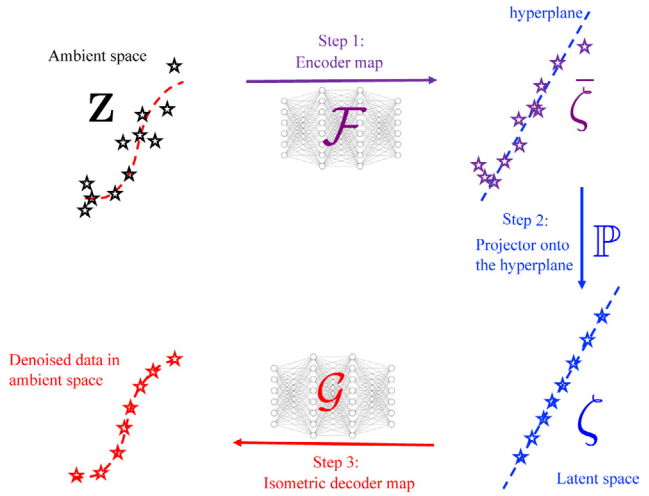


Fig. 1. The schematic representation of each step in the introduced local projection and the manifold recovery from a noisy point cloud in this paper. First, the encoder function maps the ambient point cloud to the latent space that is approximately a hyperplane. Second, the points out of the hyperplane are linearly projected onto the hyperplane. Finally, in the third step, the decoder function maps the hyperplane points onto the ambient space. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

1.3. Organization of the rest of this paper

The organization of the rest of the paper is as follows. Sections 2 and 3 provide the formulation for the global and local minimization problems that constitute the data-driven paradigm. In particular, emphasis is placed on Section 3, where the geometric autoencoder designed to filter out noise is formulated, and the corresponding machine learning problem used to train the neural networks is documented. Section 4 then provides a number of numerical examples to verify the implementation and validate the proposed data-driven method with experimental data. A concluding remark is then provided in Section 5. To avoid overburdening the readers with details, the hyperparameters used to train the embedding as well as other implementation details are provided in the Appendix.

2. Global projection step

The distance-minimization approach leads to a double minimization problem, which can be solved by the alternating minimization algorithm. In this algorithm, the solver repeatedly projects a solution belonging to a set to another set until the convergence is achieved. Following Kirchdoerfer and Ortiz [2], we refer to the projection from the constitutive manifold onto the conservation manifold as the *global* projection. Whereas the counterpart that projects an admissible element from the conservation manifold onto the constitutive manifold is referred to as the *local* projection. In this section, we provide the formulation for the global projection from the data manifold onto the equilibrium manifold. Since the symmetry of the second Piola–Kirchhoff stress is sufficient to enforce the balance of angular momentum, a solution admissible to the balance of linear momentum for finite elasticity is considered to be an element of the equilibrium (or conservation) manifold [2,12].

2.1. Problem statement for the global step

For completeness, we briefly outline the global optimization problem of which the solution is updated iteratively with the one obtained from the local optimization step. For a pair of given strain and stress fields $\boldsymbol{Z}^*(\boldsymbol{X}) = (\boldsymbol{E}^*(\boldsymbol{X}), \boldsymbol{S}^*(\boldsymbol{X}))$, we aim to find solution fields $\boldsymbol{U}(\boldsymbol{X})$, $\boldsymbol{E}(\boldsymbol{X})$, and $\boldsymbol{S}(\boldsymbol{X})$ such that the solution fields satisfy the balance of linear momentum and admissible boundary conditions while they have the minimum *distance* to the given strain and stress fields, i.e., $\boldsymbol{Z}^*$. Mathematically the problem statement can be written as follows:

$$\arg\min_{\mathbf{Z}} \int_{\Omega} D_M^2(\mathbf{Z}, \mathbf{Z}^*)\, d\Omega,$$

$$\begin{aligned} &\text{subject to:} && \nabla^X \cdot \boldsymbol{P} + \bar{\boldsymbol{B}} = \boldsymbol{0} \quad \text{in } \Omega, \\ & && \boldsymbol{P}\boldsymbol{N} = \bar{\boldsymbol{T}} \quad \text{on } \Gamma_P, \\ & && \boldsymbol{U} = \bar{\boldsymbol{U}} \quad \text{on } \Gamma_U, \\ & && \boldsymbol{E} = \frac{1}{2}\{\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}\} \quad \text{in } \Omega, \end{aligned} \tag{1}$$

where $D_M(\cdot,\cdot)$ is a valid distance function defined in the phase space spanned by the Green–Lagrange strain tensor $\boldsymbol{E}$ and the second Piola–Kirchhoff stress tensor $\boldsymbol{S}$, $\boldsymbol{U}$ is the displacement vector in the reference coordinate system, $\boldsymbol{F} = \boldsymbol{I} + \nabla^X \boldsymbol{U}$ is the deformation gradient, $\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S}$ is the first Piola–Kirchhoff stress tensor, $\bar{\boldsymbol{B}}$ is the body force in the reference coordinate system, $\bar{\boldsymbol{T}}$ is the applied traction vector over the external surface of the undeformed body Γ_P, $\boldsymbol{N}$ is the unit normal vector to the external surface of the undeformed body $\partial\Omega$, $\bar{\boldsymbol{U}}$ is the applied displacement vector over the external surface of the undeformed body Γ_U such that $\Gamma_U \cap \Gamma_P = \varnothing$ and $\Gamma_U \cup \Gamma_P = \partial\Omega$, and Ω is the space occupied by material points in the reference coordinate system $\boldsymbol{X}$. In this work, the global optimization step is formulated to minimize the classical energy norm, following the approach of Nguyen and Keip [20] and He et al. [17], i.e.,

$$D_M^2(\boldsymbol{Z}, \boldsymbol{Z}^*; \mathbb{C}) = \frac{1}{2}(\boldsymbol{E} - \boldsymbol{E}^*) : \mathbb{C} : (\boldsymbol{E} - \boldsymbol{E}^*) + \frac{1}{2}(\boldsymbol{S} - \boldsymbol{S}^*) : \mathbb{C}^{-1} : (\boldsymbol{S} - \boldsymbol{S}^*), \tag{2}$$

where the first and second terms incorporate the contributions of strain and stress differences, respectively, and $\mathbb{C}$ is a fourth-order positive definite tensor with minor ($\mathbb{C}_{IJKL} = \mathbb{C}_{JIKL} = \mathbb{C}_{IJLK}$) and major ($\mathbb{C}_{IJKL} = \mathbb{C}_{KLIJ}$) symmetries. Note that this tensor is not necessarily related to the stiffness tensor. It can be viewed as a weighting operator that re-scales the values of the stress and strain component in the norm used for the global optimization step [2,17].

2.2. Euler–Lagrange equation for the global constrained optimization problem

The solution of the constrained optimization problem formulated in Eq. (1) is obtained via a standard finite element method. To obtain the Galerkin form, we formulate the constrained optimization problem via the method of Lagrange multipliers, i.e.,

$$\Pi(\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{Q}) = \int_\Omega D_M^2(\boldsymbol{Z}, \boldsymbol{Z}^*)\, d\Omega + \int_\Omega \boldsymbol{Q}(\boldsymbol{X}) \cdot (\nabla^X \cdot \boldsymbol{P} + \boldsymbol{B})\, d\Omega - \int_{\Gamma_P} \boldsymbol{Q}(\boldsymbol{X}) \cdot (\boldsymbol{P}\boldsymbol{N} - \bar{\boldsymbol{T}}), \tag{3}$$

where $\boldsymbol{Q}$ is the Lagrange multiplier vector field that enforces the balance law; we restrict the displacement solution field $\boldsymbol{U}$ to satisfy the Dirichlet boundary condition exactly, and the strain field $\boldsymbol{E}$ depends on the displacement field via the compatibility relations. The corresponding Euler–Lagrange equations to this optimization problem can be obtained by the corresponding first-order variations as follows:

$$\begin{aligned} \delta\Pi = &\int_\Omega \delta\boldsymbol{U} \cdot \frac{\partial \boldsymbol{E}(\boldsymbol{U})}{\partial \boldsymbol{U}} : \mathbb{C} : (\boldsymbol{E} - \boldsymbol{E}^*)\, d\Omega - \int_\Omega \delta\boldsymbol{U} \cdot \frac{\partial \boldsymbol{F}(\boldsymbol{U})}{\partial \boldsymbol{U}} : (\nabla^X \boldsymbol{Q} \cdot \boldsymbol{S})\, d\Omega \\ &\int_\Omega \nabla^X \delta\boldsymbol{Q} : \boldsymbol{P}\, d\Omega - \int_\Omega \delta\boldsymbol{Q} \cdot \bar{\boldsymbol{B}}\, d\Gamma + \int_{\Gamma_P} \delta\boldsymbol{Q} \cdot \bar{\boldsymbol{T}}\, d\Gamma \\ &\int_\Omega \delta\boldsymbol{S} : [\mathbb{C}^{-1} : (\boldsymbol{S} - \boldsymbol{S}^*) - \boldsymbol{F}^T \cdot \nabla^X \boldsymbol{Q}]\, d\Omega, \end{aligned} \tag{4}$$

where $\delta\boldsymbol{U}$, $\delta\boldsymbol{Q}$, and $\delta\boldsymbol{S}$ are any admissible variations corresponding to the unknown fields. The above derivation is simplified by restricting $\delta\boldsymbol{U} = \delta\boldsymbol{Q} = \boldsymbol{0}$ on the Dirichlet boundary Γ_U, i.e., the admissibility conditions. At the stationary point, the first variation is zero for any admissible variations ($\delta\Pi(\boldsymbol{U}, \boldsymbol{Q}, \boldsymbol{S}) = \Pi(\boldsymbol{U} + \delta\boldsymbol{U}, \boldsymbol{Q} + \delta\boldsymbol{Q}, \boldsymbol{S} + \delta\boldsymbol{S}) = 0\ \forall\ \delta\boldsymbol{U}, \delta\boldsymbol{Q}, \delta\boldsymbol{S}$). As such, the Euler–Lagrange equations reads,

$$\int_\Omega \delta\boldsymbol{U} \cdot \frac{\partial \boldsymbol{E}(\boldsymbol{U})}{\partial \boldsymbol{U}} : \mathbb{C} : (\boldsymbol{E} - \boldsymbol{E}^*)\, d\Omega - \int_\Omega \delta\boldsymbol{U} \cdot \frac{\partial \boldsymbol{F}(\boldsymbol{U})}{\partial \boldsymbol{U}} : (\nabla^X \boldsymbol{Q} \cdot \boldsymbol{S})\, d\Omega = 0, \tag{5}$$

$$\int_\Omega \nabla^X \delta\boldsymbol{Q} : \boldsymbol{P}\, d\Omega - \int_\Omega \delta\boldsymbol{Q} \cdot \bar{\boldsymbol{B}}\, d\Gamma + \int_{\Gamma_P} \delta\boldsymbol{Q} \cdot \bar{\boldsymbol{T}}\, d\Gamma = 0, \tag{6}$$

$$\int_\Omega \delta\boldsymbol{S} : [\mathbb{C}^{-1} : (\boldsymbol{S} - \boldsymbol{S}^*) - \boldsymbol{F}^T \cdot \nabla^X \boldsymbol{Q}]\, d\Omega = 0. \tag{7}$$

For computational purposes, we construct the stress field as the dependent field via the localization theorem in the last equation Eq. (7), which leads to the following relation,

$$\boldsymbol{S} = \boldsymbol{S}^* + \mathbb{C} : (\boldsymbol{F}^T \cdot \nabla^X \boldsymbol{Q}) \quad \text{in } \Omega. \tag{8}$$

As such, the only independent fields are the displacement and out-of-balance Lagrange multiplier vectors with two nonlinear Eqs. (5) and (6). For brevity, the spatial discretization via finite elements is outlined in Appendix C.

3. Local projection step

In this section, we formulate the local optimization problem and introduce the manifold de-nosing algorithm that enables the local projection step necessary to generate incremental updates for the distance-minimization data-driven solver. To keep the simplicity of the original distance-minimization method while eliminating the time-consuming discrete search, Bahmani and Sun [13] introduce an embedding method that performs the local minimization step in an imaginary space that admits the Euclidean structure. However, this framework is not designed to handle noisy databases since the mappings between the ambient and latent spaces are designed to be bijective and hence do not allow any information loss necessary for the filtering. In this work, we relax the bijective requirement for the forward and backward mapping functions and construct the mapping functions in an auto-encoder fashion instead [21]. The overall procedure for the de-noising procedure is summarized in Fig. 1.

3.1. Problem statement for the local step

In the local minimization step of the data-driven paradigm, a solution $\boldsymbol{Z}$ belonging to the conservation manifold is mapped onto the constitutive manifold. Such a mapping must be distance minimizing in which the distance is measured by an appropriate metric $D_{\mathcal{D}}$ consistent with the data set $\mathcal{D}$. Kirchdoerfer and Ortiz [2] use the same energy norm used in the global optimization step to assign each integration point the optimal stress-strain pair of the discrete data set $\mathcal{D}$. Kanno [15] argues that, although the metric structure of the constitutive manifold is generally not known *a priori*, it can be approximated via a kernel method if the data points distribute ubiquitously in the phase space. However, the presence of noise might make it difficult to reconstruct the underlying manifold robustly. In addition, solving the distance minimization optimization problem with the presence of noise on the manifold itself could be time-consuming and non-trivial [22].

To address these issues, we propose a geometric approach where we construct a *flattened* latent space that can be isometrically mapped back to a de-noised constitutive manifold. To further simplify the local optimization problem, we also introduce additional constraints and a projection step such that the resultant de-noised latent space is flat, with the stress and strain components related linearly.

This linearity enables us to minimize the distance between the points in the conservation manifold determined from the global iterative step and the latent space that represents the constitutive manifold through a projection. Since the flattened latent space has the same normal everywhere, the local optimization problem can be advanced through an analytical solution. Note that, in addition to the global embedding strategy employed in this paper, one may also leverage the simpler metric of Euclidean space by embedding a subset of constitutive data locally (cf. He et al. [17]).

As depicted in Fig. 1, a projection that maps the vector $\boldsymbol{\zeta}$ onto the hyperplane of the constitutive law through the normal vector of the hyperplane. Consequently, the local minimization step is formulated in the latent space, which reads,

$$\underset{\boldsymbol{\zeta}^* \in \mathcal{L}}{\arg\min} \quad d^2_{\mathcal{L}}(\boldsymbol{\zeta}, \boldsymbol{\zeta}^*), \tag{9}$$

where $d_{\mathcal{L}}$ can be any norm equivalent to the Euclidean norm of the same dimension, $\boldsymbol{\zeta}^* = \mathbb{P}(\mathcal{F}(\boldsymbol{Z}^*))$ and $\boldsymbol{\zeta} = \mathbb{P}(\mathcal{F}(\boldsymbol{Z}))$ are the local updated constitutive responses admissible to $\mathcal{L}$ and that of the conversation manifold accordingly. The de-noised latent space is obtained in an offline manner through the construction of a composite mapping $\mathcal{F}$ and projector $\mathbb{P}$.

The constructions of $\mathcal{F}$ and $\mathbb{P}$ and the related isometric mapping $\mathcal{G}$ that maps the de-noised latent space back to the phase space with data on an underlying de-noised manifold for a given set of discrete data points will be discussed further in Section 3.2.

3.2. Design of the geometric autoencoder

Our goal is to train neural networks to obtain a pair of mapping functions $\mathcal{F}$ and $\mathcal{G}$ showcased in Fig. 1 by leveraging the expressivity power of deep neural networks [23,24]. We consider a point cloud of stress-strain pair data for path-independent elasticity problems in which the phase space is spanned by the independent components of the symmetric stress and strain tensors. Here we introduce unsupervised learning to generate a pair of mapping functions parameterized by two deep neural networks $\mathcal{F}_{\boldsymbol{\theta}} : \mathbb{R}^{2n} \mapsto \mathbb{R}^{2n}$ and $\mathcal{G}_{\boldsymbol{\omega}} : \mathbb{R}^{2n} \mapsto \mathbb{R}^{2n}$ where $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ are their unknown parameters, and n is the number of independent components in a symmetric second order tensor (see Remark 1). Similar to [21], we call $\mathcal{F}$ and $\mathcal{G}$ *encoder* and *decoder* functions, respectively (see Fig. 1). The vector $\boldsymbol{Z} = [\boldsymbol{E}, \boldsymbol{S}]$ is an element of the phase space that concatenates the vector representation of strain and stress tensors in Mandel's notation.

3.2.1. Geometric encoder and the flattened latent space

The encoder function maps elements in the ambient space to the latent space, i.e., $\bar{\boldsymbol{\xi}} = \mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{Z})$, where the text colors hereafter are consistent with those used in Fig. 1 for clarity. We then denote a set of discrete data points $\boldsymbol{Z}^i \in \mathcal{M}$ where $\mathcal{M}$ is assumed to be a Riemannian manifold, but this same set of points can also be described in the $\mathbb{R}^{2n}$ phase space where $\boldsymbol{Z}^i$ can be considered as the position vectors of data points in the phase space.

The linear projector $\mathbb{P}$ projects points in the ambient space onto the closet points on the target hyperplane, i.e., $\boldsymbol{\zeta} = \mathbb{P}(\bar{\boldsymbol{\xi}})$, and essentially reduces the dimension of the constitutive manifold by filtering data stored in the orthogonal basis out. This step assumes that the hyperplane stores the de-noised data such that the noise and the data are orthogonal to each other in the latent space.

