

RESEARCH

Open Access



# Data-driven non-intrusive reduced order modelling of selective laser melting additive manufacturing process using proper orthogonal decomposition and convolutional autoencoder

Shubham Chaudhry<sup>1</sup>, Azzedine Abdedou<sup>1</sup> and Azzeddine Soulaïmani<sup>1\*</sup>

\*Correspondence:  
Azzeddine Soulaïmani  
Azzeddine.soulaïmani@etsmtl.ca  
<sup>1</sup>Department of Mechanical  
Engineering, École de Technologie  
Supérieure, Montréal, QC  
H3C 1K3, Canada

## Abstract

This study proposes and compares two data-driven, non-intrusive reduced-order models (ROMs) for additive manufacturing (AM) processes: a combined proper orthogonal decomposition-artificial neural network (POD-ANN) and a convolutional autoencoder-multilayer perceptron (CAE-MLP). The POD-ANN model utilizes proper orthogonal decomposition to create a reduced-order model, which is then combined with an artificial neural network to establish a surrogate model linking the snapshot matrix to the input parameters. This approach effectively reduces the dimensionality of the high-fidelity snapshot matrix and constructs a regression framework for accurate predictions. Conversely, the CAE-MLP model employs a 1D convolutional autoencoder to reduce the spatial dimension of a high-fidelity snapshot matrix derived from numerical simulations. The compressed latent space is then projected onto the input variables using a multilayer perceptron (MLP) regression model. This method leverages deep learning techniques to handle the complexity of the data and improve prediction accuracy. The accuracy and efficiency of both models are evaluated through thermo-mechanical analysis of an AM-built part. The comparison of statistical moments from high-fidelity simulation results with ROM predictions reveals a strong correlation. Furthermore, the predictions are validated against experimental results at various locations. While both models demonstrate good agreement with experimental data, the CAE-MLP model outperforms the POD-ANN model in terms of prediction accuracy and performance. The findings highlight the potential of integrating reduced-order modeling techniques with machine learning algorithms to enhance the analysis of complex AM processes. The proposed models offer a robust framework for future research and applications in the field of additive manufacturing, providing high precision and efficiency.

**Keywords** Reduced-order model, Convolutional autoencoder, Additive manufacturing, Deep learning, Proper orthogonal decomposition

## Introduction

Laser Powder Bed Fusion (L-PBF) is a cutting-edge Additive Manufacturing (AM) technology that fabricates 3D parts by fusing successive layers of powder using a high-energy laser beam. L-PBF comprises two primary techniques: Electron Beam Melting (EBM), which uses a high-energy electron beam, and Selective Laser Melting (SLM), which employs a high-power laser to bond powder layers. This process enables the production of complex, robust geometries with minimal material waste, significantly reducing manufacturing costs. Despite these advantages, L-PBF faces challenges such as part distortion, dimensional inaccuracies, and premature build failures. This study focuses specifically on the SLM process, analyzing temperature evolution and strain in the fabricated part to improve understanding of its thermal and mechanical behavior during manufacturing.

In SLM, steep thermal gradients occur between the melt pool (reaching several thousand degrees Celsius) and surrounding solidified regions, which remain near ambient or preheat temperatures. These gradients induce residual stresses that can cause deformation and incompatibility. The combination of extreme temperature differences and a rapidly moving high-power laser requires highly nonlinear, finely meshed numerical models for accurate simulation. Moreover, SLM involves multiple length and time scales, further complicating numerical analyses. Rapid temperature changes in the melt pool occur within milliseconds to seconds, while the layer-by-layer building process may extend over hours or even days, with prolonged cooling periods. Phase transformations and latent heat during the powder-to-liquid-to-solid transitions add to the complexity and computational cost of thermal analysis. As a result, there is an urgent need for advanced numerical methods or alternative approaches to reduce these computational demands.

Data-driven machine learning models offer a promising alternative for studying the SLM process [1, 2]. Ravichander et al. [3] developed a neural network model to predict SLM outputs, training on experimental data and applying the model to generate new datasets. Similarly, Chaudhry et al. [4] introduced an ML framework to construct a surrogate model for SLM optimization. Other studies [5–9] have also explored ML approaches for additive manufacturing. However, as dataset dimensionality grows, these methods face scalability challenges. For instance, in [4], the data matrix for training the deep neural network (DNN) had dimensions of  $360 \times 287$ . Expanding this to  $360 \times 97,650$  by increasing input vector sizes would render the DNN approach impractical due to the explosion in trainable parameters, which can reach millions or even billions, depending on network depth and size.

