Deep Autoencoders for Nonlinear Physics-Constrained Data-Driven Computational Framework with Application to Biological Tissue Modeling

Xiaolong He[1], **Qizhi He**[2], **Jiun-Shyan Chen**[1]
[1] Department of Structural Engineering, University of California, San Diego, La Jolla, CA 92093
xiaolong-he@ucsd.edu, js-chen@ucsd.edu
[2] Physical and Computational Sciences Directorate, Pacific Northwest National Laboratory, Richland, WA 99354
qizhi.he@pnnl.gov

Abstract

Physics-constrained data-driven computing is an emerging paradigm that directly integrates material database into physical simulations of complex materials, bypassing the construction of classical constitutive models. However, most of the developed data-driven computing approaches are based on simplistic distance minimization and thus suffer from dealing with high-dimensional applications and lack generalization ability. This study proposes a deep learning enhanced data-driven computing framework to address these fundamental issues for nonlinear materials modeling. To this end, an autoencoder, a special multi-layer neural network architecture, is introduced to learn the underlying low-dimensional embedding representation of the material database. Incorporating the offline trained autoencoder and the discovered embedding space in online data-driven computation enables to search for the optimal material state from database in low-dimensional embedding space, enhancing the robustness and predictability of data-driven computing on limited material data. To enhance stability and convergence of data-driven computing, a convexity-preserving interpolation scheme is introduced for constructing the material state on the low-dimensional embedding space given by autoencoders. The effectiveness and enhanced generalization performance of the proposed approach are examined by modeling biological tissue with experimental data.

1 Nonlinear physics-constrained data-driven modeling

The physical equations governing the deformation of an elastic solid in a domain Ω^X bounded by a Neumann boundary Γ_t^X and a Dirichlet boundary Γ_u^X are given as

$$\begin{cases} DIV(\mathbf{F}(\mathbf{u})\cdot\mathbf{S})+\mathbf{b}=\mathbf{0}, & \text{in } \Omega^X,\\ \mathbf{E}=\mathbf{E}(\mathbf{u})=(\mathbf{F}^T\mathbf{F}-\mathbf{I})/2, & \text{in } \Omega^X,\\ (\mathbf{F}(\mathbf{u})\cdot\mathbf{S})\cdot\mathbf{N}=\mathbf{t}, & \text{on } \Gamma_t^X,\\ \mathbf{u}=\mathbf{g}, & \text{on } \Gamma_u^X, \end{cases} \tag{1}$$

where $\mathbf{u}$ is the displacement vector, $\mathbf{E}$ is the Green Lagrangian strain tensor, and $\mathbf{S}$ is the second Piola-Kirchhoff (2nd-PK) stress tensor. The superscript X denotes the reference (undeformed) configuration. $\mathbf{F}$ is the deformation gradient related to $\mathbf{u}$, defined as $\mathbf{F}(\mathbf{u})=\partial(\mathbf{X}+\mathbf{u})/\partial\mathbf{X}$, where $\mathbf{X}$ is the material coordinate. $\mathbf{b}$, $\mathbf{N}$, $\mathbf{t}$, and $\mathbf{g}$ are the body force, the surface normal on Γ_t^X, the traction on Γ_t^X, and the prescribed displacement on Γ_u^X, respectively.

To solve the boundary value problem (BVP) in Eq. (1), material laws describing the stress-strain relation are required. However, it is difficult to construct phenomenological material models for complex material systems. The physics-constrained data-driven computing framework (Kirchdoerfer and Ortiz 2016; Ibanez et al. 2018; He and Chen 2020) offers an alternative to directly utilize material data in physical simulations by formulating the BVP as an optimization problem under physical constraints.