The assumption that the noise and data are orthogonal to each other comes from the orthogonality principle, which states that the best approximation of a vector $\boldsymbol{x}$ of the ambient space V by a vector $\boldsymbol{y}$ of a subspace $W \subseteq V$ of this ambient space V is to have the error vector $\boldsymbol{e}$ being orthogonal to the element of a closed subspace W where $\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{e}$. Geometrically, this setting essentially minimizes the magnitude of the error vector, as any admissible vector $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{y}$ would be of greater magnitude if $\boldsymbol{y}$ and $\boldsymbol{e}$ are not orthogonal with each other. As such, the orthogonality principle is a necessary and sufficient condition for the optimality of a Bayesian estimator (cf. Moon and Stirling [25]).

In our work, we first apply the manifold hypothesis to the constitutive signal data. In other words, we assume all the signal data are elements of a Riemannian manifold. While there is no global orthogonal basis for a manifold, the local tangential space of the constitutive manifold is spanned by a set of orthogonal bases. Recall that we aim to enforce the decoder to be isometric. An important property of such an isometric mapping is that it is also conformal (cf. Lang [26]), which means that any angle between two vectors of the tangential space of the constitutive data is preserved when they are mapping from the latent space to the final space of the decoder. As such, the normal vectors that are orthogonal to the constitutive sub-manifold in the ambient space remain orthogonal in the latent space. Consequently, this enables us to apply the orthogonality principle to obtain the best-estimated data.

Note that the true hyperplane of the ambient space would be of n dimensions. However, in our neural network approach, the mapping only constrains the latent space to resemble a hyperplane through a constraint (see Eq. (14)). Hence, the dimension of the coordinate system that expresses the latent space remains of $2n$ dimension, but the intrinsic dimension of the latent space is reduced to n due to the flatness of the latent space. For example, a plane in $\mathbb{R}^3$ can be described in a 3D coordinate system, while a 2D coordinate system is sufficient. In the classical autoencoder in which, the dimensions of the embedding layer may affect the amount of data being filtered out and hence affect how signals and noises are decomposed. In our case, the weighting function of the autoencoder may also affect the decomposition, as shown in the parametric study in Appendix E.

3.2.2. Isometry decoder

After the de-noising step (see Fig. 1), we want the decoder step to be a local isometry and diffeomorphism such that the geodesic can be preserved. Here, the diffeomorphism is enforced by the reconstruction error, which will be defined later. Meanwhile, to ensure the decoder generates mapping that preserves the local isometry, a new constraint is added to the loss function of the autoencoder, i.e., Eq. (15).

Recall that a decoder function maps the point cloud data from the latent space onto the ambient space, i.e.,

$$\boldsymbol{Z}^i = \mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta}^i). \tag{10}$$

Here we assume that the local tangent space of the constitutive manifold $T_z\mathcal{M}$, the phase space, and the latent space (i.e., the approximated hyperplane $T_\zeta\mathcal{N}$) are all Euclidean space of dimension $2n$. As such, a necessary condition to fulfill Eq. (28) is to restrict the tangent operator of the decoder function, i.e.,

$$\boldsymbol{J}(\boldsymbol{\zeta};\boldsymbol{\omega}) := \partial \mathbf{Z}/\partial \boldsymbol{\zeta} = \partial \mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta})/\partial \boldsymbol{\zeta}, \tag{11}$$

such that $\boldsymbol{J}(\boldsymbol{\zeta};\boldsymbol{\omega})^T = \boldsymbol{J}(\boldsymbol{\zeta};\boldsymbol{\omega})^{-1}$; see Appendix B for the proof . In other words, $\boldsymbol{J}(\boldsymbol{\zeta};\boldsymbol{\omega})$ is an orthogonal matrix (cf. [27]). For instance, if a set of points on a curve in $\mathbb{R}^2$ is mapped onto a straight line isometrically, this constraint can be viewed as enforcing the mapping $\mathcal{G}_{\boldsymbol{\omega}}$ to be a pure bending without any stretch or compression.

3.3. Training of neural network autoencoder

The modified autoencoder we used in this paper can be trained via the following weighted multi-objective optimization statement:

$$\underset{\boldsymbol{\theta},\boldsymbol{\omega}}{\arg\min} \quad w_{\text{recn}}\mathcal{L}_{\text{recn}} + w_{\text{lin}}\mathcal{L}_{\text{lin}} + w_{\text{iso}}\mathcal{L}_{\text{iso}} \tag{12}$$

$$\mathcal{L}_{\text{recn}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta}^i) - \mathbf{Z}^i\|_2^2 \tag{13}$$

$$\mathcal{L}_{\text{lin}} = \frac{1}{N}\sum_{i=1}^{N} \|\bar{\boldsymbol{\zeta}}^i_{n:2n} - \boldsymbol{K}\bar{\boldsymbol{\zeta}}^i_{1:n}\|_2^2 \tag{14}$$

$$\mathcal{L}_{\text{iso}} = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{J}^T(\boldsymbol{\zeta}^i)\boldsymbol{J}(\boldsymbol{\zeta}^i) - \boldsymbol{I}_{2n\times 2n}\|_F^2 \tag{15}$$

where $\bar{\boldsymbol{\zeta}}_{1:n}$ indicates the first n components of $\bar{\boldsymbol{\zeta}}$, $\boldsymbol{I}_{2n\times 2n}$ is the identity matrix of size $2n$, $\|\cdot\|_2$, and $\|\cdot\|_F$ are the L^2-norm and the Frobenius norm, respectively. The weighting parameters w_{recn}, w_{lin}, and w_{iso} control the contribution of errors associated with the **recon**struction, **lin**earity, and **iso**metric conditions, respectively. The first term (Eq. (13)) is introduced to enforce the consistency of the forward and backward operations between latent and ambient spaces, i.e., $\mathcal{G} = (\mathcal{F}\circ\mathbb{P})^{-1}$. With the presence of noise, this objective must be supplemented with constraints that filter out the noise. The second term (Eq. (14)) enforces the linearity constraint in the latent space such that the mapped data points are admissible to the designated hyperplane where the stress-strain relation becomes linear. Intuitively, if this term is enforced successfully, then the phase space should deform in such a way that the stress-strain curve in the deformed configuration of the phase space becomes linear. Note that the matrix $\boldsymbol{K}$ is assumed to be symmetric, positive-definite, and known a priori. This matrix controls the hyperplane properties in the latent space (see Section 3.4.) The orthogonal linear projector $\mathbb{P}$ is defined with respect to this matrix as follows [13]:

$$\boldsymbol{\zeta} = \mathbb{P}(\bar{\boldsymbol{\zeta}}) \quad \text{s.t.} \tag{16}$$

$$\boldsymbol{\zeta}_{1:n} = \frac{1}{2}(\bar{\boldsymbol{\zeta}}_{1:n} + \boldsymbol{K}^{-1}\bar{\boldsymbol{\zeta}}_{1:n},) \tag{17}$$

$$\boldsymbol{\zeta}_{n:2n} = \boldsymbol{K}\boldsymbol{\zeta}_{1:n}. \tag{18}$$

The third term (Eq. (15)) constrains the decoder mapping such that it preserves the distance between two close points in the same tangent space $T_z\mathcal{M}$ upon the mapping.

Note that the issue of gradient conflict may arise (cf. Yu et al. [28] and Bahmani and Sun [29]) due to the potential gradient conflict among the three objectives. As shown schematically in Fig. 1, the encoder merely maps the point clouds onto an approximated hyperplane, then a linear projection that assumes the orthogonality of the data and noise in the latent space filters out the noise portion of the data. Then, the reconstruction of de-noised data in the original phase space is obtained by enforcing local isometry of the mapping between the latent and the reconstructed space to ensure that the distance of any two data points remains the same upon transformation.

In summary, the de-noising procedure hypothesizes that the dimension orthogonal to the hyperplane is in the space of the noise. Hence, together with the nonlinear dimensional reduction afforded by the encoder, and the isometric reconstruction enabled by the decoder, the constitutive manifold can be de-noised in a nonlinear manner. Just as the classical autoencoder used for de-noising of data-driven paradigm (e.g., [30]), the proposed

method requires careful tuning of the hyperparameters. In our case, the weighting parameters w_{recn}, w_{lin}, and w_{iso} are additional hyperparameters where any changes of them may affect how high-frequency data may be perceived as noise or data. The implications of these hyperparameters to filter out data are explored both in the numerical experiments in Section 4 and in a detailed parametric study in Appendix E. More systematic studies on hyperparameter tuning are important but are out of the scope of this study.

3.4. Neural network architecture and implementation details

In this work, we parametrize the encoder and decoder functions by the residual neural networks [31]. With a slight abuse of notation, the superscript indicates the output of each residual block, the ith residual block $\mathcal{F}^i_{\boldsymbol{\theta}} : \boldsymbol{Z}^i \mapsto \boldsymbol{Z}^{i+1}$ has the following structure,

$$\boldsymbol{Z}^{i+1} = \boldsymbol{Z}^i + \bar{\mathcal{F}}^i_{\boldsymbol{\theta}}(\boldsymbol{Z}^i), \tag{19}$$

where $\bar{\mathcal{F}}^i$ is the multilayer perceptron (MLP) [32]. The residual neural network with l residual blocks can be written as:

$$\mathcal{F}_{\boldsymbol{\theta}} := \mathcal{F}^l_{\boldsymbol{\theta}} \circ \mathcal{F}^{l-1}_{\boldsymbol{\theta}} \cdots \circ \mathcal{F}^1_{\boldsymbol{\theta}}, \tag{20}$$

where $\circ$ indicates function composition. We prefer this class of neural networks due to their stability regarding exploding and vanishing gradient issues in the case of deeper architectures and better performance in learning bijective transformations [31,33,34].

Although a neural network may exhibit sufficient expressibility to potentially represent a given function upon training, successfully training such a neural network could remain challenging. When the data manifold is too complex, finding the mapping $\mathcal{F}$ and $\mathcal{G}$ could be challenging. One potential reason for the increased difficulty is the non-convexity of the multi-objective optimization problem, in which we are only solved via a vanilla first-order optimizer such as the stochastic gradient descent. Similar to domain decomposition methods, one way to reduce the complexity is to explicitly create coordinate charts and atlas such that the local embeddings of local regions of a constitutive manifold are learned and then stitched back together by enforcing the consistency of co-domains [35,36]. However, such an approach may introduce other challenges, such as determining the appropriate numbers of local patches and ensuring the closure of the learned manifold.

Similar to our previous work [13], the linearity constraint is inspired by Hooke's law and its Euclidean structure. If two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are related by $\boldsymbol{v} = \boldsymbol{K}\boldsymbol{u}$ where $\boldsymbol{K}$ is a symmetric positive-definite matrix, then there exists at least one hyperplane parameterized by its unit normal vector $\boldsymbol{c}$ such that $\boldsymbol{p}^T\boldsymbol{c} = 0$ where the vector $\boldsymbol{p}^T = [\boldsymbol{u}^T, \boldsymbol{v}^T]$ is the concatenation of the two vectors. For example, if $\boldsymbol{K} \in \text{SO}(m)$, then $\sqrt{2m}\boldsymbol{c}^T = [\mathbf{1}^T_m\boldsymbol{K}, \mathbf{1}^T_m]$ is the hyperplane unit normal vector where $\mathbf{1}_m$ is the vector of size m with all components equal to 1. This correspondence between the hyperplane concept and the linearity condition ensures the Euclidean structure of the latent space hence the validity of the Euclidean metric in the latent space.

Remark 1 (*Dimensionality and Geometry*)**.** Theoretically, a perfect embedding that maps the constitutive data onto the latent space may reduce the dimension by 1 for each affine equation added. In Eq. (14), the linearity constraint introduces N number of affine equations where N is the number of individual components of strain or stress tensors, and $2N$ is the dimensions of the phase space of the stress-strain data such that there is a linear relationship between the stress and strain in the latent space. As such, if the training is successful, the latent space is of a lower dimension (N) than that of the ambient space ($2N$). Note that this dimensionality reduction is not enforced explicitly by reducing the number of neurons assigned to the bottleneck middle layer in the autoencoder. Rather, it is achieved in a geometric sense such that the flattened manifold exhibits a geometric structure of lower dimensionality (and hence may *potentially* parameterize with fewer parameters.)

4. Numerical examples

In this section, we introduce three representative numerical examples to verify the implementation, demonstrate how the de-noising data-driven paradigm works, and examine the performance of the models while given noisy data, i.e.,

1. In Section 4.1, we demonstrate how the isometrically constrained autoencoder can be trained to generate an embedding.
2. In Section 4.2, we verify the implementation of the de-noising data-driven paradigm by designing a boundary value problem with a known solution, and using it to test whether the paradigm is able to produce the correct solution with noisy data.
3. In Section 4.3, we examine the performance of the de-noising data-driven paradigm across the limited data and big data regimes via a two-dimensional cantilever beam problem under pure bending. Noisy datasets with different sizes and noise of different amplitudes are synthesized to compare the convergence of different schemes with respect to the data size.
4. In Section 4.4, we assess the robustness and accuracy of the proposed model to handle noisy experimental data that are inherently of higher dimensions due to the anisotropy with a set of biaxial plane-stress loading tests commonly found in biomechanics literature.

Unless otherwise specified, the weighting functions for each constraint in the neural network training loss function are set to 1 in these numerical examples. In the numerical experiments where different neural network strategies are compared, we purposely use the same set of hyperparameters to ensure these results remain comparable. For brevity, the hyperparameters used to train the neural network are provided in Appendix D.

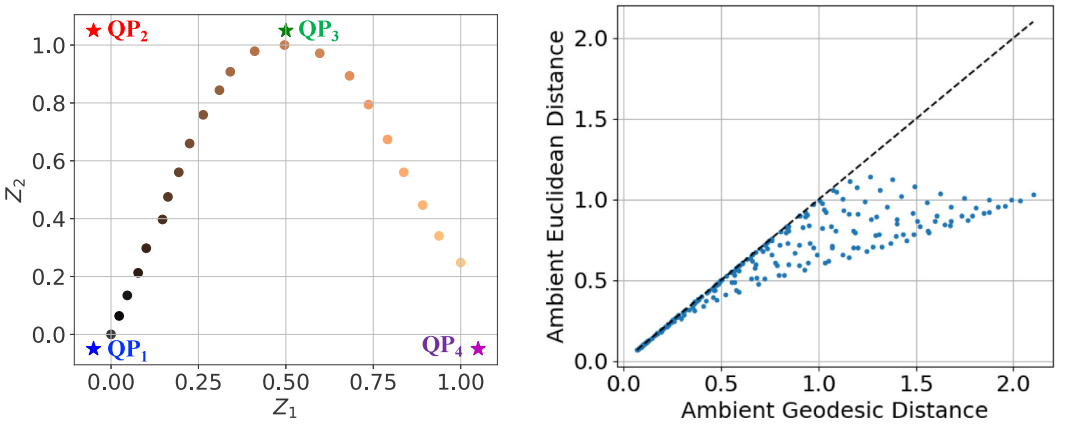
Remark 2 (*Noise Generator*)**.** In the numerical examples where synthetic noisy data are used, we use a Gaussian noise procedure utilized in other published work of data-driven mechanics (cf. Kirchdoerfer and Ortiz [2], Kanno [37], He and Chen [12] and He et al. [30] so that the performance of our work can be assessed and compared. To assess the performance of the proposed approach against real noisy data, we introduced the last example, where unfiltered raw data was collected from experiments collected by Jett et al. [38,39]. In the latter case, the noise structure is not known prior. In engineering practice, noise properties are sensitive to how these data are collected. For instance, noise could be of non-Gaussian nature [40–42] and exhibits scale fluctuation [43]. Since our focus of this paper is not on generating the most realistic noise obtained from sensors or experiments, we did not attempt to emulate these scenarios with more advanced noise generators. Note that the generative autoencoders are sometimes trained with purposely generated noise sampling from a parametrized distribution to regularize the data distribution in the latent space. In such a case, the nature of the noise may have implications on the representation capacity of the latent space (cf. [44]. Examining, generating, and leveraging different types of noise for autoencoders is a highly active research area but is out of the scope of this study. Readers may refer to Ho et al. [45] and Nichol and Dhariwal [46] for details.

Remark 3 (*Physics Constraints for Machine Learning*)**.** In this paper, we have not imposed additional constraints to incorporate the physics-based priors such as poly-convexity, ellipticity, material frame indifference, etc. [47–53]. Enforcing those constraints may improve the robustness of the de-noising procedure. While such treatment may not be necessary for a high-quality data set, it could be crucial in practice for ensuring the robustness of the de-noising process. Such an extension is not considered in this paper but will be considered in the future.

4.1. Demonstrative one-dimensional example

In this first example, we use a synthetic dataset to demonstrate the effect of the isometry constraint on the learned mappings between latent and ambient spaces. Fig. 2(a) shows the set of data points of the constitutive response and the query points that could originate from the global optimization step. Here, we hypothesize that the constitutive data points belong to a sub-manifold $\mathcal{M} \subseteq \mathbb{R}^2$.

In the proposed data-driven paradigm, query points (star points in this figure) are the out-of-manifold points that we aim to project onto the constitutive manifold, which is a geometric prior learned from the stress-strain data. As such, one must introduce a distance measure to formulate a distance-minimization problem at the local step. Not surprisingly, such a distance-minimization problem can be sensitive to the assumed geometry of the data. Here, we conduct a simple experiment where we simply use the $L_2 norm$ of $\mathbb{R}^2$ to calculate the distance between two data points, which we denoted as Ambient Euclidean distance. Then we construct a simple constitutive manifold by jointing points with line segments. This enables us to compute the geodesic distance by summing the length of segments on the shortest path between the two points in the point cloud. As shown in Fig. 2(b), the Euclidean



(a) data points in the ambient space **(b)** pairwise distances

Fig. 2. (a) the database and query points (QP) are shown by circles and star markers, respectively. The points are colored for ease of tracking in each step in the local projection process. (b) A comparison between the discrete geodesic distances and Euclidean distances for randomly picked pairs of data points. Each point in (b) represents two distance measures for a chosen pair of data points (i.e., circle points in (a)). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

norm and the geodesic distance are close to each other when the two points are close to each other. However, the difference between these two measures becomes more significant for points further apart. This difference indicates that the metrics equipped for different data geometry may affect the notion of distance. This discrepancy may affect the choice of the admissible solution in the data-driven paradigm.