Recently, reduced-order modeling (ROM) has gained popularity in the computational community as an effective way to reduce computational costs without sacrificing accuracy, especially when dealing with large datasets. ROM enables rapid surrogate models for expensive simulations, making it particularly valuable for optimization tasks, real-time tracking, and online predictions in both industrial applications and fundamental science.

One of the most widely used ROM techniques is Proper Orthogonal Decomposition (POD). Originally introduced by Pearson [10] in 1901, POD has evolved into a robust and efficient tool for ROM analysis across various fields [11]. Recent developments have introduced non-intrusive methods for determining the coefficients of linear POD approximations through data-driven approaches, while preserving the integrity of

the governing equations [12]. These include stochastic frameworks such as POD with Polynomial Chaos Expansion (POD-PCA) and POD combined with Artificial Neural Networks (POD-ANN), which build regression models linking input parameters to the coefficients of the POD basis [13–17]. Zhao et al. [18] applied POD to analyze the thermal behavior of electron beam melting (EBM) systems, using ABAQUS to investigate temperature distributions in a moving energy source. Similarly, studies [19–21] proposed POD-based ROM approaches for linear and nonlinear transient heat transfer problems, demonstrating strong agreement between standard FEM simulations and POD-FEM results.

The essence of POD lies in constructing a high-fidelity snapshot matrix to extract a small set of eigenmodes and the coefficients of a linear basis derived from these modes. Zhao et al. [22] used POD to develop a parameter map and POD bases via a regression tree method, which was later applied to predict outcomes for new sets of input variables. In related studies [14, 15], POD modes were combined with machine learning models, including artificial neural networks (ANNs). This integration of POD and ML has proven highly effective for handling large datasets and complex processes.

Recent advances in additive manufacturing have leveraged Physics-Informed Neural Networks (PINNs) to model thermal-mechanical behavior and melt-pool dynamics. Liao, et al. [23] developed a PINN-based thermal model for SLM with parameter inference, and Li et al. [24] provided one of the first 3D PINN formulations for temperature and flow prediction in SLM. More recently, Sharma et al. [19] introduced thermomechanically augmented PINNs for stress prediction, and Zhu et al. [25] proposed a transfer learning-enhanced PINN (TLEPINN) for rapid meltpool morphology estimation. These studies underscore the growing integration of PINNs in additive manufacturing for real-time, highfidelity modeling and highlight future directions for extending the capabilities of data-driven reduced-order models.

Building on this context, the present study proposes a framework for non-intrusive reduced-order models that combine POD and CAEs in two distinct approaches to reduce the dimensionality of high-fidelity matrices obtained from finite element simulations. These frameworks are entirely data-driven and aim to predict strain fields in SLM-built parts efficiently while preserving accuracy. While POD's linearity is a strength, it also limits its ability to represent complex dynamic systems [26]. It often fails to capture transient or highly localized features due to its reliance on global basis functions, which may not localize well in space or time. The quality of snapshots is another critical factor, as insufficient coverage of significant regimes can lead to inadequate reduced models. Modal truncation, often determined heuristically, introduces a trade-off between computational efficiency and the ability to capture essential dynamics. Furthermore, the orthogonality constraint of POD modes may not align with the actual physics in certain systems, such as turbulent flows, where relevant structures are not orthogonal. Although POD modes are often considered physically interpretable, this clarity can diminish in highly complex systems, making it harder to relate them to meaningful physical phenomena.

To address these limitations, nonlinear manifolds have been explored as alternatives for dimensionality reduction, with several approaches leveraging deep learning techniques [27, 28]. One such method is the autoencoder, which consists of two components: an encoder and a decoder. Jin et al. [29] introduced a convolutional neural network

(CNN)-based autoencoder (CAE) model for detecting anomalies in AM-built parts. CAEs incorporate operations such as convolution, multilayer perceptron, upscaling, and pooling [30, 31], which help reduce the number of trainable parameters. However, due to their complexity, CAEs are prone to overfitting if not properly regularized. Compared to POD, the CAE-MLP model generally demands more training and prediction time; however, this trade-off is justified by its higher accuracy, especially for strongly nonlinear systems. Unlike traditional linear bases, CAEs excel at capturing complex, hierarchical, and spatially localized features. This expressive nonlinear power is essential for accurately modeling turbulent flows, highly transient phenomena, or systems where linear assumptions fail. Moreover, CAEs can generalize effectively across diverse operating conditions, potentially eliminating the need for retraining or recalibration when applied to new scenarios. Although CAEs require higher initial training costs, they enable high-fidelity surrogate predictions that can accelerate convergence in optimization loops and design studies—offsetting their slightly slower prediction times. In related studies [32–34], CAEs have been combined with Long Short-Term Memory (LSTM) networks to develop surrogate modeling frameworks, particularly for time-dependent predictions in AM-built parts.