In this framework, the material behavior is described by strain-stress pairs, $\hat{\mathbf{z}}=(\hat{\mathbf{E}},\hat{\mathbf{S}})$, defined as the *material state* given by the material database, $\mathbb{E}=\{\hat{\mathbf{z}}_I\}_{I=1}^M\in\mathcal{E}$, where M is the number of material data points and $\mathcal{E}$ is an admissible set of material database over the domain Ω. The strain-stress pairs, $\mathbf{z}=(\mathbf{E},\mathbf{S})\in\mathcal{C}$, that satisfy the physical equations (Eq. (1)) are defined as the *physical state*, where $\mathcal{C}=\{\mathbf{z}|\text{ Eq. (1)}\}$ is a physical admissible set. The data-driven solutions are obtained by fixed-point global-local iterations to minimize the distance between the material and physical states. In the *local step*, the material data-driven local solver searches for the optimal strain-stress pairs $\hat{\mathbf{z}}^*$ closest to a given physical state $\mathbf{z}$ by minimizing a distance functional defined as follows:

$$\mathcal{F}(\mathbf{z},\hat{\mathbf{z}}^*)=\min_{\hat{\mathbf{z}}\in\mathcal{E}}\int_{\Omega^X} d_z^2(\mathbf{z},\hat{\mathbf{z}})d\Omega, \tag{2}$$

with

$$d_z^2(\mathbf{z},\hat{\mathbf{z}})=d_E^2(\mathbf{E}(\mathbf{u}),\hat{\mathbf{E}})+d_S^2(\mathbf{S},\hat{\mathbf{S}}), \tag{3a}$$

$$d_E^2(\mathbf{E}(\mathbf{u}),\hat{\mathbf{E}})=\frac{1}{2}(\mathbf{E}(\mathbf{u})-\hat{\mathbf{E}}):\hat{\mathbb{C}}:(\mathbf{E}(\mathbf{u})-\hat{\mathbf{E}}), \tag{3b}$$

$$d_S^2(\mathbf{S},\hat{\mathbf{S}})=\frac{1}{2}(\mathbf{S}-\hat{\mathbf{S}}):\hat{\mathbb{C}}^{-1}:(\mathbf{S}-\hat{\mathbf{S}}), \tag{3c}$$

where $\hat{\mathbb{C}}$ is a predefined symmetric and positive-definite tensor used to properly regulate the distance between $\mathbf{z}$ and $\hat{\mathbf{z}}$.

Given the optimal material data $\hat{\mathbf{z}}^* = (\hat{\mathbf{E}}^*, \hat{\mathbf{S}}^*)$ obtained from the local step in Eq. (2), the *global step* of the data-driven problem is to search for the closest physical state, formulated as follows:

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{S}} \mathcal{F}(\mathbf{E}(\mathbf{u}), \mathbf{S}; \hat{\mathbf{E}}^*, \hat{\mathbf{S}}^*) \\ \text{subject to: } &DIV(\mathbf{F}(\mathbf{u}) \cdot \mathbf{S}) + \mathbf{b} = \mathbf{0} \ \text{ in } \ \Omega^X, \\ &(\mathbf{F}(\mathbf{u}) \cdot \mathbf{S}) \cdot \mathbf{N} = \mathbf{t} \ \text{ on } \ \Gamma_t^X, \end{aligned} \tag{4}$$

which can be solved by the Lagrange multiplier method and a nonlinear numerical solver. Note that the compatibility constraint in Eq. (1b) is directly enforced by computing $\mathbf{E}$ from $\mathbf{u}$.

2 Autoencoders for nonlinear material manifold learning

Autoencoders (DeMers and Cottrell 1993; Hinton and Salakhutdinov 2006) aim to optimally copy its input to output and retain the most representative features in a low-dimensional embedding layer. Hence, it allows for effective noise filtering, dimensionality reduction, and hidden structure discovery of data. As shown in Fig. 1, an autoencoder contains an encoder function $\mathbf{h}_{\text{enc}}(\cdot;\boldsymbol{\theta}_{\text{enc}}) : \mathbb{R}^d \to \mathbb{R}^p$ and a decoder function $\mathbf{h}_{\text{dec}}(\cdot;\boldsymbol{\theta}_{\text{dec}}) : \mathbb{R}^p \to \mathbb{R}^d$, such that the autoencoder is