After training the encoder and decoder functions, we plot the output of each step in the proposed local projection scheme in Figs. 3 and 4 for the isometric and non-isometric mapping functions, respectively. As the results show, some data points, including the query points, may not lay down on the target hyperplane by the encoder mapping function $\mathcal{F}$ in the first step (see Fig. (b)). This discrepancy could be attributed to the noise, the unsuccessful satisfaction of the linearity condition, or the long distance between the data points and the underlying manifold (e.g., query point 2). In the second step, the closest-point linear projection $\mathbb{P}$ projects all points onto the hyperplane. In the third and last step, the points on the hyperplane are mapped to the real latent space via the decoder function $\mathcal{G}$. Per the reconstruction step (i.e., Step 3), we observe that both schemes recover almost the same manifold. But their behaviors differ slightly for those query points too far away from the manifold, i.e., query points 2 and 4 in Figs. 3(d) and 4(d). Recall that this reconstruction step is equivalent to the projection onto the discovered manifold. One may ask which projection is more accurate for those query points too far away from the underlying manifold? The answer depends on the distance measure assumption in the ambient space which we aim to avoid in this study. However, all the distance measures converge to the Euclidean distance as the points get closer. Therefore, in solving a real BVP, we suggest applying the external loads with sufficiently small increments to avoid inconsistency between the manifold structure and the final results; this helps to not deal with query points too far away from the underlying admissible manifold.

Fig. 5 compares the calculated Euclidean distances in the embedding (latent) space with the geodesic distances in the ambient space. As the results suggest, the isometric training is able to construct a latent Euclidean space with the same geometrical structure as the ambient space. However, the distance is not preserved by the non-isometric training, especially for points too far away from each other. Notice that this feature opens some other interesting directions for future studies, such as 1- performing a nearest neighbor search in the latent space that has fewer dimensions; hence it may accelerate the search engine for high dimensional data, and 2- a consistent hybridization of the fully model-free approach (for regains of the phase-space that has sufficiently high-quality data) and the proposed scheme in the **latent** space.

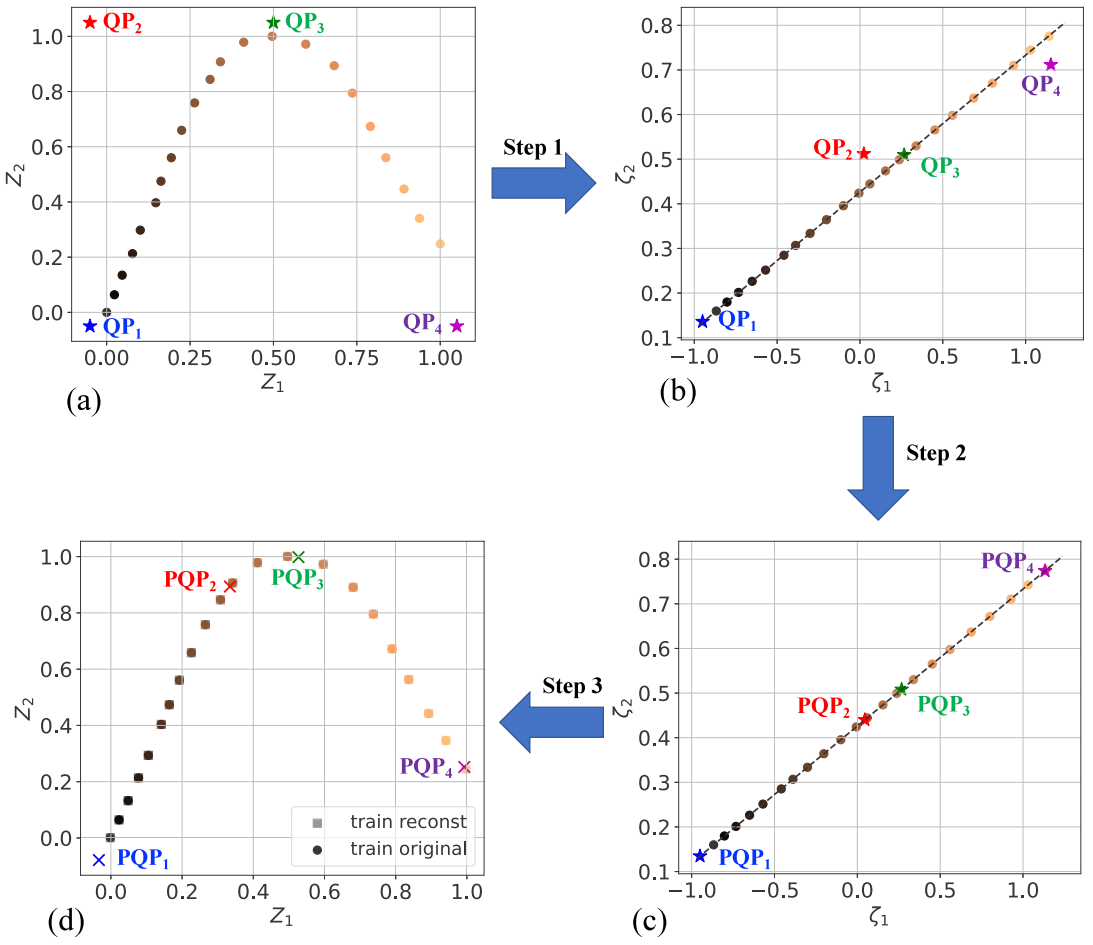


Fig. 3. Local projection steps for the *isometric* training. The steps are consistent with those in Fig. 1. These results are obtained after training the encoding and decoding functions.

4.2. *Verification exercise via a manufactured one-dimensional solution*

In this second example, we introduce a simple one-dimensional boundary value problem (see Fig. 6) with a known analytical solution to (1) verify the implementation and (2) test whether the data-driven algorithm may deliver the correct solution when given noisy data.

As such, we generate a synthetic data set via a 1D version of the neo-Hookean model. We then regarded the data generated from the neo-Hookean model as the ground truth signal and purposely injected Gaussian noise in the data set such that the resultant data is the sum of the ground truth data and noise.

In the one-dimensional case, the 2nd Piola–Kirchhoff stress S is related to the Green-Lagrangian strain $E = \frac{1}{2}(F^2 - 1)$ by the following relationship:

$$S = 2(C_1(\lambda - \lambda^{-2}) + C_2(1 - \lambda^{-3})), \tag{21}$$

where $\lambda = \sqrt{2E + 1}$ is the principal stretch, $F = 1 + \frac{dU(X)}{dX}$ is the deformation gradient, and $C_1 = 10$ MPa and $C_2 = 1$ GPa are material constants.

To obtain the parabola solution field $U(X) = 0.5X^2$, following the method of manufactured solution [54], the body force field $B(X)$ is obtained by substituting the parabola displacement field into the balance of linear momentum for the bar domain, i.e., $B(X) = -\frac{dS(E(X))}{dX}$. The explicit form of the body force is obtained via the

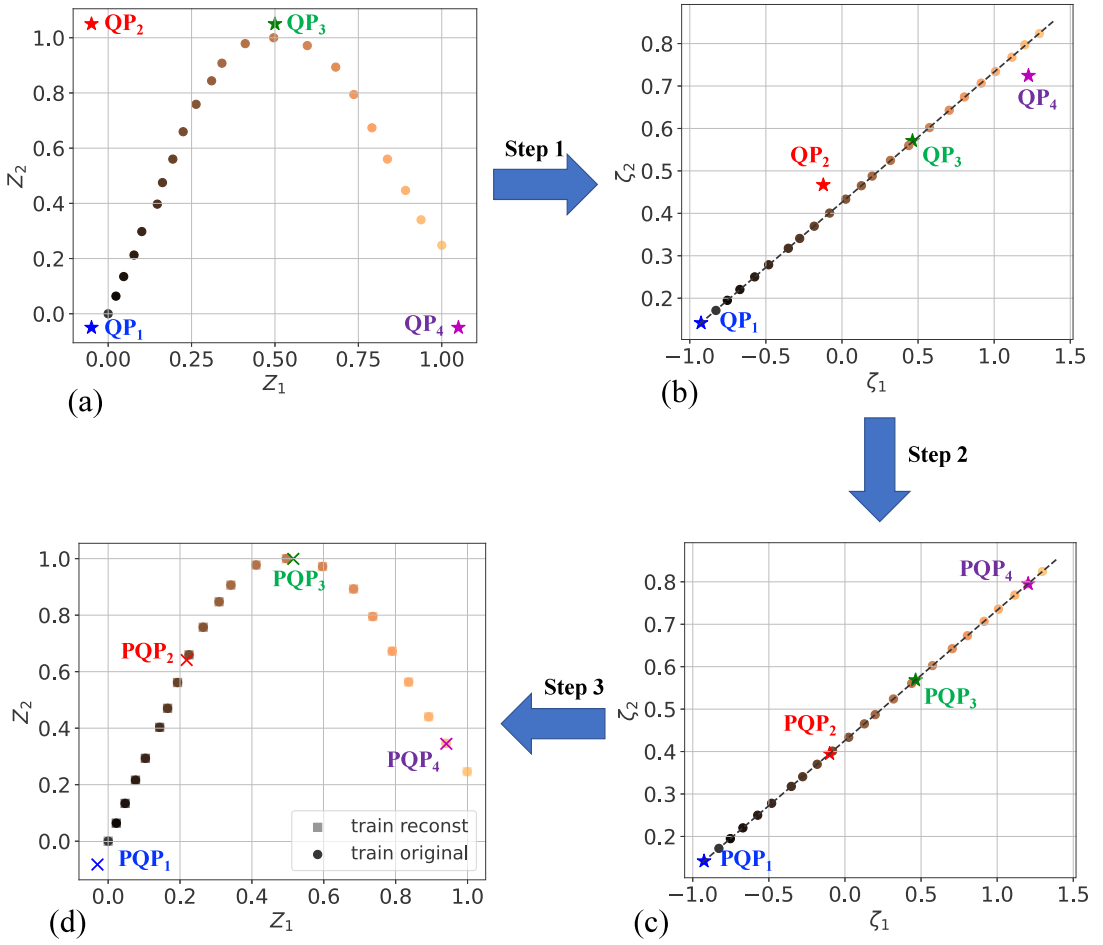
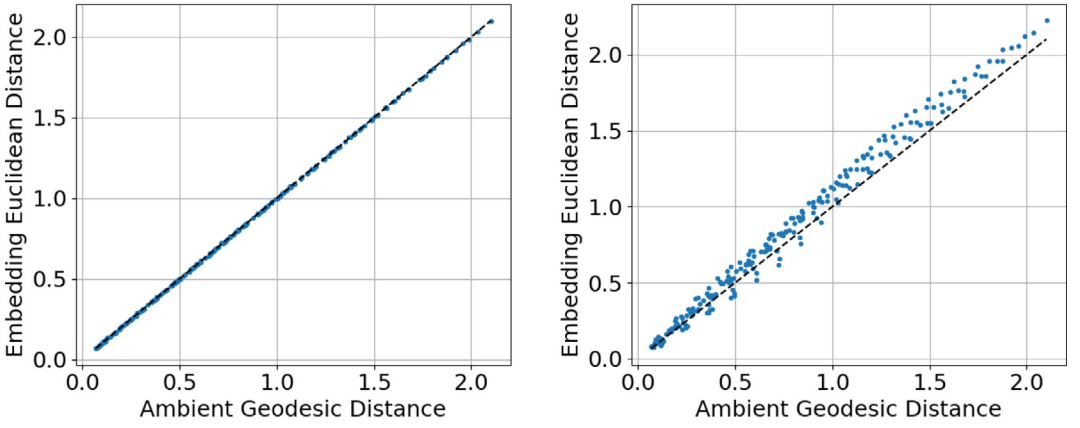


Fig. 4. Local projection steps for the *non-isometric* training. The steps are consistent with those in Fig. 1. These results are obtained after training the encoding and decoding functions.



(a) trained by isometric objective **(b)** trained without isometric objective

Fig. 5. A comparison between the *ambient* discrete geodesic distances of the randomly picked pairs of data points, the same pairs shown in Fig. 2(b), and the Euclidean distances of the same pairs in the constructed *latent* space by (a) an isometric and (b) non-isometric decoder functions.

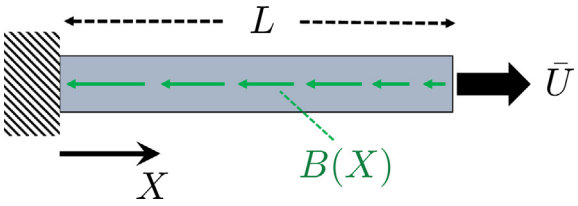


Fig. 6. The one-dimensional bar with length $L = 1$m, body force $B(X)$, and applied displacement $\bar{U} = 0.5$ m. We use 20 uniform finite elements with linear basis functions to discretize the domain.

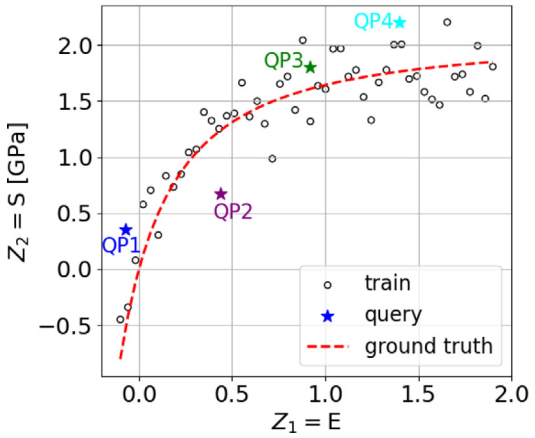


Fig. 7. A synthesized material constitutive behavior database. E and S are the Green–Lagrange strain and 2nd Piola–Kirchhoff stress, respectively.

`SymPy` package [55] using symbolic calculations. For brevity, the lengthy mathematical expression of the body force is not provided here but can be easily obtained.

We first synthesize a noiseless material database $\{(E_i, S_i)\}_{i=1}^N$ by sampling $N = 50$ data points with equidistant Green–Lagrange strain values in the range $E_i \in [-0.1, 1.9]$ from the Neo-Hookean model. The noisy database $\{(E_i, \hat{S}_i)\}_{i=1}^N$ is synthesized by adding a random distribution to the noiseless stress values $\hat{S}_i = S_i + \mathcal{N}(0, 2|S_i|)$ where $\mathcal{N}(\mu, \sigma)$ is the normal distribution with the mean and standard deviation μ and σ, respectively. The data points are shown in Fig. 7.

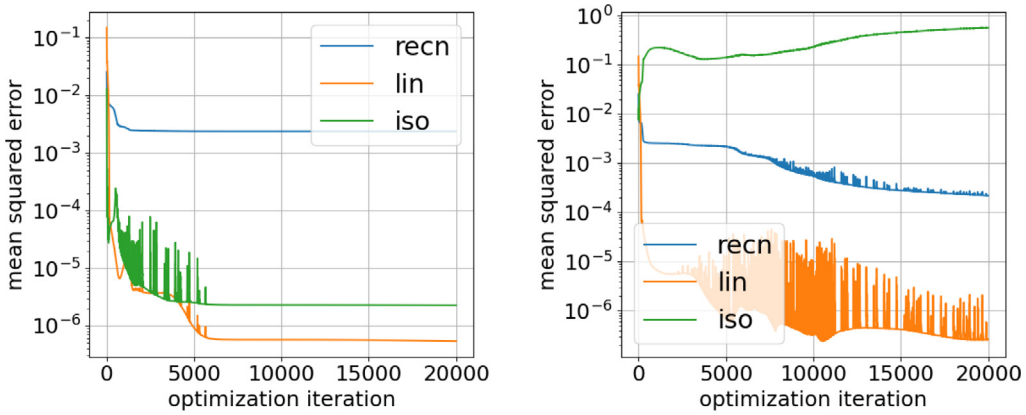
We first conduct a numerical experiment to test the necessity of the isometric constraint and how this constraint affects the property of the autoencoder. To achieve this goal, we train two autoencoders with the same neural network architectures but are given two different sets of loss functions; one trained with the isometric constraint in Eq. (12) enforced, and another one trained without. The neural network training performances are shown in Fig. 8.

4.2.1. Dimension reduction

In both cases, the linearity constraint (Eq. (14)) is enforced successfully with the corresponding MSE less than 10^{-6}. This indicates that both autoencoders are capable of reducing the intrinsic dimensions of the data by mapping the data from the ambient space to the flat latent space.

4.2.2. Isometry and flatness of latent space

While both autoencoders may enforce the flatness of the latent space, this flatness is achieved by different types of mappings that deform the constitutive manifold differently. In the case where the isometry is explicitly enforced (see Fig. 8(a)), the MSE losses of the three constraints in Eq. (12) reduce gradually and stay stable after about 7000 iterations. On the other hand, in the case that the loss of isometry is only monitored but not enforced (see Fig. 8(b)), we observe a gradient conflict issue between the isometric constraints Eq. (15) and the other two constraints that govern the reconstruction and linearity objectives. In particular, the MSE of the isometric constraints Eq. (15) shown



(a) ED 1: $w_{\text{recn}} = w_{\text{lin}} = w_{\text{iso}} = 1$ **(b)** ED 2: $w_{\text{recn}} = w_{\text{lin}} = 1$ and $w_{\text{iso}} = 0$

Fig. 8. Training curves for autoencoder trained for the noisy Neo-Hookean data with isometric constraints (LEFT) and without it (RIGHT). These three curves correspond to the losses on reconstruction Eq. (13) (blue), linearity Eq. (14) (orange), and isometry Eq. (15) (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

in Fig. 8(b) increases while the MSE of the reconstruction Eq. (13) and linear constraints Eq. (14) both decrease. This result indicates that, when the isometry constraint is turned off, the flatness of the latent space is achieved in a manner that does not preserve distance. Meanwhile, the proposed autoencoder is able to preserve the distance while flattening the constitutive manifold. The fact that the losses on the linearity and isometry both reduce during the training suggests that the distance-preserving property can be enforced without negatively impacting our goal to reduce the intrinsic dimension of the latent space.