This study proposes a framework of non-intrusive reduced-order models (ROMs) for the parametric analysis of the SLM building process. The data-driven framework integrates POD and CAEs in two distinct approaches to reduce the dimensionality of high-fidelity matrices obtained from finite element simulations. These solution matrices represent normal directional strains computed using Workbench Additive software [35]. This comparison aims to evaluate how the model architectures and training procedures influence the learned latent modes and the accuracy of reconstructions and predictions. In the first approach, POD bases of the strain matrices are computed, and a multilayer perceptron (MLP) is trained to map input variables to the POD coefficients. This surrogate model is then used to predict strains for new sets of input parameters. In the second approach, dimensionality reduction is performed using a CAE, where the spatial dimension is encoded into a latent space. An MLP maps this latent space to the input variables, while the decoder reconstructs the original spatial dimension from the latent representation. Together, the trained MLP and decoder reconstruct strain fields for new process parameters.

The proposed framework is applied to the stochastic analysis of a benchmark SLM-built part, AMB2018-1, produced by the National Institute of Standards and Technology (NIST) [36]. This work introduces new capabilities for reduced-order modeling of complex physical systems.

The paper is organized as follows: Sect. [Mathematical modelling](#) presents the mathematical background of POD and CAE. Section [Results and Discussion](#) evaluates the performance of the proposed framework through a comparison with benchmark results. Section [Conclusion](#) provides concluding remarks and directions for future research.

## **Mathematical modelling**

### **Proper orthogonal decomposition (POD)**

POD was originally developed in the study of turbulent flow fields to decompose random vector fields into deterministic functions that capture the fluctuating kinetic energy of the flow [37, 38]. The underlying idea was that a finite set of deterministic functions,

known as POD modes, could effectively describe the flow structure. Since its inception, POD has been widely adopted across various scientific and engineering domains.

POD exhibits two key properties: optimality and orthogonality. Optimality ensures that the decomposition is the most efficient, with the leading modes capturing the maximum possible energy compared to any other linear decomposition of the data. Orthogonality ensures that the time series of the modal coefficients are linearly uncorrelated, a critical feature for constructing stable and accurate reduced-order models.

Let us suppose  $Y = [y_1, y_2, \dots, y_n]$  is a real-valued  $m \times n$  matrix whose rank is  $d \leq \min\{m, n\}$  with columns  $y_j \in R^m$ ,  $1 \leq j \leq n$ . POD is combined with singular value decomposition (SVD) to obtain the reduced order model and a low-rank approximation that is easy to compute [38]. The SVD assures that there are real numbers  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_d > 0$  and orthogonal matrices  $\Psi \in R^{m \times m}$ , with columns  $\{\psi_i\}_{i=1}^m$ , and  $\Phi \in R^{n \times n}$ , with columns  $\{\varphi_j\}_{j=1}^n$ , such that

$$\Psi^T Y \Phi = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \quad (1)$$

Here,  $D = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_d) \in R^{d \times d}$ . The zero blocks in Eq. (1) have the appropriate dimensions, and  $T$  represents the matrix transpose. In addition,  $\{\psi_i\}_{i=1}^d$  and  $\{\varphi_j\}_{j=1}^d$  satisfy the properties

$$Y \varphi_i = \sigma_i \psi_i \text{ and } Y^T \psi_i = \sigma_i \varphi_i \text{ where } i = 1, 2, \dots, d \quad (2)$$

which are eigenvectors of  $YY^T$  and  $Y^TY$ , respectively, with the eigenvalues  $\lambda_i = \sigma_i^2 > 0$ ,  $i = 1, 2, \dots, d$ . Also,  $\{\psi_i\}_{i=d+1}^m$  and  $\{\varphi_j\}_{j=d+1}^n$  are eigenvectors with the eigenvalue 0 of  $YY^T$  and  $Y^TY$  (if  $d < m$  and  $d < n$ ).

From Eq. (1) we can write

$$Y = \Psi \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \Phi^T \quad (3)$$

For a finite number of initial  $L$  modes, the truncation criteria are imposed on the singular values as shown below:

$$\frac{\sum_{l=L+1}^r \sigma_l^2}{\sum_{l=1}^r \sigma_l^2} \leq \delta \quad (4)$$

where  $\delta$  is a small parameter. So, every mode vector  $V_j$  may be calculated from the  $j$ th column of  $\varphi$  as

$$V_j = \frac{1}{\sigma_j} Y \Phi_j$$

.