$$\tilde{\mathbf{x}} = \mathbf{h}(\mathbf{x}; \boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}) := \mathbf{h}_{\text{dec}}(\cdot;\boldsymbol{\theta}_{\text{dec}}) \circ \mathbf{h}_{\text{enc}}(\mathbf{x};\boldsymbol{\theta}_{\text{enc}}), \tag{5}$$

where d is the input dimension, $p < d$ is the embedding dimension, $\boldsymbol{\theta}_{\text{enc}}$ and $\boldsymbol{\theta}_{\text{dec}}$ are the trainable parameters of the encoder and the decoder, respectively. $\tilde{\mathbf{x}}$ is the output of the autoencoder, a reconstruction of the original input $\mathbf{x}$. With $p < d$, the encoder $\mathbf{h}_{\text{enc}}$ is trained to learn an intrinsic representation of $\mathbf{x} \in \mathbb{R}^d$, denoted as the embedding $\mathbf{x}' \in \mathbb{R}^p$, whereas the decoder $\mathbf{h}_{\text{dec}}$ is trained to reconstruct the input data by mapping the embedding $\mathbf{x}'$ back to the high-dimensional space $\tilde{\mathbf{x}} \in \mathbb{R}^d$.

In this study, autoencoders are employed to discover the intrinsic low-dimensional material embedding of the given material dataset $\mathbb{E} = \{\hat{\mathbf{z}}_I\}_{I=1}^M$, where $\hat{\mathbf{z}}_I = (\hat{\mathbf{E}}_I, \hat{\mathbf{S}}_I)$. The optimal parameters $(\boldsymbol{\theta}^*_{\text{enc}}, \boldsymbol{\theta}^*_{\text{dec}})$ of the autoencoder $\mathbf{h}(\cdot;\boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}})$ are obtained by minimizing the following loss function:

$$\begin{aligned} (\boldsymbol{\theta}^*_{\text{enc}}, \boldsymbol{\theta}^*_{\text{dec}}) = \arg\min_{\boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}} &\frac{1}{M} \sum_{I=1}^{M} ||\mathbf{h}(\hat{\mathbf{z}}_I; \boldsymbol{\theta}_{\text{enc}}, \boldsymbol{\theta}_{\text{dec}}) - \hat{\mathbf{z}}_I||^2 \\ &+ \beta \sum_{l=1}^{L+1} ||\mathbf{W}^{(l)}||_F^2, \end{aligned} \tag{6}$$

where L is the number of hidden layers, $\mathbf{W}$ is the weight coefficient of the autoencoder, β is a regularization parameter, and $||\cdot||_F$ denotes the Frobenius norm. The first term in the loss function is the reconstruction error of all training data and the second term is a L_2-norm based weight regularization term used to avoid over-fitting (Goodfellow, Bengio, and Courville 2016).

With the trained autoencoder $\mathbf{h}(\cdot;\boldsymbol{\theta}^*_{\text{enc}}, \boldsymbol{\theta}^*_{\text{dec}})$, a low-dimensional embedding space can be defined, i.e., $\mathcal{E}' = \{\mathbf{z}' \in \mathbb{R}^p | \mathbf{z}' = \mathbf{h}_{\text{enc}}(\mathbf{z}; \boldsymbol{\theta}^*_{\text{enc}}), \forall \mathbf{z} \in \mathcal{Z}\}$, in which the material state is described by a lower-dimensional coordinate system $\mathbf{z}'$. Here, the prime symbol $(\cdot)'$ is used to denote the quantities defined in the embedding space, and $\mathcal{Z}$ denotes the high-dimensional phase space where the material states $\hat{\mathbf{z}}$ and the physical states $\mathbf{z}$ are defined. For example, the embedding set of the given material data is $\mathbb{E}' = \{\hat{\mathbf{z}}'_I\}_{I=1}^M \subset \mathcal{E}'$, where $\hat{\mathbf{z}}'_I = \mathbf{h}_{\text{enc}}(\hat{\mathbf{z}}_I; \boldsymbol{\theta}^*_{\text{enc}})$ for $\hat{\mathbf{z}}_I \in \mathbb{E}$.