4.2.3. Property of reconstruction mapping

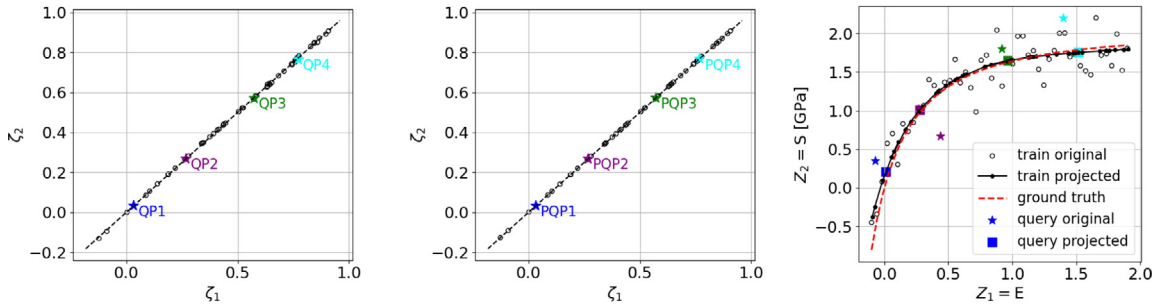
Note that the reconstruction error of the isometric autoencoder is about one order higher than the non-isometric counterpart, as shown in the blue curves in Fig. 8. This is expected, as the de-noising process is expected to make the de-noised signal different from the noisy data. Meanwhile, an interesting aspect revealed in this numerical experiment is *how* the auto-encode achieves the reconstruction from the flattened latent space back to the ambient space. By comparing the green and blue curves in Fig. 8(b), one may observe that the reduction of the reconstruction error is at the expense of further violating the isometry constraint.

The importance of the isometry property on the decoder is demonstrated in the two reconstructed constitutive manifolds shown in Fig. 9 (with isometry enforced) and Fig. 10 (without isometry). While both approaches use a projection in the latent space to suppress the noise, the decoder that enforces isometry is able to keep the noise filtered out from the signal in the reconstructed constitutive manifold. On the other hand, allowing the violation of the isometry observed in Fig. 8(b) may make it possible for the neural networks to learn back the noisy pattern such that the reconstruction loss can be further suppressed.

4.2.4. Sensitivity on neural network initialization

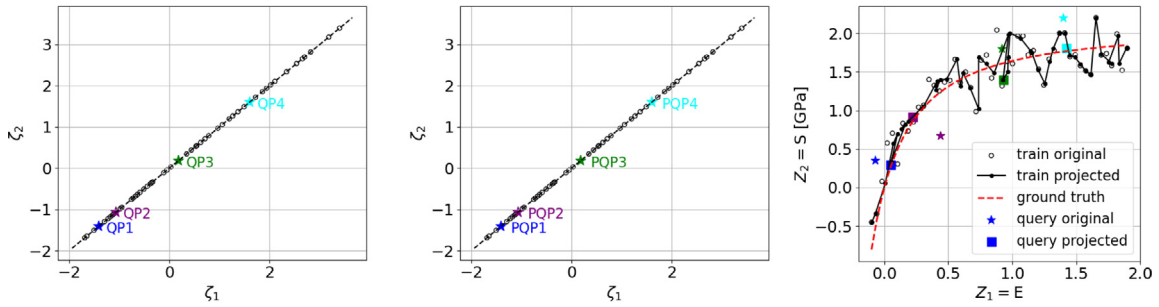
Note that the constructed constitutive manifold may exhibit dependence on the neural network initialization. Since neural network training usually leads to a highly non-convex optimization problem where a unique global optimum might not exist or be found [56]. To gather further evidence on the importance of the isometry and to assess the robustness of the de-noising procedure, we repeat the training process five times, each time with a different neural network initialization. The results of these five trials are shown in Fig. 11. This additional experiment suggests that the previously mentioned outcomes are independent of neural network initialization. Note that the performance of the autoencoder is also sensitive to the choices of hyperparameters. For brevity, we did not introduce additional studies on hyperparameter tuning.

Remark 4 (*Stress-free Configuration*)**.** The de-noised manifold might not always pass the origin exactly, e.g., see Fig. 9(c). This implies that the reference configuration may not be stress-free. Presumably, enforcing the reference



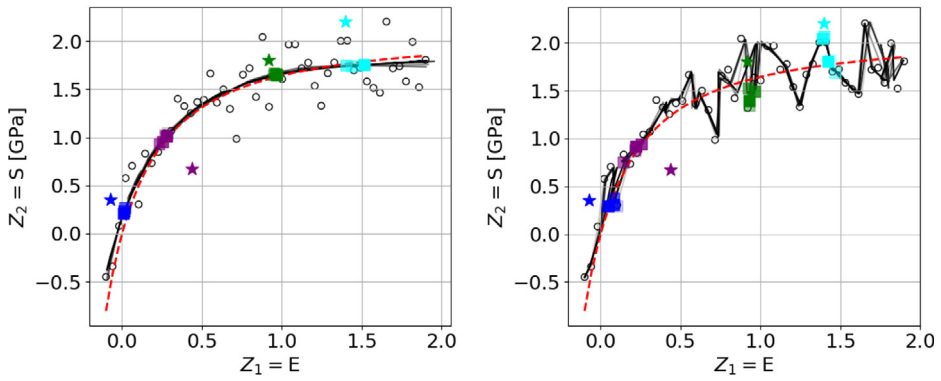
(a) Step 1: Encoder **(b)** Step 2: Projection **(c)** step 3: Reconstruction

Fig. 9. A comparison between the *Local* projection steps for the *isometric* training when the noisy Neo-Hookean data is used. The steps are consistent with those in Fig. 1.



(a) Step 1: Encoder **(b)** Step 2: Projection **(c)** Step 3: Reconstruction

Fig. 10. A comparison between the *Local* projection steps for the *non-isometric* training when the noisy Neo-Hookean data is used. The steps are consistent with those in Fig. 1.



(a) isometric training **(b)** non-isometric training

Fig. 11. A study of the neural network initialization effect on discovered manifolds. Different color intensities are used to indicate different random seeds (in total 5 random seeds). Solid lines pass through the projected training data (circles). Square points show the projected query points (stars). The dashed line indicates the ground truth. The results are shown after (a) isometric and (b) non-isometric neural network trainings. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

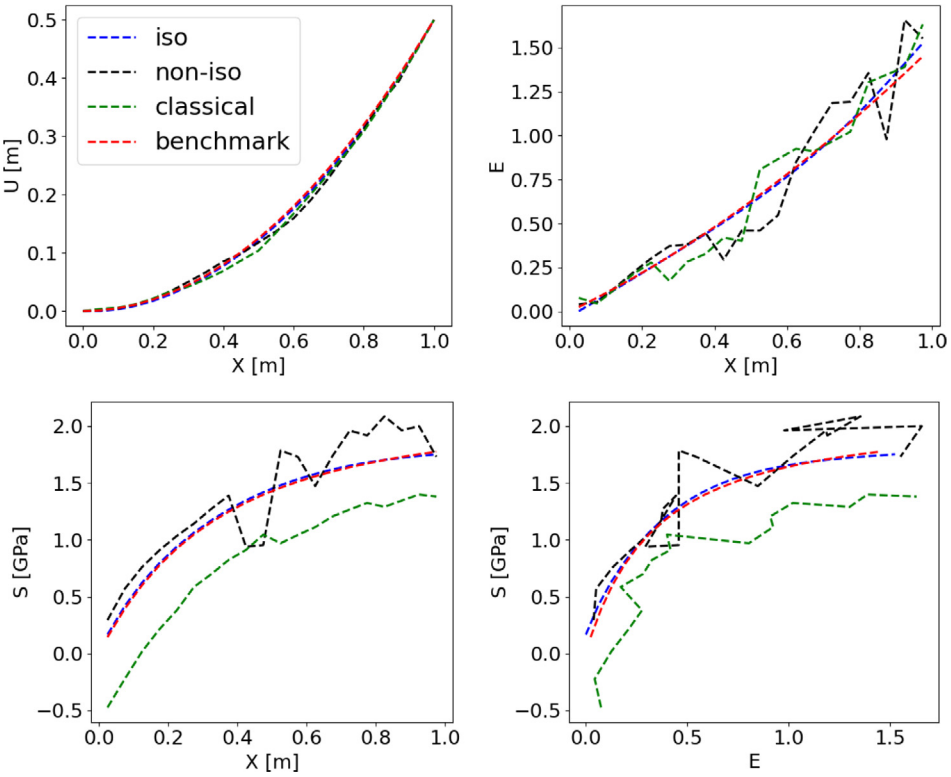


Fig. 12. A comparison of response fields obtained by different schemes and the exact manufactured solution for the 1D bar BVP. U denotes the displacement, E and S denote the strain and stress measures accordingly.

configuration to be stress-free is possible by introducing additional constraints in the loss function. Alternatively, one may design a neural network architecture that inherently maps the zero state exactly to itself, e.g., $f(x;\boldsymbol{\theta}) = x\mathcal{F}(x;\boldsymbol{\theta})$.

4.2.5. Boundary value problem solution via different data-driven schemes

A successful de-noising of the constitutive manifold is of practical value only if it can lead to the correct solutions of a boundary value problem. To this end, we compare three different local projection algorithms:

1. Autoencoder with the local isometry enforced (denoted as "iso")
2. Autoencoder without the local isometry enforced (denoted as "non-iso")
3. Classical nearest neighbor search (cf. Kirchdoerfer and Ortiz [2], denoted as "classical").

The parameter for the global optimization step $\mathcal{C} = 0.66$ GPa is set the same among all the schemes. Due to the non-linearity, we solve this BVP in three linearly increased loading steps for all the schemes. The optimal strain-stress values $(E^* - S^*)$ at all quadrature points at the initial step are set to zero. Twenty uniform elements discretize the bar domain. The classical linear basis functions are used for the unknown fields and their associated test functions.

The displacement, strain, and stress fields along the bar are plotted in Fig. 12. As the results suggest, there is a good agreement among all the methods for the displacement field prediction. However, data-driven paradigm with the isometric autoencoder is found to make more accurate strain and stress field predictions that are also free of spurious oscillations.

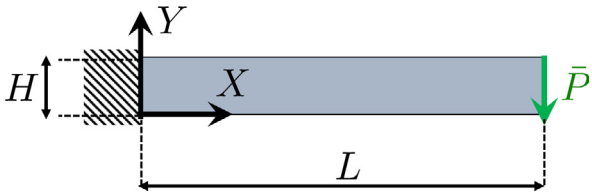


Fig. 13. The 2D beam problem with the length $L = 20$ mm and the height $H = 1$ mm. The total shear force is $\bar{P} = 10EI/L^2$ where the beam moment of inertia and Young's modulus are $I = H^3/12$ and $E = 4.8$ GPa, respectively.

4.3. Representative example 1: sensitivity analysis on cantilever beam simulations in the finite strain regime

The purpose of this example is to analyze how the performance of the de-noising data-driven paradigm is affected by the sampling size in a more general setting where the dimension of the boundary value problem is increased to 2D, and the geometrical nonlinearity of the deformation is considered. In this example, we solve the same 2D plane-strain cantilever beam problem under the bending loading condition previously used in [17]. Noisy data sets are synthesized to examine how the availability of the data and the amount of noise affect the accuracy and robustness of the de-noising data-driven paradigm. The geometry of the body and the boundary conditions are depicted in Fig. 13. The left end of the bend is fixed, while uniform shear traction is applied over the right side with the total force $\bar{P}$.

To synthesize the database, the Saint Venant–Kirchhoff phenomenological model with Young's modulus $E = 4.8$ GPa and Poisson's ratio $\nu = 0$ is used. The noiseless Green–Lagrange strain components are evenly sampled from the interval $\tilde{E}_{ij} \in [-0.02, 0.02]$ to generate databases with sizes 10^3, 20^3, and 30^3 data points. The noisy databases are generated by a procedure used previously in [11,17] where Gaussian noises proportional to the size of data sets are added to each component of strain and stress tensors. For each component (E_{ij} or S_{ij}), the added noise has zero mean, while the corresponding standard deviation is equal to the maximum value of that component among all noiseless data points scaled by the factor $0.4/\sqrt[3]{|\mathcal{D}|}$. $|\mathcal{D}| \in \{10^3, 20^3, 30^3\}$ is the dataset size. As such, the dataset with fewer data points also fluctuates in the parametric space more.

4.3.1. Sensitivity of the autoencoders with respect to data and noise

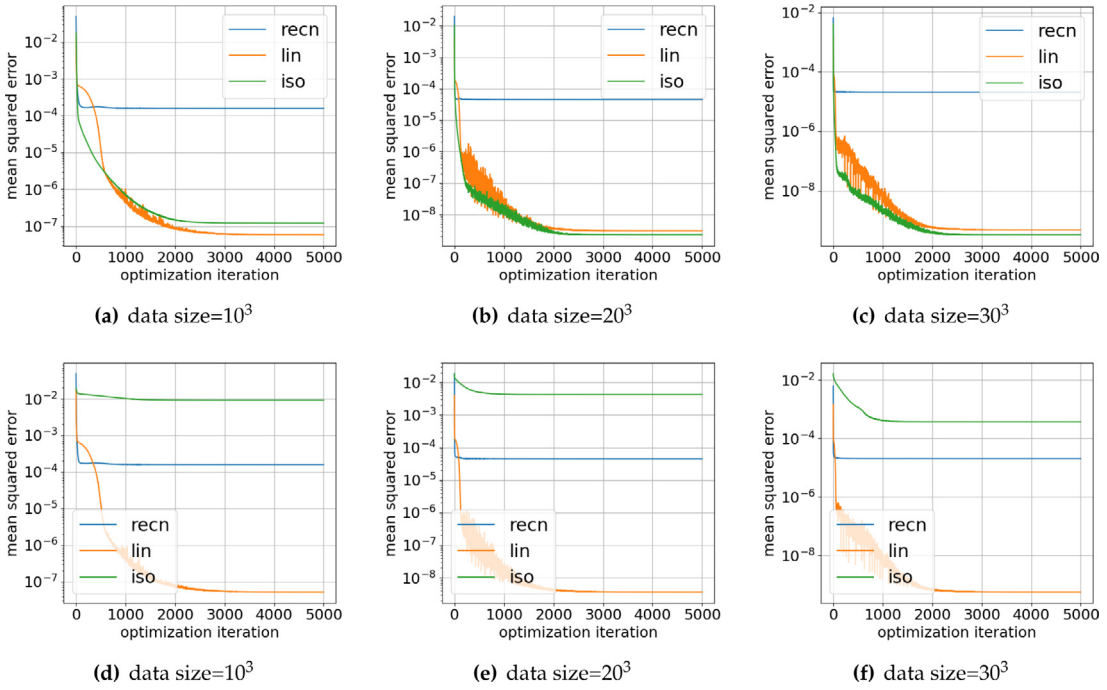
To assess the sensitivity of the de-noising data-driven algorithm with respect to the amount of data, we train two autoencoders (with and without isometry constraints) with three noisy data sets described earlier.

Fig. 14 reports the training performance of these two sets of autoencoders trained with different amounts of noise data. All MLP functions are initialized with the same parameters at the beginning of the optimization iterations. These training problems differ based on the amount of data utilized in their training process and their isometric constraint weight $w_{\text{iso}} = \{0, 1\}$. Both isometric and non-isometric autoencoders exhibit better performance when given more training data. Interestingly, even without explicitly enforcing the isometric constraints (see Fig. 14(d)–(e)), adding more data also leads to a minor improvement in fulfilling the isometry constraints. Furthermore, the overall trends of the MSE for all three constraints are also monotonic decreasing. This suggests that, unlike the previous 1D example, there is no noticeable gradient conflict observed during the training. Given the additional facts that both the isometric and non-isometric cases lead to very similar MSE for the linearity and reconstruction constraints, it is plausible that the learned constitutive manifolds and the latent spaces obtained via different strategies and data sets of different sizes are all very similar. In this particular case, the isometric constraint merely ensures the distance-preserving property but does not exhibit a major influence on the MSE of reconstruction.

4.3.2. Sensitivity of the data-driven simulations with respect to data and noise

To examine further the performance of the autoencoders trained with different amounts of noisy data, we simulate the same pure bending BVP with each of these autoencoders and compare the corresponding results.

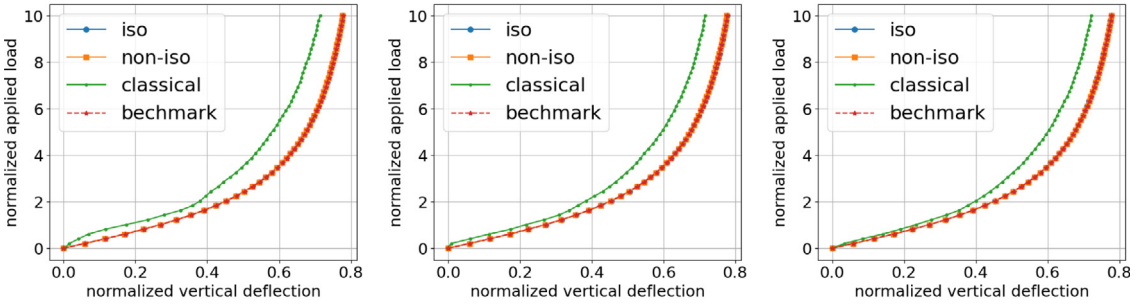
In the global step of the data-driven paradigm, the tensor used to constitute the energy norm Eq. (8) is set to be equal to the elasticity tensor of a linear elastic material with $\nu = 0$ and $E = 4.8$ GPa. Meanwhile, the local step is solved via projection in the latent space. For comparison purposes, three additional simulations obtained via the classical local optimization step used in [2] are included. Meanwhile, The benchmark solution is the solution



(a) data size=10^3 **(b)** data size=20^3 **(c)** data size=30^3

(d) data size=10^3 **(e)** data size=20^3 **(f)** data size=30^3

Fig. 14. Training performance of autoencoder trained with (top row) and without (bottom row) the isometry constraints when given noisy data of different sampling densities.