Thus, the POD mode matrix can be constructed as below

$$V = [V_1 | \dots | V_j | \dots | V_L] \in R^{m \times L} \quad (5)$$

Once the POD modes are obtained, they are then used to calculate the projection coefficients ( $v$ ) for the snapshot matrix as shown:

$$v = V^T Y$$

Similarly, the POD bases and the projection coefficients can be used to find the approximation matrix of  $Y$ :

$$Y^* = V V^T Y = V v$$

The quality of the compression/expansion process can be captured by the relative projection error shown in the equation:

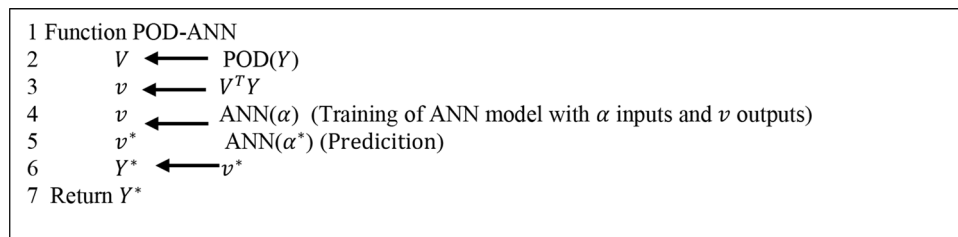
$$RE_{POD} = \sum_{j=1}^n \frac{\| (Y)_j - (Y^*)_j \|_2}{\| (Y)_j \|_2} \quad (6)$$

where  $j$  represents the  $j^{\text{th}}$  column of the targeted matrix and  $\| \cdot \|_2$  is the  $L^2$ - norm.

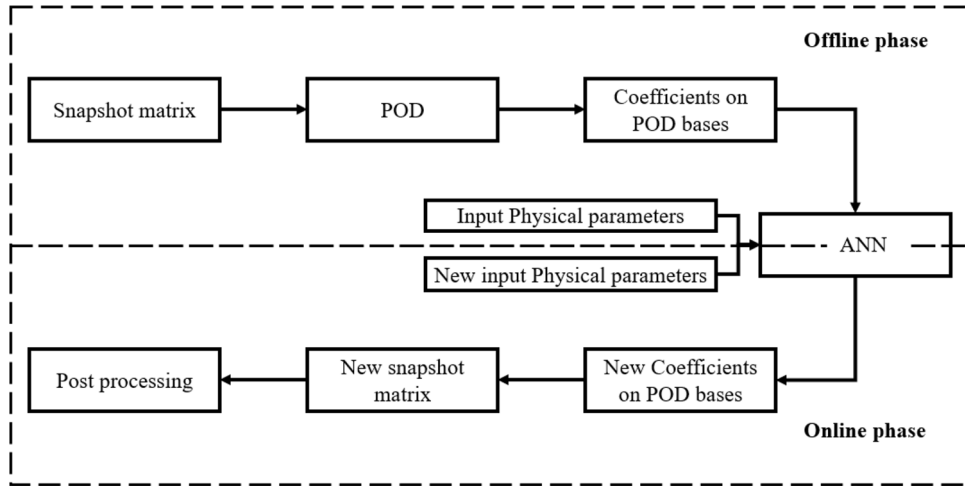
#### Proper orthogonal decomposition and artificial neural network (POD-ANN)

A schematic of the POD-ANN method is presented in Fig. 1. In the first step, the POD algorithm extracts the dominant modes from the snapshot matrix. These modes are then used to compute the projection coefficients  $v$ , providing a low-dimensional representation of the original data. Next, an artificial neural network (ANN) is trained to map the input variables to these coefficients  $v$ . The POD-ANN framework consists of two stages: the offline phase, where the ANN is trained using input parameters  $\alpha$  and corresponding projection coefficients  $v$ . Here, 80% of the data is allocated for training and 20% for testing; and the online phase, where the trained ANN predicts new projection coefficients  $v^*$  for a given set of physical parameters  $\alpha^*$ . These predicted coefficients are then reconstructed into the original high-dimensional space using the process described in Algorithm 1.