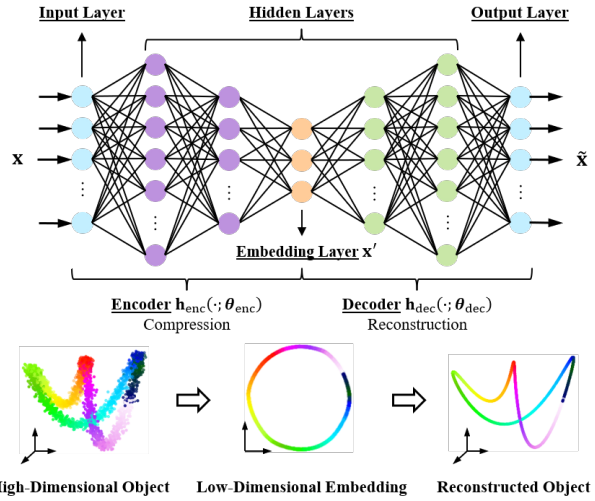

Figure 1: Schematic of an autoencoder consisting of an encoder and a decoder, with the embedding dimension smaller than the input dimension. The encoder learns an intrinsic low-dimensional embedding of a high-dimensional input object, whereas the decoder optimally reconstructs the input object from the low-dimensional embedding.

3 Auto-embedding data-driven (AEDD) solver

The nonlinear physics-constrained data-driven computing framework described in Section 1 is conducted in the high-dimensional phase space $\mathcal{Z}$ (called *data space*), with the physical state $\mathbf{z}_\alpha \in \mathcal{C}$, the material data $\hat{\mathbf{z}}_\alpha \in \mathcal{E}$, and the material dataset $\mathbb{E}$ defined in $\mathcal{Z}$. The subscript "α" is used to denote the quantities at integration points. To enhance solution accuracy and generalization ability of data-driven computing, deep manifold learning enabled by autoencoders is introduced into the material data-driven local solver. The autoencoders are trained offline and the trained encoder $\mathbf{h}_{\text{enc}}$ and decoder $\mathbf{h}_{\text{dec}}$ functions are employed directly in the online data-driven computation. As such, the encoder maps an arbitrary point from the data space to the embedding space, i.e. $\mathbf{z}'_\alpha = \mathbf{h}_{\text{enc}}(\mathbf{z}_\alpha)$, whereas the decoder performs the reverse mapping, i.e. $\tilde{\mathbf{z}}_\alpha = \mathbf{h}_{\text{dec}}(\mathbf{z}'_\alpha)$.

In the proposed AEDD local solver, the local step defined in Eq. (2) is reformulated by three steps:

$$\textit{Step 1}: \quad \mathbf{z}'_\alpha = \mathbf{h}_{\text{enc}}(\mathbf{z}_\alpha), \tag{7a}$$

$$\textit{Step 2}: \quad \hat{\mathbf{z}}'^*_\alpha = \mathcal{I}\left(\{\Psi_I(\mathbf{z}'_\alpha); \hat{\mathbf{z}}'_I\}_{I \in \mathcal{N}_k(\mathbf{z}'_\alpha)}\right) \tag{7b}$$

$$\textit{Step 3}: \quad \hat{\mathbf{z}}^*_\alpha = \mathbf{h}_{\text{dec}}(\hat{\mathbf{z}}'^*_\alpha), \tag{7c}$$

for $\alpha = 1, ..., N_{int}$, where $\hat{\mathbf{z}}'_I \in \mathbb{E}'$, N_{int} is the number of integration points, and $\mathcal{I}$ is the convexity-preserving interpolation operator defined as