(a) data size=10^3 **(b)** data size=20^3 **(c)** data size=30^3

Fig. 15. A comparison of tip deflection profiles during the loading steps with different schemes. The horizontal axis represents U_Y^{tip}/L for the bottom tip on the loading side of the beam where U_Y^{tip} is the vertical displacement. The vertical axis represents the normalized applied load PL^2/EI during 50 linearly increased loading steps where the beam moment of inertia is $I = H^3/12$.

obtained by the conventional displacement-based finite element solver with constitutive responses obtained from the noiseless ground-truth constitutive model.

The bottom tip vertical displacement of the beam is recorded during the load steps in Fig. 15. There is a good agreement between the isometric and non-isometric cases compared with the benchmark. The classical distance-minimization paradigm also captures the overall trend and improves as the data size grows, but the results are less accurate.

Since hyperparameters and the initial weights of the neural network may both affect the performance, we repeated the numerical experiments multiple times, each with a neural network initialized by different initial weights but of the same architecture (the same hyperparameters listed in Appendix D.2).

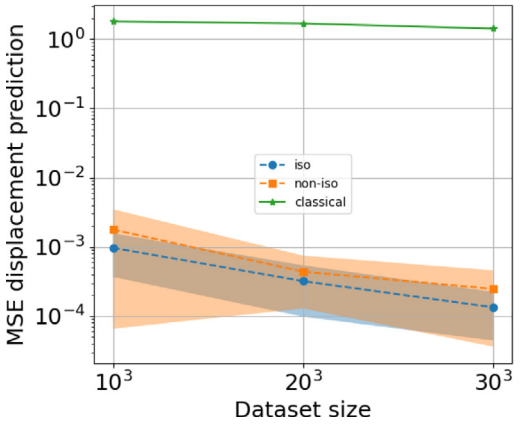


Fig. 16. Nodal displacement errors in the computational domain for different methods when different data sizes are used. In cases of isometric and non-isometric embedding functions, four different neural networks with different initialized random weights are trained with the same configuration. The shaded area shows mean $\pm$ 0.85 std.

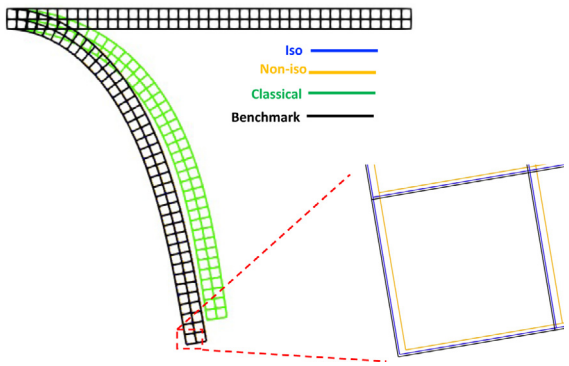


Fig. 17. The deformed computational domain obtained by different schemes for the bending beam problem.

The mean square error (Eq. (22)) between all the nodal displacement values in the finite element mesh obtained by the data-driven methods and the benchmark are plotted in Fig. 16.

$$\mathrm{MSE}_{\mathrm{disp}} = \frac{1}{N_{\mathrm{node}}} \sum_{i=1}^{N_{\mathrm{node}}} \|\boldsymbol{U}^i_{\mathrm{gt}} - \boldsymbol{U}^i_{\mathrm{dd}}\|_2^2, \tag{22}$$

where the superscript i indicts the nodal index and subscripts gt and dd refer to the ground truth and data-driven solutions, respectively. As the results suggest, the isometric training performs slightly better than the non-isometric case in terms of smaller mean errors and variance. For better visualization, the displacement solution is plotted in Fig. 17. These solutions are obtained by one of the found mapping functions (one random seed) with the same initial weights for both isometric and non-isometric cases.

In summary, the solutions obtained from the two autoencoders and the classical data-driven paradigm exhibit comparable performance, with the isometric approach producing a slightly more accurate deformed configuration. Given the similar reconstruction errors of these six autoencoders previously discussed in Section 4.3.1, these comparable performances are expected. On the other hand, we have also noticed that the performance of the two manifold embedding approaches is both not very sensitive to the amount of data and the amplitude of noises added to the observed data. This set of results suggests that both manifold embedding approaches may deliver robust predictions even in the presence of noisy data, provided that the embedding is successful (as indicated by the loss function Eq. (12)).

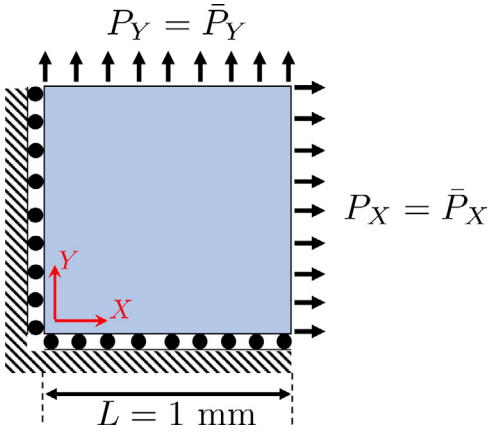


Fig. 18. The geometry and boundary conditions of the plane-stress biaxial test. Normal roller constraints are used on the left and bottom edges. The uniform traction is applied over the right and top edges.

4.4. Representative example 2: biaxial test: validation against experimental data

The purpose of this last example is to test and demonstrate the performance of the de-noising data-driven paradigm when given realistic experimental data for real-world applications. In this problem, we solve the plane-stress biaxial loading for an anisotropic material shown in Fig. 18. This is a common experiment in biomedical applications [57] and has also been used to test the robustness of a data-driven paradigm [17]. In this set of numerical examples, we use both synthesized and real experimental data. In the first set of numerical experiments, we use synthesized data via Fung's model and introduce artificial noises to compare the data-driven solutions with the available benchmark. In the second numerical experiment, we then utilize real experimental data to simulate a soft tissue specimen undergoing finite strain deformation to examine the proposed de-noising data-driven paradigm.

4.4.1. Simulations with synthetic noisy data

Datasets are generated from one of Fung's phenomenological constitutive models [30,58] with the following energy functional:

$$\psi = \frac{c}{2}\left[\exp\left(\bar{\psi}\right) - 1\right], \tag{23}$$

$$\bar{\psi} = \begin{bmatrix} E_{XX} & E_{YY} \end{bmatrix} \begin{bmatrix} a_1 & a_3 \\ a_3 & a_2 \end{bmatrix} \begin{bmatrix} E_{XX} \\ E_{YY} \end{bmatrix} \tag{24}$$

where $c = 960$ Pa, $a_1 = 48.4$, $a_2 = 46.3$ and $a_3 = 15.1$ are material properties chosen from [59] for the Ileum biological tissue with the anisotropy index 0.97. The strain-stress relations can be straightforwardly derived as follows:

$$\hat{S}_{XX} = c\exp(\bar{\psi})\left(a_1 E_{XX} + a_3 E_{YY}\right), \tag{25}$$

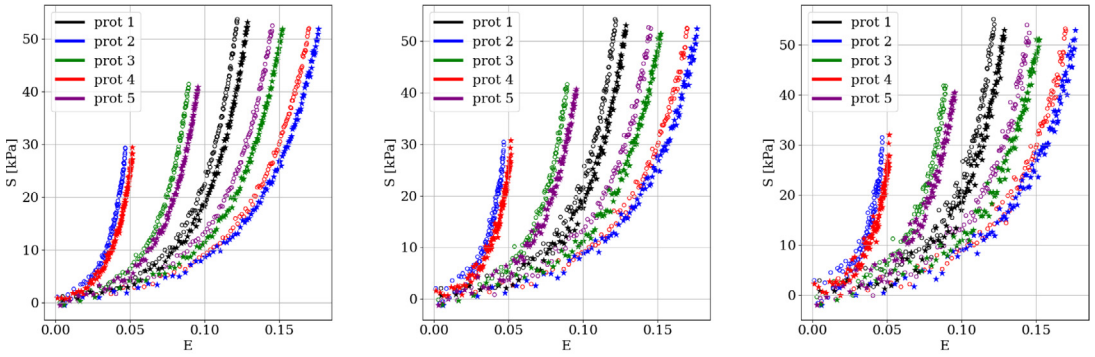
$$\hat{S}_{YY} = c\exp(\bar{\psi})\left(a_3 E_{XX} + a_2 E_{YY}\right), \tag{26}$$

and the shear component vanishes; the noiseless stress components are decorated by a hat $\hat{S}$.

The noiseless data (benchmark) is obtained by a displacement-based finite element solver with Fung's phenomenological energy functional. Different force-driven biaxial simulations are conducted based on the different ratios of external loads as $\bar{P}_X/\bar{P}_Y \in \{1, 0.5, 0.75, 2, 1.3\bar{3}\}$ labeled by protocols 1-5, respectively. These simulations start from zero loading, and the loading is linearly increased in 100 steps up to $\bar{P}_X = 60$ kPa for protocols 1-3 and $\bar{P}_Y = 60$ kPa for protocols 4-5. We add Gaussian noise to the noiseless data as follows:

$$S_{IJ} = \hat{S}_{IJ} + \mathcal{N}(0, \sigma), \tag{27}$$

where $\sigma \in \{0.5, 1, 1.5\}$ kPa is the assumed standard deviation. The three generated datasets based on the level of the injected noise are plotted in Fig. 19. Note that we assume the shear components are all zero in the datasets



(a) std=0.5 **(b)** std=1 **(c)** std=1.5

Fig. 19. Synthesized noisy datasets using Fung's model. Each protocol shows a different boundary condition in the biaxial test. The data is represented as $E_{XX} - S_{XX}$ and $E_{YY} - S_{YY}$ curves; see [39] for more details on this type of representation. Data points with circular and star markers correspond to $E_{XX} - S_{XX}$ and $E_{YY} - S_{YY}$ curves, respectively, and their colors indicate their protocol types. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and training tasks due to the plane stress condition. In the following results, we test a protocol (labeled as 6) with a vertical-to-horizontal loading ratio of 1.5 as a test case where such a loading condition is not included in the training data. Notice that even for test cases (protocols 1 and 4) with loading conditions similar to the train data, the sampling points in the train data (50 data points) are different from the test scenarios (80 loading steps).

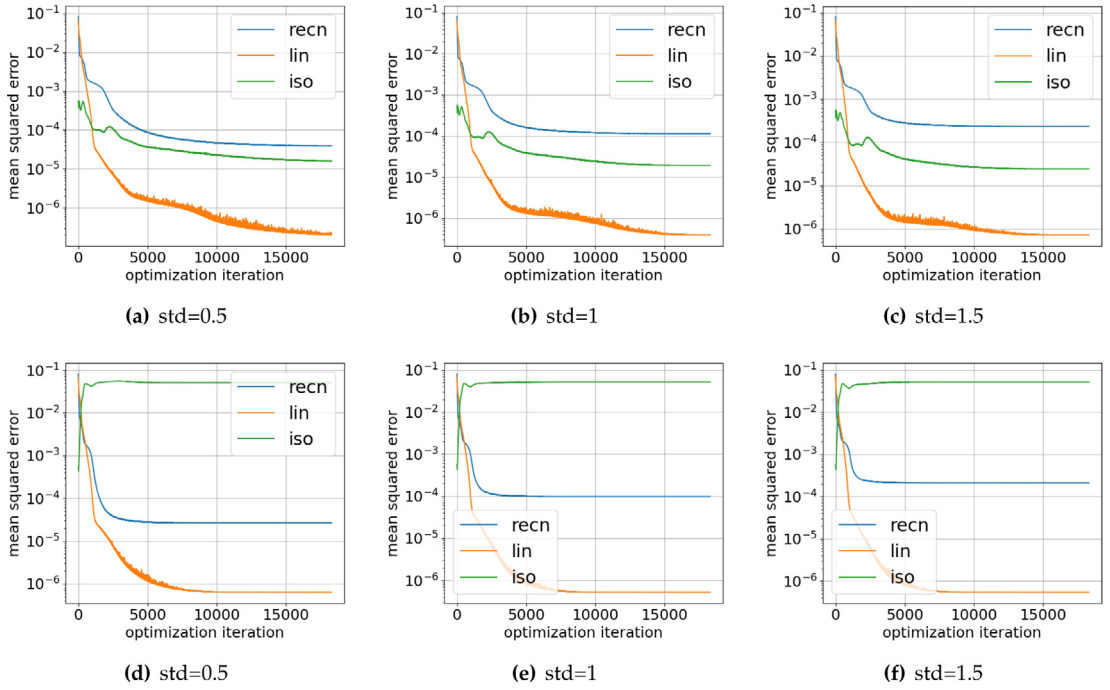
The training curves for isometric and non-isometric training cases are plotted in Fig. 20. As the data becomes noisier, the reconstruction error increases in both cases. Also, the reconstruction error reaches a plateau, meaning that the network set-up is not too expressive to over-fit the noise too much. Both of these observations are necessary for a good training process. The linearity constraint remains almost the same, indicating that the neural network is sufficiently expressive for the hyperplane construction regardless of the noise level.

The data-driven simulations are conducted in the force-based control with 80 equal loading steps. The global optimization parameter $\mathbb{C}$ has only non-zero diagonal terms as follows $[\mathbb{C}_{IJ}] = \sigma(S_{IJ})/\sigma(E_{IJ})$ where $\sigma(\cdot)$ is the standard deviation operator. For a consistent comparison among all datasets, we fix this parameter as $[\mathbb{C}_{XX}, \mathbb{C}_{YY}] = [332.3, 321.7]$ kPa which is calculated based on the noiseless data.

The obtained Green–Lagrange strain versus the second Piola–Kirchhoff stress is shown in Fig. 21. The benchmark solution refers to the solution obtained by the common displacement-based finite element solver where the phenomenological constitutive law is utilized.

We observe that increasing the standard deviation of the noise (see Fig. 19(a)–(c)) may still affect the accuracy and smoothness of the constitutive laws. However, there exists a good agreement between the recovered noiseless response. In this numerical example, the isometric and non-isometric methods yield similar reconstructed stress-strain curves. This is consistent with the training results shown in 20(a) and (b) where the 5 orders of difference in the isometry loss impose a marginal effect (less than an order) on the reconstruction and linearity losses. As such, the isometry constraint here merely plays the role of insurance but does not impose a significant difference on the mappings enabled by the trained neural network.

Remark 5 (*Alternative De-noising Data-driven Paradigms*)**.** In this paper, we re-implemented the data-driven paradigm in [2] and used it to generate benchmark results. It should be noted that there are other options available that may handle noisy data in the data-driven paradigm, as we indicated in the introduction. For example, He et al. [17] use a partition of unity with convexity for local data reconstruction on the latent space for de-noising and stability purposes. For brevity, we do not re-implement these algorithms and will not compare all the feasible options. Readers may refer to Kirchdoerfer and Ortiz [11], He and Chen [12], Eggersmann et al. [60] and He et al. [17] for these alternatives.



(a) std=0.5 **(b)** std=1 **(c)** std=1.5

(d) std=0.5 **(e)** std=1 **(f)** std=1.5

Fig. 20. Training performance of the isometric (top row) and non-isometric (bottom row) autoencoders trained from data sets with noises of different amplitudes. The ground truth data is generated from Fung's model.