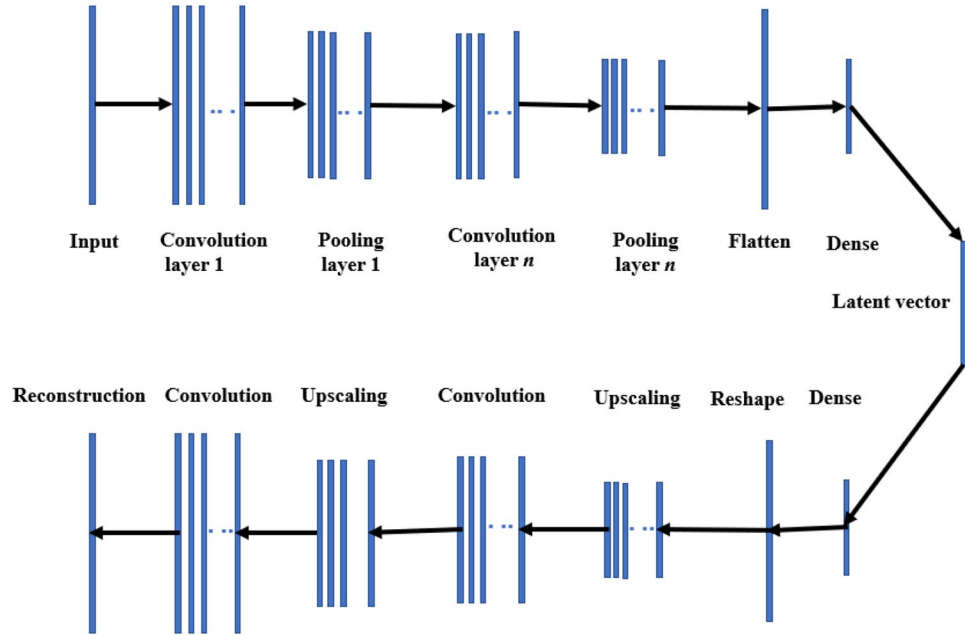
An autoencoder is a type of deep neural network designed for unsupervised feature extraction and dimensionality reduction. Its architecture is well-suited for modal decomposition due to the inclusion of nonlinear activation functions. However, standard autoencoders can struggle to decompose input fields with multiscale coherent features. This limitation can be mitigated by incorporating convolutional layers, which are adept at processing high-dimensional spatial data. Convolutional autoencoders (CAEs) have gained prominence in image recognition and are increasingly applied in nonlinear reduced-order modeling. Unlike classical autoencoders with fully connected layers, CAEs employ convolutional layers to efficiently handle high-dimensional inputs [31]. A CAE consists of two main components: the encoder, which compresses the input matrix into a latent space using convolution, pooling, and dense layers; and the decoder, which reconstructs the original input dimensions from the latent space through upscaling, convolution, and dense layers. The architecture of a CAE is illustrated in Fig. 2. In this



**Algorithm 1** Flowchart of the POD-ANN method



**Fig. 1** Sketch of the POD-ANN process



**Fig. 2** Pictorial representation of a 1D convolution autoencoder architecture

study, we adopt a similar CAE structure to encode and decode the spatial dimensions of the input snapshot matrix.

In the convolutional layers, each unit in a feature map is connected to a localized region of the preceding layer through a kernel and activation function. This localized connection allows the network to extract dominant features from the input efficiently. The operation of a 1D convolutional layer can be mathematically expressed as [39]:

$$h_i^l = \sigma \left( H^{l-1} * f_i^l + b_i^l \right) \quad (7)$$

in which the  $*$  denotes the 1D convolution operator,  $h_i^l \in R^{D_l \times 1}$  represents the  $i^{th}$  feature of the  $l^{th}$ ,  $\sigma$  is the nonlinear activation function,  $H^{l-1} = [h_1^{l-1}, h_2^{l-1}, \dots, h_{N_{f_{l-1}}}^{l-1}]$  represents the convolution layer  $l-1$ ,  $b_i^l$  gives bias value with  $i \in N_{f_l}$ ,  $f_i^l$  represents



kernel for layer  $l$  and  $l \in (1, n)$ . The depth of the convolution layer is represented by the total number of layers. After each layer, the pooling layers are inserted to decrease the dimension of the features by an amount that is defined by the kernel size of the pooling layer.

As presented in the previous section, the snapshot matrix consists of a set of  $n$  high-fidelity solutions obtained from the numerical simulation  $\{y(\alpha_s) \in R^m, s = 1, \dots, n\}$ . In this solver,  $\alpha_s$  is the  $s^{th}$  value of the random variable  $\alpha$  in its data sample with size  $n$ , which follows a probability density function  $\phi(\alpha)$ . All these vectors' solutions are combined and form a global snapshot matrix:

$$Y = [y_1 \dots y_s \dots y_n] \in R^{m \times n} \quad (8)$$

where  $m$  is the total number of computational nodes in the spatial domain. The global snapshot matrix created above is divided into two sets, one with 80% and the other with 20% of the data for training and testing, respectively.