$$\mathbf{z}'_{recon} = \mathcal{I}\left(\{\Psi_I(\mathbf{z}'); \hat{\mathbf{z}}'_I\}_{I\in\mathcal{N}_k(\mathbf{z}')}\right) = \sum_{I\in\mathcal{N}_k(\mathbf{z}')} \Psi_I(\mathbf{z}')\hat{\mathbf{z}}'_I, \tag{8}$$

where $\mathbf{z}'_{recon}$ is the reconstruction of $\mathbf{z}'$, $\hat{\mathbf{z}}'_I$ is the material data embedding in $\mathbb{E}'$, $\mathcal{N}_k(\mathbf{z}')$ is the index set of the k nearest neighbor points of $\mathbf{z}'$ selected from $\mathbb{E}'$. The interpolation functions are

$$\Psi_I(\mathbf{z}') = \frac{\phi(\mathbf{z}' - \hat{\mathbf{z}}'_I)}{\sum_{J=1}\phi(\mathbf{z}' - \hat{\mathbf{z}}'_J)}, \tag{9}$$

where $\phi(\mathbf{z}' - \hat{\mathbf{z}}'_I) = 1/||\mathbf{z}' - \hat{\mathbf{z}}'_I||^2$ is a positive kernel function representing the weight on the data set $\{\hat{\mathbf{z}}'_I\}_{I\in\mathcal{N}_k(\mathbf{z}')}$. The positive interpolation functions satisfy the partition of unity, $\sum_{I\in\mathcal{N}_k(\mathbf{z}')}\Psi_I(\mathbf{z}') = 1$, which ensures transformation objectivity and convexity of the interpolation scheme, Eq. (8).

The schematic of data-driven computing with the AEDD local solver is illustrated in Fig. 2, where the integration point index α is dropped for brevity. For example, at the v-th global-local iteration, after the physical state $\mathbf{z}^{(v)}$ (the blue-filled triangle) is obtained from the global step (Eq. (4)), *Step 1* of the local solver (Eq. (7a)) maps the sought physical state from the data space to the embedding space by the encoder, $\mathbf{z}'^{(v)} = \mathbf{h}_{\text{enc}}(\mathbf{z}^{(v)})$, depicted by the white-filled triangle in Fig. 2. In *Step 2*, k nearest neighbors of $\mathbf{z}'^{(v)}$ based on Euclidean distance are sought in the embedding space and the optimal material embedding solution $\hat{\mathbf{z}}'^{*(v)}$ (the red square) is reconstructed by using the proposed convexity-preserving interpolation (Eqs. (8-9)). Lastly, in *Step 3*, the optimal material embedding state $\hat{\mathbf{z}}'^{*(v)}$ is transformed from the embedding space to the data space by the decoder, $\hat{\mathbf{z}}^{*(v)} = \mathbf{h}_{\text{dec}}(\hat{\mathbf{z}}'^{*(v)})$ (the red star in Fig. 2). Subsequently, this material state $\hat{\mathbf{z}}^{*(v)}$ from the local solver in Eq. (7) is used in the next physical state update $\mathbf{z}^{(v+1)}$. This process completes one global-local iteration. The iterations proceed until the distance between the physical and material states is within a tolerance, yielding the data-driven solution denoted by the green star in Fig. 2, which ideally is the intersection between the physical manifold and material manifold in the data space.

Here, the nearest neighbors searching and locally convex reconstruction of the optimal material state are processed in the filtered (noiseless) low-dimensional embedding space, resulting in the enhanced robustness against noise and accuracy of the local solution.