4.4.2. Simulations with experiment data

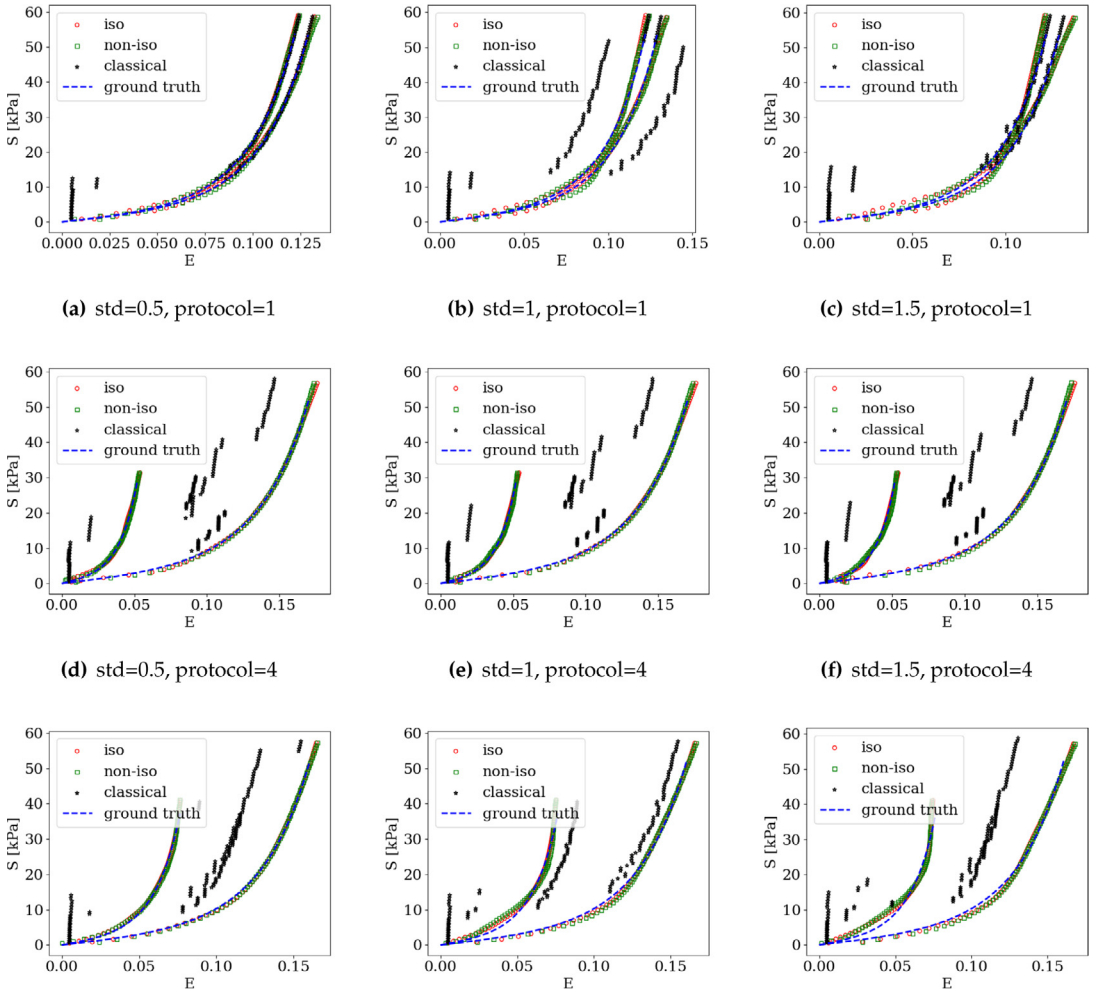
We use the reported experimental data from [38,39]. Particularly, the dataset with the label `MVAL1.txt` is utilized for the mapping functions training and the data-driven solver. This data is obtained by the biaxial test on the mitral valve anterior leaflet; refer to [38] for more details. Since the raw data is reported based on the principal stretches and first Piola–Kirchhoff stress values, we transform the data into Green–Lagrange strains and second Piola–Kirchhoff stresses as follows:

$$[E_{XX}, E_{YY}] = [0.5\lambda_X^2 - 1, 0.5\lambda_Y^2 - 1],$$
$$[S_{XX}, S_{YY}] = [P_{xX}/\lambda_X, P_{yY}/\lambda_Y],$$

where λ_X and λ_Y are stretch ratios in the horizontal and vertical directions. The dataset used in this study is plotted in Fig. 22. The isometric and nonisometric training plots are provided in Fig. 23. The same observation as described for Fung's data training holds here.

The solver parameters are kept the same as in the previous case study. Since the ground truth response is unavailable, we only compare the proposed schemes with the classical distance-minimization method. Also, the experimental data for protocol loading 6 is not available The obtained results for different biaxial loading conditions are shown in Fig. 24. Both de-noising schemes lead to similar stress-strain responses at the latter stage of the experiments. The classical nearest neighbor approach may lead to erroneous constitutive responses in the limited data regime.

In between the de-noising schemes, the discrepancy is more profound at the initial loading steps. One hypothesis that may explain this phenomenon is that the noise of the acquired data is probably of similar amplitude during the experiments. As such, when the deformation is small, the effect of the noise is more profound. Hence, the isometry condition plays a more important role in reconstructing the data in the earlier phase of the experiment. On the contrary, the de-noising becomes less crucial when the magnitude of the stress increases such that the ratio between noise and true signal gradually reduces. Sparsity, noise, and non-linearity at the beginning of loading steps may result in spurious step-wise constitutive behavior when the classical distance-minimization algorithm is used.



(a) std=0.5, protocol=1 **(b)** std=1, protocol=1 **(c)** std=1.5, protocol=1

(d) std=0.5, protocol=4 **(e)** std=1, protocol=4 **(f)** std=1.5, protocol=4

(g) std=0.5, protocol=6 **(h)** std=1, protocol=6 **(i)** std=1.5, protocol=6

Fig. 21. A comparison among different data-driven schemes and the benchmark solution obtained by the displacement-based finite element solver with the phenomenological constitutive law. The data-driven methods use noisy Fung's type data with different amounts of artificially injected Gaussian noise. Each protocol shows a different biaxial boundary condition, and *std* indicates the standard deviation.

5. Conclusions

We present a manifold embedding data-driven paradigm where a modified autoencoder is designed to handle noisy manifold data while preserving the underlying geometry of the data. The encoding of the noisy data is split into two steps, where the nonlinear dimensional reduction is achieved through two steps. First, a mapping function is used to bend the constitutive manifold to an approximated hyperplane. Then, by assuming orthogonality between the noise and true signal in the latent space, a projection from the approximated hyperplane onto the actual hyperplane may enable nonlinear filtering of high-frequency noise on the manifold. Finally, we introduce an additional local isometry constraint in the loss function that trains the neural network responsible for reconstructing data from the latent space back to the phase space. This constraint enables the learned neural network decoder to reconstruct the data while preserving the distance of two points in the latent space and those mapped onto the reconstructed constitutive manifold upon the decoding. This distance-preserving autoencoder is shown to be able to help preventing the noise from being incorporated back into the nonlinear decoding. Three sets of numerical experiments are used to examine the performance of the de-noising manifold embedding data-driven algorithm. Boundary value problems

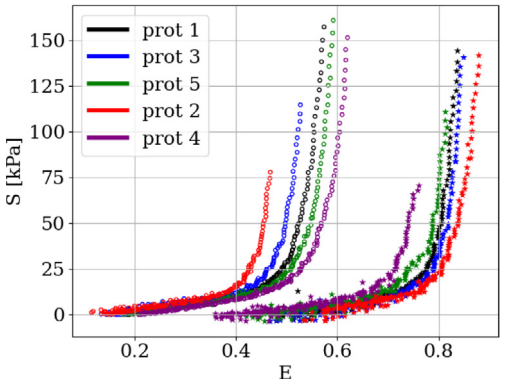


Fig. 22. The experimental dataset from a biaxial test on a specimen of mitral valve anterior leaflet [38]. Data points with circular and star markers correspond to $E_{XX} - S_{XX}$ and $E_{YY} - S_{YY}$ curves, respectively, and their colors indicate their protocol types. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

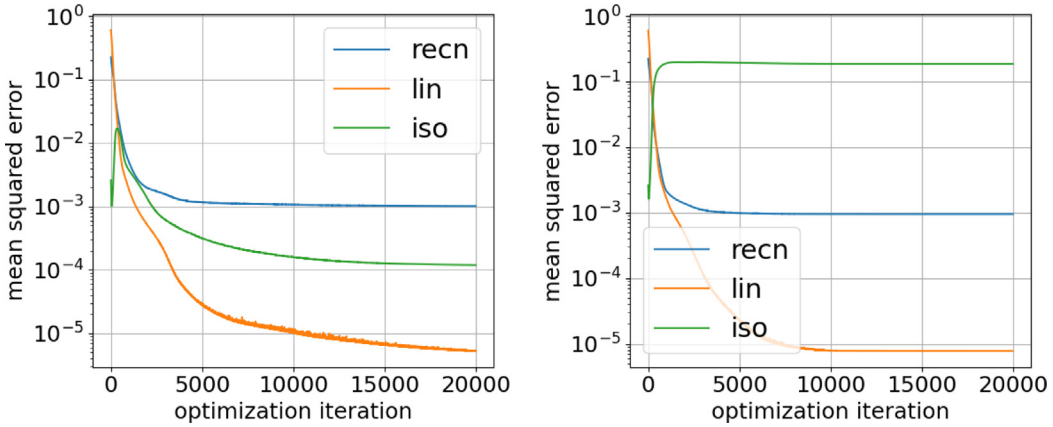
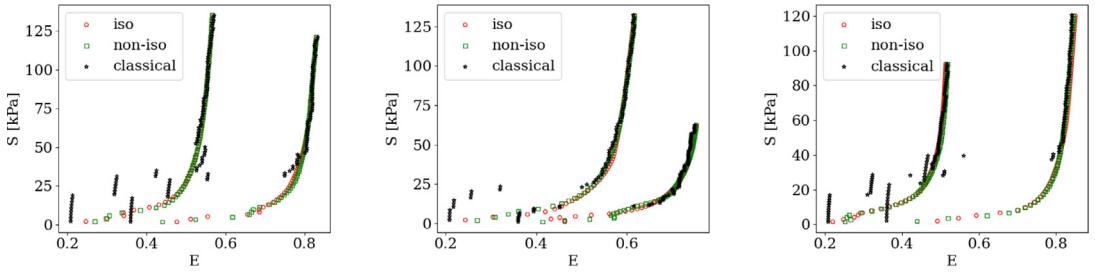


Fig. 23. Training performance plots on the real dataset for the isometric (left) and non-isometric (right) mapping functions.



(a) protocol 1 **(b)** protocol 4 **(c)** protocol 6

Fig. 24. A comparison among different data-driven schemes for different biaxial boundary conditions (protocols).

conducted with both synthetic and real experimental data are conducted. The numerical results suggest that the proposed model is capable of delivering accurate and robust responses when given noisy data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The two anonymous reviewers have provided many insightful feedback and suggestions that lead to improvements of this paper. Their effort and time are gratefully acknowledged. The authors are supported by the National Science Foundation, United States under grant contracts CMMI-1846875 and OAC-1940203, and the Dynamic Materials and Interactions Program from the Air Force Office of Scientific Research, United States under grant contracts FA9550-21-1-0391 and FA9550-21-1-0027. These supports are gratefully acknowledged. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors, including the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors, including the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Appendix A. Mathematical preliminary for isometric autoencoder for constitutive manifold

Here we review the basic concepts of the manifolds relevant to the construction of the isometric autoencoder. To make the content self-contained for the target readers, a minimal review of the terminologies, definitions, and lemmas essential to establish the local optimization step on the de-noised manifold is provided herein. To simplify the formulation, we only consider n-dimensional constitutive manifolds that are connected, Hausdorff, and second countable with the tangential spaces that are locally Euclidean. For a more comprehensive yet concise review on this topic, please refer to, for instance, Marsden and Hughes [61] and Fong and Tino [62].

Let $\mathcal{N}$ be a Euclidean manifold equipped with the Euclidean metric that is the square root of the inner product. Let $\mathcal{G} : \mathcal{N} \to \mathcal{M}$ be a smooth mapping that maps the Euclidean manifold to a smooth Riemannian manifold $\mathcal{M}$ with the corresponding tangent space at $\mathcal{G}(\boldsymbol{\zeta}) \in \mathcal{M}$ denoted by $T_{\mathcal{G}(\boldsymbol{\zeta})}\mathcal{M}$ where $\boldsymbol{\zeta} \in \mathcal{N}$. Then, the corresponding associated map between the tangential spaces of $\mathcal{N}$ and $\mathcal{M}$ denoted as $T_{\boldsymbol{\zeta}}\mathcal{N}$ and $T_{\mathcal{G}(\boldsymbol{\zeta})}\mathcal{M}$ may read $(D\mathcal{G})_{\boldsymbol{\zeta}} : T_{\boldsymbol{\zeta}}\mathcal{N} \to T_{\mathcal{G}(\boldsymbol{\zeta})}\mathcal{M}$ (cf. Definition 1). Due to the equipped Euclidean metric, the Euclidean manifold is a special case of an n-dimensional "flat" space that admits a global coordinate system. On the other hand, an n-dimensional topological manifold can be represented locally by a local coordinate system corresponding to a tangent space for a given point that provides the local linear approximation of the manifold via the coordinate charts.

Definition 1 (*n-Dimensional Topological Manifold*)**.** A n-dimensional topological manifold $\mathcal{M}$ satisfies the following conditions (cf. Fong and Tino [62]).

1. Hausdorff condition, which requires that $\forall \boldsymbol{Z}_a, \boldsymbol{Z}_b \in \mathcal{M}, \exists U, V \in T_z\mathcal{M}$ such that $\boldsymbol{Z}_a \in U, \boldsymbol{Z}_b \in V$ and $U \cap V = \emptyset$. In other words, any distant points in this manifold is neighborhood-separable.
2. Second countable, i.e., the tangential space $T_z\mathcal{M}$ is spanned by a finite-dimensional basis.
3. Locally Euclidean of dimension n, every points in the manifold $\forall \boldsymbol{Z} \in \mathcal{M}$ has a neighborhood homeomorphic to either an n-dimensional Euclidean space $\mathbb{R}^n$ or an open ball in $\mathbb{R}^n$.

To construct a pair of distance-preserving mappings between $\mathcal{N}$ and $\mathcal{M}$, we need to first properly define a distance that is consistent with the geometrical properties of the manifold. This step requires additional structures to be equipped such that local information can be stitched together to form a global counterpart applicable to the entire manifold. Recall that our strategy here is to perform the de-noising task from the Euclidean manifold, then map back the de-noised manifold via an inverse map. The drawback of this strategy is that the mappings generated from the neural network do not necessarily preserve geometrical properties such as distance and convexity. Nevertheless, there are geometrical properties, such as angles and curvature that can be studied and controlled locally through the tangent spaces $T_z\mathcal{M}$. This can be done by considering the pushforward of the local coordinate map (see, for instance, [62]).

With the basis of the tangent space defined, we may consider the necessary condition to make the map $\mathcal{G}$ being local isometry (see Definition 2) where the transformation between two tangent spaces $T_\zeta\mathcal{N}$ and $T_{\mathcal{G}(\zeta)}\mathcal{M}$ is distance preserving locally.

Definition 2 (*Local Isometry*)**.** A map between two Riemannian manifolds $\mathcal{N}$ and $\mathcal{M}$ is local isometry if the associate map $(D\mathcal{G})_\zeta$ is a local diffeomorphism such that for all point $\boldsymbol{\zeta} \in \mathcal{N}$ and all vectors $d\boldsymbol{\zeta}_a, d\boldsymbol{\zeta}_b \in T_\zeta\mathcal{N}$,

$$\langle d\boldsymbol{\zeta}_a, d\boldsymbol{\zeta}_b \rangle = \langle (D\mathcal{G})_\zeta(d\boldsymbol{\zeta}_a), (D\mathcal{G})_\zeta(d\boldsymbol{\zeta}_b) \rangle, \tag{28}$$

where $\langle \cdot, \cdot \rangle$ is the inner product on $T_\zeta\mathcal{N}$ and $(D\mathcal{G})_\zeta$ is the associate map $(D\mathcal{G})_\zeta : T_\zeta\mathcal{N} \to T_{\mathcal{G}(\zeta)}\mathcal{M}$.

Here, the diffeomorphism of $\mathcal{G}$ requires it to be invertible, and both $\mathcal{G}$ and the inverse of $\mathcal{G}$ remain differentiable. Finally, the Riemannian isometry of $\mathcal{G}$ between $\mathcal{N}$ and $\mathcal{M}$ can be achieved by enforcing that both local isometry and the diffeomorphism hold such that the geodesic distance between any pair of arbitrary points in $\mathcal{N}$ is preserved upon the transformation (see, for instance, [63]).

Appendix B. Local isometry condition

Here, we prove that the orthogonality constraint on the Jacobian operator is equivalent to the local isometry. Let $\boldsymbol{\zeta}$ be a vector admissible to the latent (Euclidean) space and $\boldsymbol{\delta\zeta}$ be an infinitesimal perturbation of length $\|\boldsymbol{\delta\zeta}\|$ in the latent space such that the following linearized approximation remains valid,

$$\mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta} + \boldsymbol{\delta\zeta}) \approx \mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta}) + \boldsymbol{J}(\boldsymbol{\zeta}; \boldsymbol{\omega}) \cdot \boldsymbol{\delta\zeta}. \tag{29}$$

Then, by enforcing $\boldsymbol{J}(\boldsymbol{\zeta}; \boldsymbol{\omega})^T = \boldsymbol{J}(\boldsymbol{\zeta}; \boldsymbol{\omega})^{-1}$, and by assuming Eq. (29) is valid,

$$\begin{aligned}
\|\mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta} + \boldsymbol{\delta\zeta}) - \mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta})\|^2 &= \Big(\mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta} + \boldsymbol{\delta\zeta}) - \mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta})\Big) \cdot \Big(\mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta} + \boldsymbol{\delta\zeta}) - \mathcal{G}_{\boldsymbol{\omega}}(\boldsymbol{\zeta})\Big) \\
&= \Big(\boldsymbol{J}(\boldsymbol{\zeta}; \boldsymbol{\omega}) \cdot \boldsymbol{\delta\zeta}\Big)^T \Big(\boldsymbol{J}(\boldsymbol{\zeta}; \boldsymbol{\omega}) \cdot \boldsymbol{\delta\zeta}\Big) \\
&= \boldsymbol{\delta\zeta}^T \cdot \boldsymbol{J}(\boldsymbol{\zeta}; \boldsymbol{\omega})^T \boldsymbol{J}(\boldsymbol{\zeta}; \boldsymbol{\omega}) \cdot \boldsymbol{\delta\zeta} \\
&= \boldsymbol{\delta\zeta}^T \cdot \boldsymbol{\delta\zeta} \\
&= \|\boldsymbol{\delta\zeta}\|^2
\end{aligned}$$

Appendix C. Finite element discretization for the global projection step

Here, we provide the ingredients to obtain the finite element discretized equations which are represented entirely in tensorial notation. The Einstein summation convention holds unless otherwise specified.