The CAE framework is divided into three parts: (1) spatial compression, (2) a regression-based multilayer perceptron (MLP), and (3) online surrogate predictions. Spatial compression reduces the dimension of the input data matrix from  $n$  to  $L$  along the spatial dimension where  $L$  represents the latent space dimension. The snapshot matrix  $Y$  is reshaped using the space encoder ( $F_{enc}$ ) part of the CAE along the spatial dimension, as given:

$$V_{xL} = F_{enc}(Y) \in R^{m \times L} \quad (9)$$

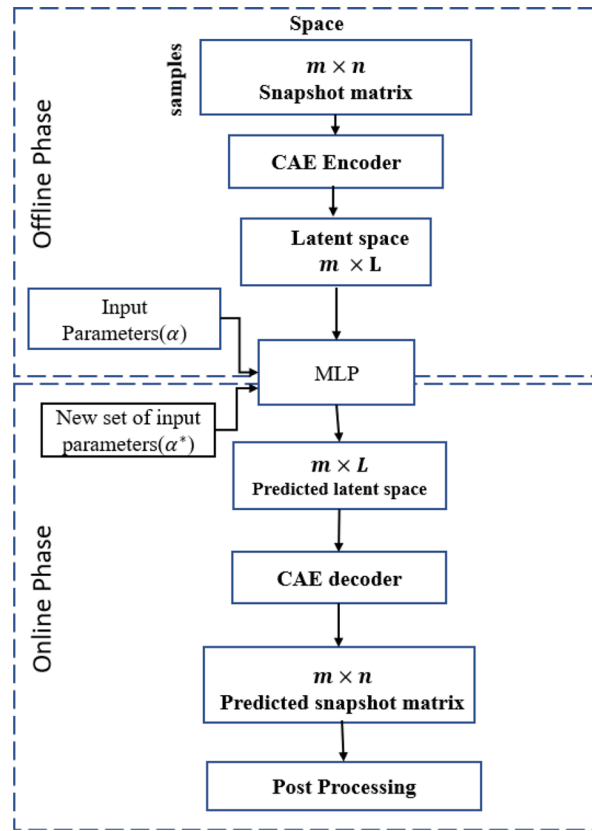
where  $V_{xL}$  represents the snapshot matrix with reduced dimension. The detailed architectures of each autoencoder used for the test set and the benchmark problem are provided in Appendices Tables A1 and A3. Once the latent space is constructed, the next stage of the CAE framework involves implementing an MLP within this latent space. The MLP model is composed of multiple fully connected layers. Detailed configurations of the MLP applied in each case are given in Appendices Tables A2 and A4. The compression of the spatial dimension using the CAE and the application of the MLP constitute the offline phase of the framework, as illustrated in Fig. 3. It is worth noting that this framework is developed using the open-source package TensorFlow [40] and optimized with the Adam optimizer employing its default parameters. To accelerate optimization and improve convergence during training, the input snapshots are normalized as follows:

$$\widetilde{u}_{si} = \frac{Y_{si} - \min(Y_s)}{\max(Y_s) - \min(Y_s)} \quad s = 1 \dots n, i = 1 \dots m \quad (10)$$

where  $\widetilde{u}_{si}$  is the normalized output for the  $s^{th}$  input parameter and for the  $i^{th}$  mesh node.

The final step involves the online surrogate prediction for a new dataset. A new set of input variables is generated using the Latin Hypercube Sampling (LHS) algorithm [41]. For each entry in the new dataset, a spatial latent vector ( $V_{xL}^*$ ) is predicted using the trained MLP regression model. This predicted latent space is then decoded back to the original dimensional space using the spatial decoder function ( $F_{dec}$ ), following the relation:





**Fig. 3** Flow chart of the CAE with its online and offline phases

1	Function CAE-MLP	
2	$V_{xL}$	$\leftarrow F_{x_{enc}}(Y)$ (Encoding snapshot matrix to latent space)
3	$V_{xL}$	$\leftarrow \text{MLP}(\alpha)$ (Training of MLP model with $\alpha$ inputs and $V_{xL}$ outputs)
4	$V_{xL}^*$	$\leftarrow \text{MLP}(\alpha^*)$ (Prediction with new input parameters)
5	$Y^*$	$\leftarrow F_{x_{dec}}(V_{xL}^*)$ (decoding latent space to original diamanesion)
6	Return $Y^*$	

**Algorithm 2** Flowchart of CAE-MLP method

$$Y^* = F_{x_{dec}}(V_{xL}^*). \quad (11)$$

The overall workflow of this framework is illustrated in Fig. 3 and detailed in Algorithm 2.

## Results and discussion

This section evaluates the POD-ANN and autoencoder models on two test cases: a 2D heat transfer problem and a 3D additive manufacturing scenario. The results obtained from these reduced-order models are compared with experimental data to demonstrate their efficiency and accuracy.