4 Data-driven biological tissue modeling

The effectiveness of the proposed AEDD computational framework is evaluated by modeling biological heart valve tissue with data from biaxial mechanical experiments on one representative porcine mitral valve posterior leaflet (Jett et al. 2018), see a schematic in Fig. 3a. There are eleven protocols, including *nine biaxial tension protocols* (1–9) with various biaxial tension ratios and *two pure shear protocols*

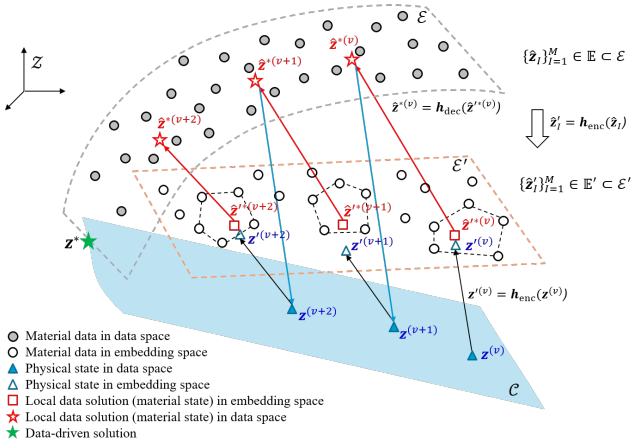

Figure 2: Geometric schematic of the proposed auto-embedding data-driven computational framework. The material data points (the gray-filled circles), $\hat{\mathbf{z}}_I$, in the phase space are related to the material embedding points (the white-filled circles) $\hat{\mathbf{z}}'_I$ via the encoder function. The low-dimensional embedding manifold is represented by the orange dash line.

(10 and 11). The normal components of the Green strain and the associated 2nd-PK stress tenors of all protocols are plotted in Fig. 3(c-d). Considering the symmetry of specimen geometry and loading conditions, a quarter model with symmetric boundary conditions is modelled, as shown in Fig. 3b. The normalized root-mean-square deviation (*NRMSD*) with respect to the maximum experimental stress data S^{true}_{max} is employed to assess the prediction performance of the methods, see Eq. (10),

$$NRMSD = \sqrt{\sum_{i}^{N_{eval}} \frac{(S_i^{AEDD} - S_i^{true})^2}{N_{eval}}} / S^{true}_{max}, \tag{10}$$

where $N_{eval} = 200$ is the number of evaluation points, S_i^{AEDD} and S_i^{true} are the predicted stress and the experimental stress data at the i-th evaluation point, respectively.

It is observed that autoencoders with an embedding dimension $p \leq 2$ could not capture a meaningful embedding representation for the material dataset with two-dimensional strain-stress pairs ($d = 6$). This is consistent with the observation in (He and Chen 2020) that the number of nearest neighbors for local convex reconstruction should be greater than the intrinsic dimensionality of data, which is 2 for the two-dimensional strain-stress material dataset. Hence, the autoencoder with an architecture of "6-4-3-4-6" is employed, where the first and the last values are the input and output dimensions, respectively, and the remaining values denote the number of neurons in hidden layers in sequence. Thus, the embedding dimension equals to 3. A hyperbolic tangent function is adopted as the activation function for all layers of autoencoders, except for the embedding layer and the output layer, where a linear function is employed instead. The regularization parameter β in the loss function

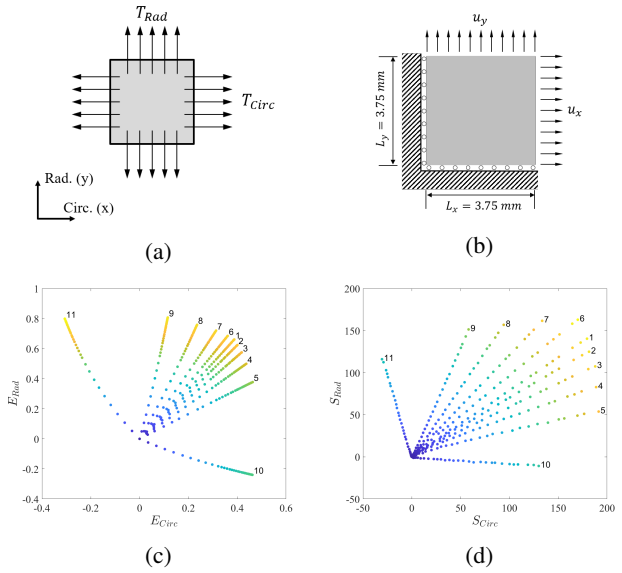

Figure 3: (a) Schematic of a mitral valve posterior leaflet specimen mounted on a biaxial testing system; (b) schematic of the model of biaxial testing in data-driven computation; (c) experimental Green strain data of all protocols; (d) experimental 2nd-PK stress data of all protocols