In this work, the displacement and Lagrange multiplier fields are discretized by the same finite element basis functions as follows,

$$U_I(\boldsymbol{X}) \approx \sum_{\alpha=1}^{N_{\text{node}}} N_\alpha(\boldsymbol{X}) \hat{U}_{I\alpha} = N_\alpha(\boldsymbol{X}) \hat{U}_{I\alpha}, \quad 1 \le I, J \le N_{\text{dim}} \tag{30}$$

$$Q_I(\boldsymbol{X}) \approx \sum_{\alpha=1}^{N_{\text{node}}} N_\alpha(\boldsymbol{X}) \hat{Q}_{I\alpha} = N_\alpha(\boldsymbol{X}) \hat{Q}_{I\alpha}, \quad 1 \le I, J \le N_{\text{dim}} \tag{31}$$

where $N_\alpha(\boldsymbol{X})$ is the standard Lagrange polynomial basis function corresponding to the node α, $\hat{U}_{I\alpha}$ is the value of Ith $\in \{1, 2, 3\}$ component of the displacement vector field at the node α, and $\hat{Q}_{I\alpha}$ is the value of Ith component of the Lagrange multiplier vector field at the node α. Accordingly, the variations can be discretized as:

$$\delta U_I(\boldsymbol{X}) \approx N_\alpha(\boldsymbol{X}) \delta\hat{U}_{I\alpha} \tag{32}$$

$$\delta Q_I(\boldsymbol{X}) \approx N_\alpha(\boldsymbol{X}) \delta\hat{Q}_{I\alpha}. \tag{33}$$

The gradient fields are approximated via:

$$\frac{\partial U_I}{\partial X_J} \approx B_{\alpha J} \hat{U}_{I\alpha}; \quad B_{\alpha J} \coloneqq \frac{\partial N_\alpha(\boldsymbol{X})}{\partial X_J} \tag{34}$$

Using these approximations in the Euler–Lagrange Eqs. (5) and (6), the following discretized equations can be obtained:

$$\delta\hat{Q}_{I\alpha}\{\int_{\Omega^h} B_{\alpha J}(\boldsymbol{X})P_{JI}(\boldsymbol{X})\,d\Omega - \int_{\Omega^h} N_\alpha(\boldsymbol{X})\bar{B}_I(\boldsymbol{X})\,d\Omega - \int_{\Gamma_P^h} N_\alpha(\boldsymbol{X})\bar{T}_I(\boldsymbol{X})\,d\Gamma\} = 0 \tag{35}$$

$$\delta\hat{U}_{I\alpha}\left\{\int_{\Omega^h} B_{\alpha K}(\boldsymbol{X})\left[\boldsymbol{F}\cdot\mathbb{C}:(\boldsymbol{E}-\boldsymbol{E}^*) - \nabla^{\boldsymbol{X}}Q\cdot\boldsymbol{S}\right]_{KI}\,d\Omega\right\} = 0 \tag{36}$$

Since these equations hold for any admissible variations, we have the following discrete residual equations:

$$\mathcal{R}^Q_{I\alpha} = \int_{\Omega^h} B_{\alpha J}(\boldsymbol{X})P_{JI}(\boldsymbol{X})\,d\Omega - \int_{\Omega^h} N_\alpha(\boldsymbol{X})\bar{B}_I(\boldsymbol{X})\,d\Omega - \int_{\Gamma_P^h} N_\alpha(\boldsymbol{X})\bar{T}_I(\boldsymbol{X})\,d\Gamma = 0 \tag{37}$$

$$\mathcal{R}^U_{I\alpha} = \int_{\Omega^h} B_{\alpha K}(\boldsymbol{X})\left[\boldsymbol{F}\cdot\mathbb{C}:(\boldsymbol{E}-\boldsymbol{E}^*) - \nabla^{\boldsymbol{X}}Q\cdot\boldsymbol{S}\right]_{KI}\,d\Omega = 0 \tag{38}$$

The above system of equations $\boldsymbol{\mathcal{R}}^Q = \boldsymbol{0}$ and $\boldsymbol{\mathcal{R}}^U = \boldsymbol{0}$ are coupled and nonlinear with respect to the unknown nodal solutions $\hat{\boldsymbol{Q}}$ and $\hat{\boldsymbol{U}}$. We find the solution iteratively via the Newton–Raphson method. The tangent operators for the Newton–Raphson method are provided below:

$$\frac{\partial\mathcal{R}^Q_{I\alpha}}{\partial\hat{Q}_{J\beta}} = \int_{\Omega^h} B_{\alpha K}F_{KO}\mathbb{C}_{OIMJ}F_{LM}B_{\beta L}\,d\Omega \tag{39}$$

$$\frac{\partial\mathcal{R}^Q_{I\alpha}}{\partial\hat{U}_{J\beta}} = \int_{\Omega^h} B_{\alpha K}S_{JI}B_{\beta K}\,d\Omega + \int_{\Omega^h} B_{\alpha K}F_{KL}\mathbb{C}_{LIJM}(\nabla^{\boldsymbol{X}}Q)_{OM}B_{\beta O}\,d\Omega \tag{40}$$

$$\frac{\partial\mathcal{R}^U_{I\alpha}}{\partial\hat{Q}_{J\beta}} = -\int_{\Omega^h} B_{\alpha L}S_{JI}B_{\beta L}\,d\Omega - \int_{\Omega^h} B_{\alpha K}(\nabla^{\boldsymbol{X}}Q)_{KO}\mathbb{C}_{OIMJ}F_{LM}B_{\beta L}\,d\Omega \tag{41}$$

$$\begin{aligned}\frac{\partial\mathcal{R}^U_{I\alpha}}{\partial\hat{U}_{J\beta}} &= \int_{\Omega^h} B_{\alpha K}\left[\mathbb{C}:(\boldsymbol{E}-\boldsymbol{E}^*)\right]_{IJ}B_{\beta K}\,d\Omega \\ &+ \int_{\Omega^h} B_{\alpha K}\left[F_{KL}\mathbb{C}_{LIJM}F_{OM} - (\nabla^{\boldsymbol{X}}Q)_{KL}\mathbb{C}_{LIJM}(\nabla^{\boldsymbol{X}}Q)_{OM}\right]B_{\beta O}\,d\Omega\end{aligned} \tag{42}$$

Appendix D. Hyperparametrs of neural networks used in numerical examples

This section provides the details on neural network hyperparameters in each example for third-party reproducibility. We use the `Python` version 3.7.12, the `NumPy` version 1.21.5 [64], and the `PyTorch` version 1.10.2 [65]; the version consistency is important for reproducibility due to the random seed generator and potential changes in the source code among different versions. First, we provide the common hyperparameters for all examples then we describe problem-specific changes.

The default values are *random seed* = 1, *mini-batch size* = 200, *activation function* = `ELU` [66], *number of residual blocks* = 1, *mini-batch random shuffle* is activated. In the context of deep learning, *mini-batch* refers to a sub sample of data used in stochastic gradient descent iterations.

The default optimization algorithm for training encoder and decoder parameters is `ADAM` with the default parameters suggested in [67]. To achieve more stable training, we use `ReduceLROnPlateau` which reduces the initial optimizer learning rate based on a predefined schedule with the default values *reduction factor* = 0.91 and *patience* = 50; refer to `PyTorch`'s Manuel for more explanation.

It is important to normalize the data before the training to make sure the scaling inconsistency of features does not impact the quality of the optimization [56]. Two common methods are called *min–max* and *uvar*. In the *min–max* method, which is the default in this work, each feature is linearly scaled such that the normalized feature has zero minimum and unit maximum. In the *uvar* method, each feature is linearly scaled such that the normalized feature has zero mean and unit variance.

All the neural networks are initialized by the He uniform approach [68] which is the default setup in `PyTorch`. During the neural network training, we save the best weights and biases based on the weighted objective value calculated on the training dataset.

D.1. 1D manufactured solution

The decoder and encoder networks have four hidden layers and each hidden layer has 30 hidden units. The initial learning rate is 0.0005. The learning rate reduction factor is changed to 0.85.

D.2. Bending test

The decoder and encoder networks have two hidden layers and each hidden layer has 10 hidden units. The initial learning rate is 0.001. The learning rate reduction factor is changed to 0.85.

D.3. Biaxial test with Fung's data

The decoder and encoder networks have four hidden layers and each hidden layer has 10 hidden units. The initial learning rate is 0.0002. The data normalization is changed to *uvar*.

D.4. Biaxial test with real data

The decoder and encoder networks have two hidden layers and each hidden layer has 10 hidden units. The initial learning rate is 0.0002. The data normalization is changed to *uvar*. The mini batch size is changed to 400.

Appendix E. Parametric studies on the 1D bar problem

Here, we provide a detailed numerical study regarding the effect of the relative contribution of each objective in the training optimization statement Eq. (12) on the manifold recovery performance. To this end, we set the weight associated with the reconstruction objective to $w_{\text{recn}} = 1$ and consider different possibilities for the other two objective weights as $w_{\text{lin}} \in \{0.001, 1, 1000\}$ and $w_{\text{iso}} \in \{0.001, 0, 1, 1000\}$.

In all cases reported in Tables 1, 2, and 3, the lower the weight of isometric condition is the more overfitted is the recovered manifold; see the reconstructed manifolds in the last columns. This observation is consistent with our earlier claim that the isometry condition may also regularize the optimization problem and avoid overfitting and noise recovery. We observe that when the relative effect of the reconstruction term is considerably less than other objectives, the reconstructed manifold tends to converge to a line fitted to the data; see the last rows. This is expected since the reconstruction term almost vanishes, and the optimizer tries to map a line to another line in the decoder branch.

A comparison among the third rows, where the weight of the isometric condition is the same in all three cases, but the weight of the linearity condition is different, shows that an appropriate balance between the isometry condition and linearity is essential. In this regard, the reconstructed manifold in the case where the weight of the linearity condition is too small is more erroneous than in the other two cases.

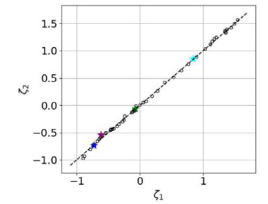
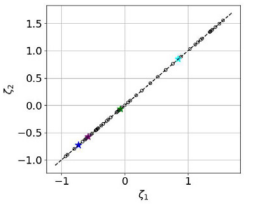
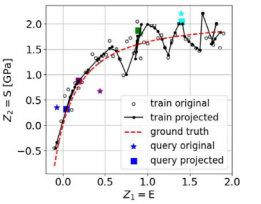
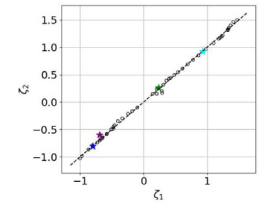
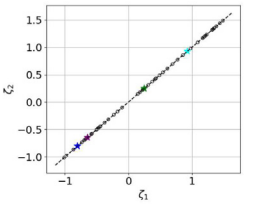
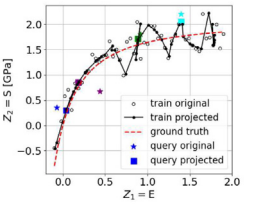
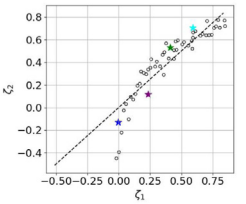
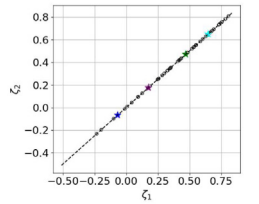
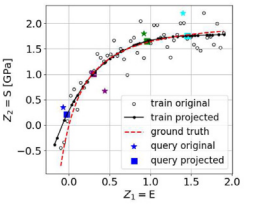
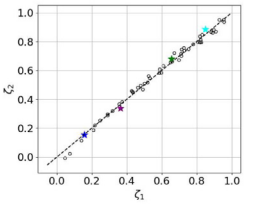
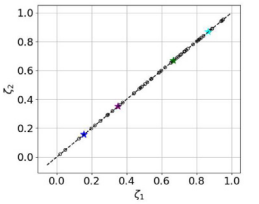
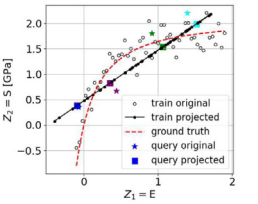
Appendix F. Local projection via nearest neighbor search

In all the numerical examples, the results obtained from the de-noised manifold embedding method is compared against the classical distance-minimization data-driven paradigm [2] where the local optimization problem is formulated as a nearest neighbor search (NNS) problem. We follow the fast NNS algorithm proposed in [69] using the KD-tree search defined for the distance function in Eq. (2). We use Mandel's representation (see Appendix G) and use factorization to enable the KD-tree search, i.e.,

$$\boldsymbol{E} : \mathbb{C} : \boldsymbol{E} = [\boldsymbol{E}]^T[\boldsymbol{C}][\boldsymbol{E}] = [\boldsymbol{E}]^T[\boldsymbol{C}_{\text{fact}}]^T[\boldsymbol{C}_{\text{fact}}][\boldsymbol{E}] = [\tilde{\boldsymbol{E}}]^T[\tilde{\boldsymbol{E}}] = \|[\tilde{\boldsymbol{E}}]\|_2^2, \tag{43}$$

where $[\tilde{\boldsymbol{E}}] = [\boldsymbol{C}_{\text{fact}}][\boldsymbol{E}]$ and $[\boldsymbol{C}_{\text{fact}}]$ is the factorization of the positive definite matrix $[\boldsymbol{C}]$ such that $[\boldsymbol{C}] = [\boldsymbol{C}_{\text{fact}}]^T[\boldsymbol{C}_{\text{fact}}]$. The kd-tree data structure designed for the Euclidean metric can be efficiently constructed for the transformed data. There are different choices for the $[\boldsymbol{C}_{\text{fact}}]$ but we use the matrix root $[\boldsymbol{C}_{\text{fact}}] = [\boldsymbol{C}]^{\frac{1}{2}}$; see [69] for more discussion.

Table 1
A comparison between the results obtained in each step of the introduced local projection for different penalty weight setting in the training optimization statement when $w_{\text{lin}} = 0.001$ and $w_{\text{recn}} = 1$ while the isometric enforcement varies.

Training objective	Latent before projection	Latent after projection	Reconstructed
$w_{\text{iso}} = 0$			
$w_{\text{iso}} = 0.001$			
$w_{\text{iso}} = 1$			
$w_{\text{iso}} = 1000$			

Appendix G. Mandel representation

In this study, the neural network training process is performed on the vector representation of the tensorial database. Also, the nearest neighbor search (needed only for the classical approach) is performed on the vector representation of the data. We use Mandel transformation to enable us to store 2nd-order tensor as vector and 4th-order tensor as matrix.

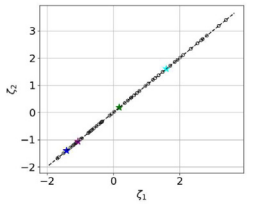
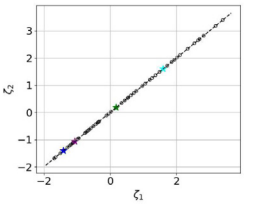
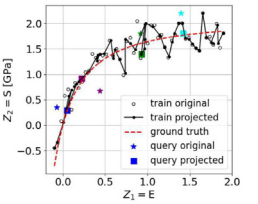
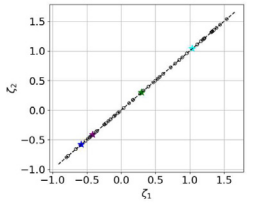
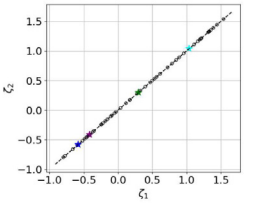
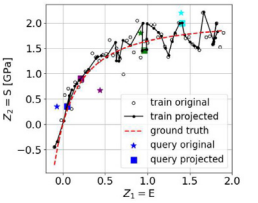
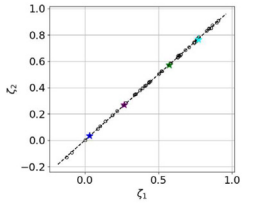
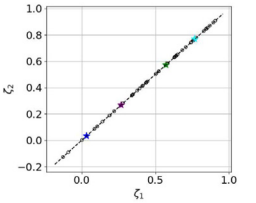
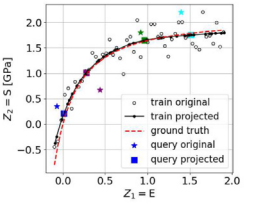
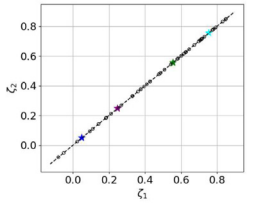
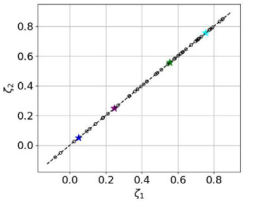
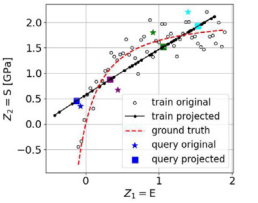
For the efficient data storage and the ease of computer implementation, we store symmetric 2nd and 4th order tensorial data via the Mandel representations. We prefer Mandel notation over the Voigt notation due to the (1) equivalence of spectral properties between the fourth order tensors and their matrix representations [70] and (2) the same treatment of strain and stress quantities in this notation.

In the Mandel notation, a 2nd order tensor $\boldsymbol{\epsilon} = \epsilon_{ij}\hat{\boldsymbol{e}}_i \otimes \hat{\boldsymbol{e}}_j$ with the symmetric condition ($\epsilon_{ij} = \epsilon_{ji}$) can be represented by an one-dimensional array $[\epsilon]$ as follows,

$$[\epsilon]^T = [\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \sqrt{2}\epsilon_{23}, \sqrt{2}\epsilon_{13}, \sqrt{2}\epsilon_{12}]^T \text{ in three dimensions} \tag{44}$$

$$[\epsilon]^T = [\epsilon_{11}, \epsilon_{22}, \sqrt{2}\epsilon_{12}]^T \text{ in two dimensions.} \tag{45}$$

Table 2
A comparison between the results obtained in each step of the introduced local projection for different penalty weight setting in the training optimization statement when $w_{\text{lin}} = 1$ and $w_{\text{recn}} = 1$ while the isometric enforcement varies.