### 2D heat transfer test case

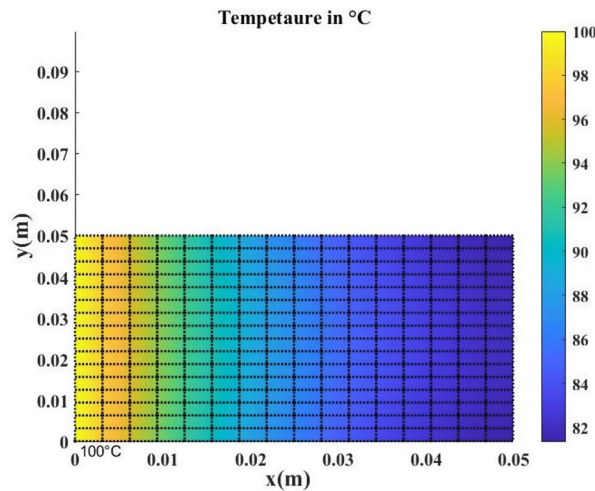
This preliminary test case investigates steady-state heat transfer through a plate of thickness  $e$  in a steady state. It serves to validate the implementation of the proposed

algorithms. The governing equation and associated boundary conditions are defined by the linear heat equation:

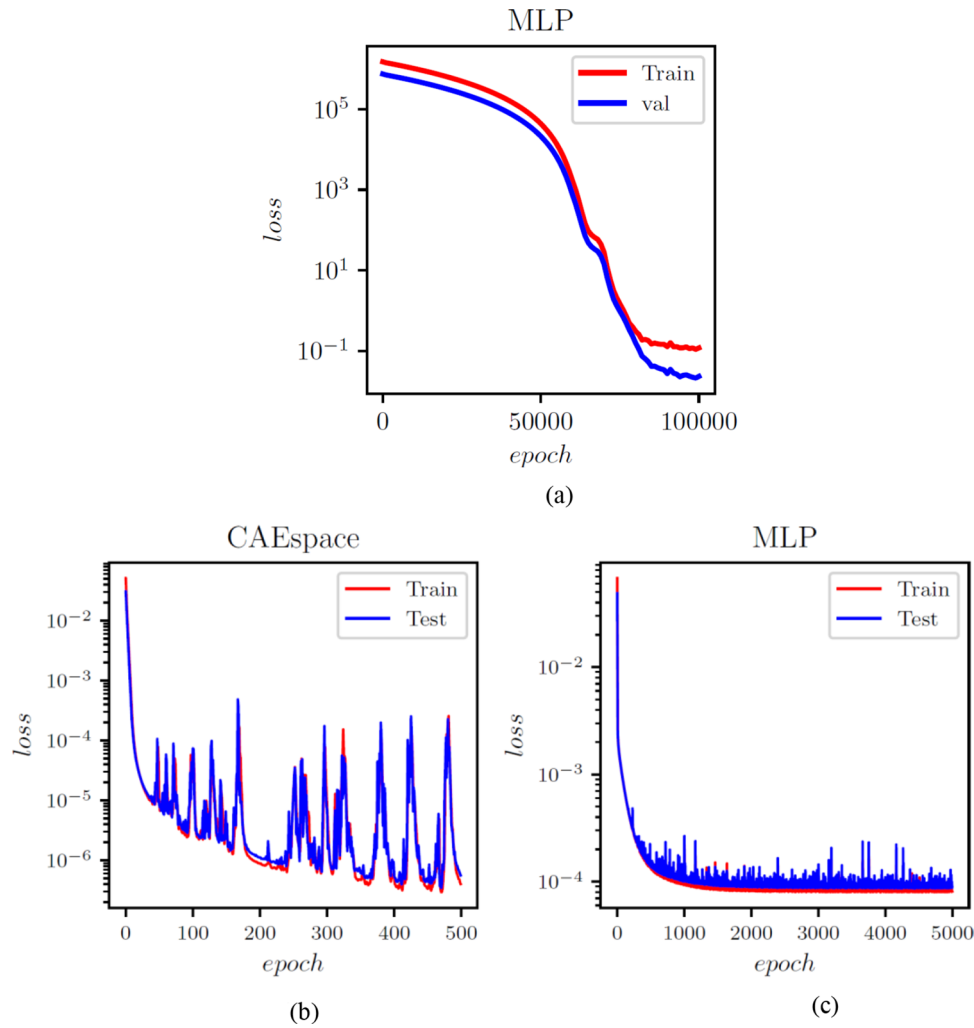
$$\text{div}(\vec{q}) = \frac{2h}{e}(T_{ex} - T) \quad (12)$$