Eq. (6) is set as 10^{-5}. An adaptive gradient algorithm, Adagrad (Duchi, Hazan, and Singer 2011), is employed. The initial learning rate is 0.1 and the number of training epochs is 2000. The training dataset contains the strain-stress data of protocols 1, 3, 4, 7, and 8, see Fig. 3(c-d), which are standardized to have zero mean and unit variance for accelerated training process.

The autoencoder is trained offline using the open-source Pytorch library (Paszke et al. 2017) and then applied in the AEDD solver during online computation of protocol 5. The data-driven simulation predicts the stress responses of the model (Fig. 3b) under the displacement-controlled loading prescribed by the deformation history of protocol 5 (Fig. 3c). As shown in Fig. 4, AEDD achieves a higher prediction accuracy than the local convexity data-driven (LCDD) computing approach (He and Chen 2020; He et al. 2020), ($NRMSD_{AEDD} = 0.061 < NRMSD_{LCDD} = 0.158$). The results demonstrate better extrapolative generalization ability of AEDD, which is attributed to the underlying low-dimensional global material manifold learned by the autoencoders. Specifically, AEDD performs local neighbor searching and locally convex reconstruction of optimal material state based on geometric distance information in the low-dimensional global embedding space, which contains the underlying manifold structure of the material data and contributes to a higher solution accuracy and better generalization performance. In contrast, LCDD performs local neighbor searching and locally convex reconstruction purely from the existing material data points without any generalization, leading to very limited extrapolative generalization ability.

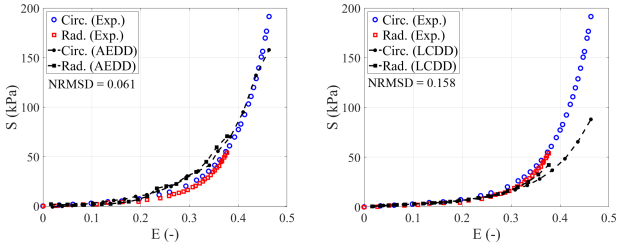

Figure 4: (a) AEDD prediction on Protocol 5; (b) LCDD prediction on Protocol 5. Protocols 1, 3, 4, 7, and 8 are used to train the autoencoder applied in AEDD

References

DeMers, D., and Cottrell, G. W. 1993. Non-linear dimensionality reduction. In *Advances in neural information processing systems*, 580–587.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12(Jul):2121–2159.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.

He, Q., and Chen, J.-S. 2020. A physics-constrained data-driven approach based on locally convex reconstruction for noisy database. *Computer Methods in Applied Mechanics and Engineering* 363:112791.

He, Q.; Laurence, D. W.; Lee, C.-H.; and Chen, J.-S. 2020. Manifold learning based data-driven modeling for soft biological tissues. *Journal of Biomechanics* 110124.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science* 313(5786):504–507.

Ibanez, R.; Abisset-Chavanne, E.; Aguado, J. V.; Gonzalez, D.; Cueto, E.; and Chinesta, F. 2018. A manifold learning approach to data-driven computational elasticity and inelasticity. *Archives of Computational Methods in Engineering* 25(1):47–57.

Jett, S.; Laurence, D.; Kunkel, R.; Babu, A. R.; Kramer, K.; Baumwart, R.; Towner, R.; Wu, Y.; and Lee, C.-H. 2018. An investigation of the anisotropic mechanical properties and anatomical structure of porcine atrioventricular heart valves. *Journal of the mechanical behavior of biomedical materials* 87:155–171.

Kirchdoerfer, T., and Ortiz, M. 2016. Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering* 304:81–101.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.