Training objective	Latent before projection	Latent after projection	Reconstructed
$w_{\text{iso}} = 0$			
$w_{\text{iso}} = 0.001$			
$w_{\text{iso}} = 1$			
$w_{\text{iso}} = 1000$			

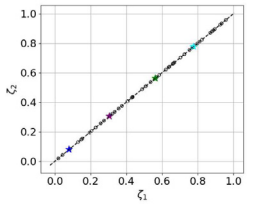
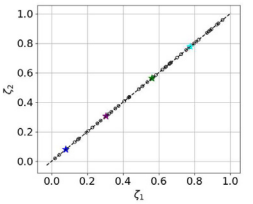
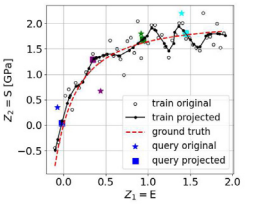
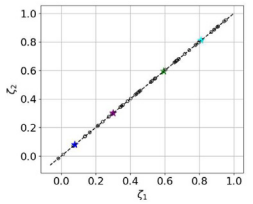
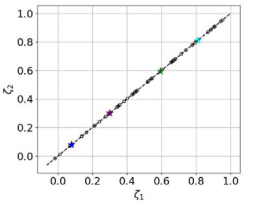
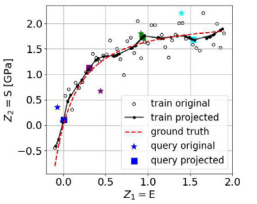
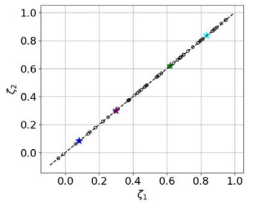
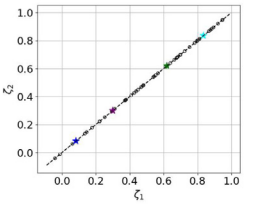
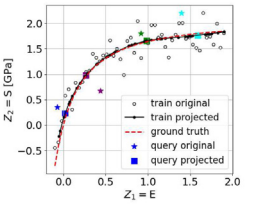
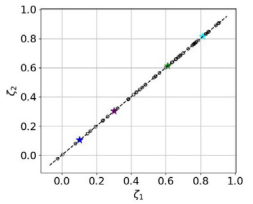
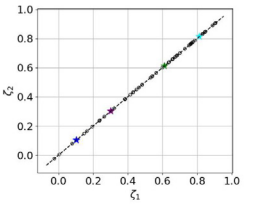
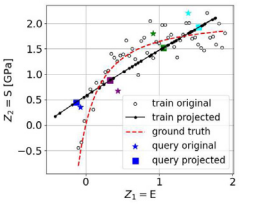
A forth order tensor $\mathbb{C} = \mathbb{C}_{ijkl}\hat{\boldsymbol{e}}_i \otimes \hat{\boldsymbol{e}}_j \otimes \hat{\boldsymbol{e}}_k \otimes \hat{\boldsymbol{e}}_l$ with the minor symmetry condition ($\mathbb{C}_{ijkl} = \mathbb{C}_{jikl} = \mathbb{C}_{ijlk}$) can be represented by a two dimensional array $[\mathbb{C}]$ as follows,

$$[\mathbb{C}] = \begin{bmatrix} \mathbb{C}_{1111} & \mathbb{C}_{1122} & \mathbb{C}_{1133} & \sqrt{2}\mathbb{C}_{1123} & \sqrt{2}\mathbb{C}_{1113} & \sqrt{2}\mathbb{C}_{1112} \\ \mathbb{C}_{2211} & \mathbb{C}_{2222} & \mathbb{C}_{2233} & \sqrt{2}\mathbb{C}_{2223} & \sqrt{2}\mathbb{C}_{2213} & \sqrt{2}\mathbb{C}_{2212} \\ \mathbb{C}_{3311} & \mathbb{C}_{3322} & \mathbb{C}_{3333} & \sqrt{2}\mathbb{C}_{3323} & \sqrt{2}\mathbb{C}_{3313} & \sqrt{2}\mathbb{C}_{3312} \\ \sqrt{2}\mathbb{C}_{2311} & \sqrt{2}\mathbb{C}_{2322} & \sqrt{2}\mathbb{C}_{2333} & 2\mathbb{C}_{2323} & 2\mathbb{C}_{2313} & 2\mathbb{C}_{2312} \\ \sqrt{2}\mathbb{C}_{1311} & \sqrt{2}\mathbb{C}_{1322} & \sqrt{2}\mathbb{C}_{1333} & 2\mathbb{C}_{1323} & 2\mathbb{C}_{1313} & 2\mathbb{C}_{1312} \\ \sqrt{2}\mathbb{C}_{1211} & \sqrt{2}\mathbb{C}_{1222} & \sqrt{2}\mathbb{C}_{1233} & 2\mathbb{C}_{1223} & 2\mathbb{C}_{1213} & 2\mathbb{C}_{1212} \end{bmatrix} \quad \text{in three dimensions,} \tag{46}$$

$$[\mathbb{C}] = \begin{bmatrix} \mathbb{C}_{1111} & \mathbb{C}_{1122} & \sqrt{2}\mathbb{C}_{1112} \\ \mathbb{C}_{2211} & \mathbb{C}_{2222} & \sqrt{2}\mathbb{C}_{2212} \\ \sqrt{2}\mathbb{C}_{1211} & \sqrt{2}\mathbb{C}_{1222} & 2\mathbb{C}_{1212} \end{bmatrix} \quad \text{in two dimensions.} \tag{47}$$

In this paper, tensorial variables introduced by brackets (i.e., $[\cdot]$) are represented by the Mandel notation, unless otherwise specified.

Table 3
A comparison between the results obtained in each step of the introduced local projection for different penalty weight setting in the training optimization statement when $w_{\text{lin}} = 1000$ and $w_{\text{recn}} = 1$ while the isometric enforcement varies.

Training objective	Latent before projection	Latent after projection	Reconstructed
$w_{\text{iso}} = 0$			
$w_{\text{iso}} = 0.001$			
$w_{\text{iso}} = 1$			
$w_{\text{iso}} = 1000$			

References

[1] Sam Madden, From databases to big data, IEEE Internet Comput. 16 (3) (2012) 4–6.
[2] Trenton Kirchdoerfer, Michael Ortiz, Data-driven computational mechanics, Comput. Methods Appl. Mech. Engrg. 304 (2016) 81–101.
[3] James R. Rice, Inelastic constitutive relations for solids: an internal-variable theory and its application to metal plasticity, J. Mech. Phys. Solids 19 (6) (1971) 433–455.
[4] Robert Eggersmann, Trenton Kirchdoerfer, Stefanie Reese, Laurent Stainier, Michael Ortiz, Model-free data-driven inelasticity, Comput. Methods Appl. Mech. Engrg. 350 (2019) 81–99.
[5] Moritz Flaschel, Siddhant Kumar, Laura De Lorenzis, Discovering plasticity models without stress data, Npj Comput. Mater. 8 (1) (2022) 1–10.
[6] Joseph Kestin, James R. Rice, Paradoxes in the Application of Thermodynamics to Strained Solids, Citeseer, 1969.
[7] Kun Wang, WaiChing Sun, Qiang Du, A cooperative game for automated learning of elasto-plasticity knowledge graphs and models with AI-guided experimentation, Comput. Mech. 64 (2) (2019) 467–499.
[8] Kun Wang, WaiChing Sun, Simon Salager, SeonHong Na, Ghonwa Khaddour, Identifying material parameters for a micro-polar plasticity model via X-ray micro-computed tomographic (CT) images: lessons learned from the curve-fitting exercises, Int. J. Multiscale Comput. Eng. 14 (4) (2016).
[9] Jamie Callan, Alistair Moffat, Panel on use of proprietary data, in: ACM SIGIR Forum, Vol. 46, ACM, New York, NY, USA, 2012, pp. 10–18.
[10] Peter Coffman, The data-centric corporation–Schlumberger in the 21st century, Oil Gas J. (2000) 8.
[11] Trenton Kirchdoerfer, Michael Ortiz, Data driven computing with noisy material data sets, Comput. Methods Appl. Mech. Engrg. 326 (2017) 622–641.
[12] Qizhi He, Jiun-Shyan Chen, A physics-constrained data-driven approach based on locally convex reconstruction for noisy database, Comput. Methods Appl. Mech. Engrg. 363 (2020) 112791.

[13] Bahador Bahmani, WaiChing Sun, Manifold embedding data-driven mechanics, J. Mech. Phys. Solids (2022) 104927.
[14] Robert Eggersmann, Laurent Stainier, Michael Ortiz, Stefanie Reese, Model-free data-driven computational mechanics enhanced by tensor voting, Comput. Methods Appl. Mech. Engrg. 373 (2021) 113499.
[15] Yoshihiro Kanno, A kernel method for learning constitutive relation in data-driven computational elasticity, Japan J. Ind. Appl. Math. 38 (1) (2021) 39–77.
[16] Jacobo Ayensa-Jiménez, Mohamed H. Doweidar, Jose A. Sanz-Herrera, Manuel Doblaré, A new reliability-based data-driven approach for noisy experimental data with physical constraints, Comput. Methods Appl. Mech. Engrg. 328 (2018) 752–774.
[17] Xiaolong He, Qizhi He, Jiun-Shyan Chen, Deep autoencoders for physics-constrained data-driven nonlinear materials modeling, Comput. Methods Appl. Mech. Engrg. 385 (2021) 114034.
[18] Matthias Hein, Markus Maier, Manifold denoising, Adv. Neural Inf. Process. Syst. 19 (2006).
[19] Matthias Hein, Markus Maier, Manifold denoising as preprocessing for finding natural representations of data, in: AAAI, 2007, pp. 1646–1649.
[20] Lu Trong Khiem Nguyen, Marc-André Keip, A data-driven approach to nonlinear elasticity, Comput. Struct. 194 (2018) 97–115.
[21] Geoffrey E. Hinton, Ruslan R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[22] Changshuo Liu, Nicolas Boumal, Simple algorithms for optimization on Riemannian manifolds with constraints, Appl. Math. Optim. 82 (3) (2020) 949–981.
[23] Kurt Hornik, Maxwell Stinchcombe, Halbert White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.
[24] Randall Balestriero, Jerome Pesenti, Yann LeCun, Learning in high dimension always amounts to extrapolation, 2021, arXiv preprint arXiv:2110.09485.
[25] Todd K. Moon, Wynn C. Stirling, Mathematical Methods and Algorithms for Signal Processing, 2000, Number 621.39: 51 MON.
[26] Serge Lang, Differential and Riemannian Manifolds, Vol. 160, Springer Science & Business Media, 2012.
[27] Andrew N. Pressley, Elementary Differential Geometry, Springer Science & Business Media, 2010.
[28] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, Chelsea Finn, Gradient surgery for multi-task learning, Adv. Neural Inf. Process. Syst. 33 (2020) 5824–5836.
[29] Bahador Bahmani, WaiChing Sun, Training multi-objective/multi-task collocation physics-informed neural network with student/teachers transfer learnings, 2021, arXiv preprint arXiv:2107.11496.
[30] Qizhi He, Devin W. Laurence, Chung-Hao Lee, Jiun-Shyan Chen, Manifold learning based data-driven modeling for soft biological tissues, J. Biomech. 117 (2021) 110124.
[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[32] Richard Lippmann, An introduction to computing with neural nets, IEEE Assp Mag. 4 (2) (1987) 4–22.
[33] Andreas Veit, Michael J. Wilber, Serge Belongie, Residual networks behave like ensembles of relatively shallow networks, Adv. Neural Inf. Process. Syst. 29 (2016).
[34] Bo Chang, Lili Meng, Eldad Haber, Lars Ruthotto, David Begert, Elliot Holtham, Reversible architectures for arbitrarily deep residual neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[35] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, Daniele Panozzo, Deep geometric prior for surface reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10130–10139.
[36] Mian Xiao, WaiChing Sun, Geometric prior of multi-resolution yielding manifolds and the local closest point projection for nearly non-smooth plasticity, Comput. Methods Appl. Mech. Engrg. 400 (2022) 115469.
[37] Yoshihiro Kanno, Simple heuristic for data-driven computational elasticity with material data involving noise and outliers: a local robust regression approach, Japan J. Ind. Appl. Math. 35 (3) (2018) 1085–1101.
[38] Samuel Jett, Devin Laurence, Robert Kunkel, Anju R. Babu, Katherine Kramer, Ryan Baumwart, Rheal Towner, Yi Wu, Chung-Hao Lee, Biaxial mechanical data of porcine atrioventricular valve leaflets, Data Brief 21 (2018) 358–363.
[39] Samuel Jett, Devin Laurence, Robert Kunkel, Anju R. Babu, Katherine Kramer, Ryan Baumwart, Rheal Towner, Yi Wu, Chung-Hao Lee, An investigation of the anisotropic mechanical properties and anatomical structure of porcine atrioventricular heart valves, J. Mech. Behav. Biomed. Mater. 87 (2018) 155–171.
[40] Jeffrey D. Banfield, Adrian E. Raftery, Model-based Gaussian and non-Gaussian clustering, Biometrics (1993) 803–821.
[41] Christian Soize, Roger Ghanem, Physical systems with random uncertainties: chaos representations with arbitrary probability measure, SIAM J. Sci. Comput. 26 (2) (2004) 395–410.
[42] Paolo Bocchini, George Deodatis, Critical review and latest developments of a class of simulation algorithms for strongly non-Gaussian random fields, Probab. Eng. Mech. 23 (4) (2008) 393–407.
[43] Gordon A. Fenton, Random field modeling of CPT data, J. Geotech. Geoenviron. Eng. 125 (6) (1999) 486.
[44] Xavier Glorot, Antoine Bordes, Yoshua Bengio, Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.
[45] Jonathan Ho, Ajay Jain, Pieter Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. 33 (2020) 6840–6851.
[46] Alexander Quinn Nichol, Prafulla Dhariwal, Improved denoising diffusion probabilistic models, in: International Conference on Machine Learning, PMLR, 2021, pp. 8162–8171.
[47] Nikolaos N. Vlassis, Puhan Zhao, Ran Ma, Tommy Sewell, WaiChing Sun, Molecular dynamics inferred transfer learning models for finite-strain hyperelasticity of monoclinic crystals: Sobolev training and validations against physical constraints, Internat. J. Numer. Methods Engrg. (2022).

[48] Faisal As' ad, Philip Avery, Charbel Farhat, A mechanics-informed artificial neural network approach in data-driven constitutive modeling, in: AIAA Scitech 2022 Forum, 2022, p. 0100.
[49] Kailai Xu, Daniel Z. Huang, Eric Darve, Learning constitutive relations using symmetric positive definite neural networks, J. Comput. Phys. 428 (2021) 110072.
[50] Dominik K. Klein, Mauricio Fernández, Robert J. Martin, Patrizio Neff, Oliver Weeger, Polyconvex anisotropic hyperelasticity with neural networks, J. Mech. Phys. Solids 159 (2022) 104703.
[51] Nikolaos N. Vlassis, Ran Ma, WaiChing Sun, Geometric deep learning for computational mechanics Part I: Anisotropic Hyperelasticity, Comput. Methods Appl. Mech. Engrg. 371 (2020) 113299.
[52] Yousef Heider, Kun Wang, WaiChing Sun, SO (3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials, Comput. Methods Appl. Mech. Engrg. 363 (2020) 112875.
[53] Patrick Weber, Jeremy Geiger, Werner Wagner, Constrained neural network training and its application to hyperelastic material modeling, Comput. Mech. 68 (5) (2021) 1179–1204.
[54] Patrick J. Roache, Code verification by the method of manufactured solutions, J. Fluids Eng. 124 (1) (2002) 4–10.
[55] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, et al., SymPy: symbolic computing in python, PeerJ Comput. Sci. 3 (2017) e103.
[56] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
[57] Michael S. Sacks, Biaxial mechanical evaluation of planar biological materials, J. Elast. Phys. Sci. Solids 61 (1) (2000) 199–246.
[58] Pin Tong, Yuang-Cheng Fung, The stress-strain relationship for the skin, J. Biomech. 9 (10) (1976) 649–657.
[59] Chiara Bellini, Paul Glass, Elena S. Di Martino, Constitutive modeling of the small intestine, in: Biomechanics of Living Organs, Elsevier, 2017, pp. 287–305.
[60] Robert Eggersmann, Laurent Stainier, Michael Ortiz, Stefanie Reese, Efficient data structures for model-free data-driven computational mechanics, Comput. Methods Appl. Mech. Engrg. 382 (2021) 113855.
[61] Jerrold E. Marsden, Thomas J.R. Hughes, Mathematical Foundations of Elasticity, Courier Corporation, 1994.
[62] Robert Simon Fong, Peter Tino, Riemannian geometry: A brief overview, in: Population-Based Optimization on Riemannian Manifolds, Springer, 2022, pp. 11–26.
[63] John M. Lee, Riemannian Manifolds: An Introduction to Curvature, Vol. 176, Springer Science & Business Media, 2006.
[64] Charles R. Harris, K. Jarrod Millman, Stéfan J. Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, et al., Array programming with NumPy, Nature 585 (7825) (2020) 357–362.
[65] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019) 8026–8037.
[66] Djork-Arné Clevert, Thomas Unterthiner, Sepp Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), 2015, arXiv preprint arXiv:1511.07289.
[67] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[68] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
[69] Bahador Bahmani, WaiChing Sun, A kd-tree-accelerated hybrid data-driven/model-based approach for poroelasticity problems with multi-fidelity multi-physics data, Comput. Methods Appl. Mech. Engrg. 382 (2021) 113868.
[70] Morteza M. Mehrabadi, Stephen C. Cowin, Eigentensors of linear anisotropic elastic materials, Quart. J. Mech. Appl. Math. 43 (1) (1990) 15–41.