where  $\vec{q}$  is the conduction flux given by Fourier's law  $\vec{q} = -K \nabla T$  ( $K$  is the conductivity,  $h$  is the coefficient of convection and  $100^\circ\text{C}$  represents the given temperature on one side. The boundary condition at the edges is  $\vec{q} \cdot \vec{n} = h(T - T_{ex})$ . The temperature is  $22^\circ\text{C}$  at the base ( $x=0$ ). In this test case, the coefficients of convection and conductivity are constant and considered input parameters, while the temperature over the entire domain is the output. A total of 300 samples were generated for the input parameters using the Lattice Hypercube Sampling (LHS) algorithm, with samples uniformly collected in the intervals  $[51, 61]$  for  $K$  and  $[42.6, 57.5]$  for  $h$ . The heat equation was solved for each input set using an in-house finite element method code. The mesh consists of quadrilateral elements with a total of 289 nodes, as shown in Fig. 4. The output snapshot matrix was used to train the POD-ANN and CAE-MLP models. The dataset of 300 samples was divided into 80% for training and 20% for testing. After training, the predictions from both models with a new set of input parameters were compared with the original snapshot matrix. A set of 5000 samples was generated using the LHS method, and the surrogate models (POD-ANN and CAE) were run to obtain new predictions. A statistical analysis of the outputs was then performed.

The POD-ANN model consists of three hidden layers, each with 50 neurons. The model determined that three POD modes were sufficient to accurately predict the outputs of the heat transfer problem. The CAE-MLP model reduces the spatial dimension from 288 to a latent space dimension of 5. More details on the encoder and decoder structure are provided in Table A1. This process results in a total of 140,587 trainable parameters. The constructed spatial latent space is then mapped to the input variables using a multilayer perceptron (MLP) with 34,053 trainable parameters; the detailed structure is presented in Table A2. The MLP and CAE are trained for 5000 and 500 epochs, respectively, and the loss convergence graphs are shown in Fig. 5. The comparison of the CAE-MLP and POD-ANN models with the snapshot matrix is performed

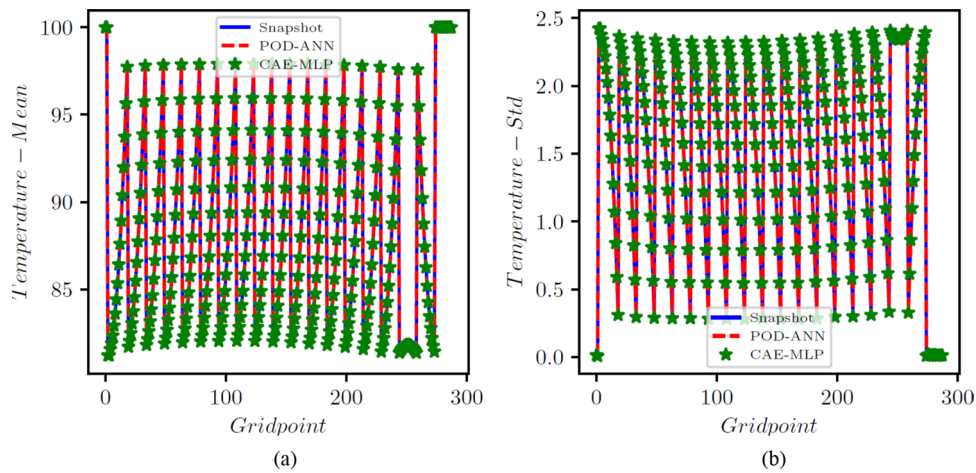


**Fig. 4** Pictorial representation of the mesh structure and temperature distribution throughout the plate



**Fig. 5** Loss function evolution with epochs for the POD-ANN (a), CAE (b) and the MLP (c) networks

using the standard deviation (std) and the mean of the predicted temperature distribution across the entire domain. The outputs are compared in Fig. 6, which shows a good match between the temperature profiles predicted by the POD-ANN and CAE-MLP models and the original snapshot matrix. The relative L2 error norm between the mean of the predicted temperature profile for the 300-snapshot matrix and each of the 5000 realizations of the POD-ANN and CAE-MLP models are  $2.88\text{e-}07$  and  $5.76\text{e-}08$ , respectively. It is important to note that the 5000 realizations are new input parameters, not those used in the training phase. Figure 6(a) shows the mean temperature variation between 100 and 800 °C across the entire domain, while Fig. 6(b) indicates the variation of the temperature profile for each node in the set for a given number of input parameters. Both models required similar times to train and predict the outputs; however, the relative L2 error norm for the CAE-MLP is lower than that of the POD-ANN. This difference indicates that the CAE-MLP predictions are more accurate than those of the POD-ANN. Table 1 provides more details about each model.



**Fig. 6** Comparison of the mean (a) and std (b) of the temperature results between the original snapshot matrix and the temperature predictions of 5000 realizations from POD-ANN and the CAE-MLP networks

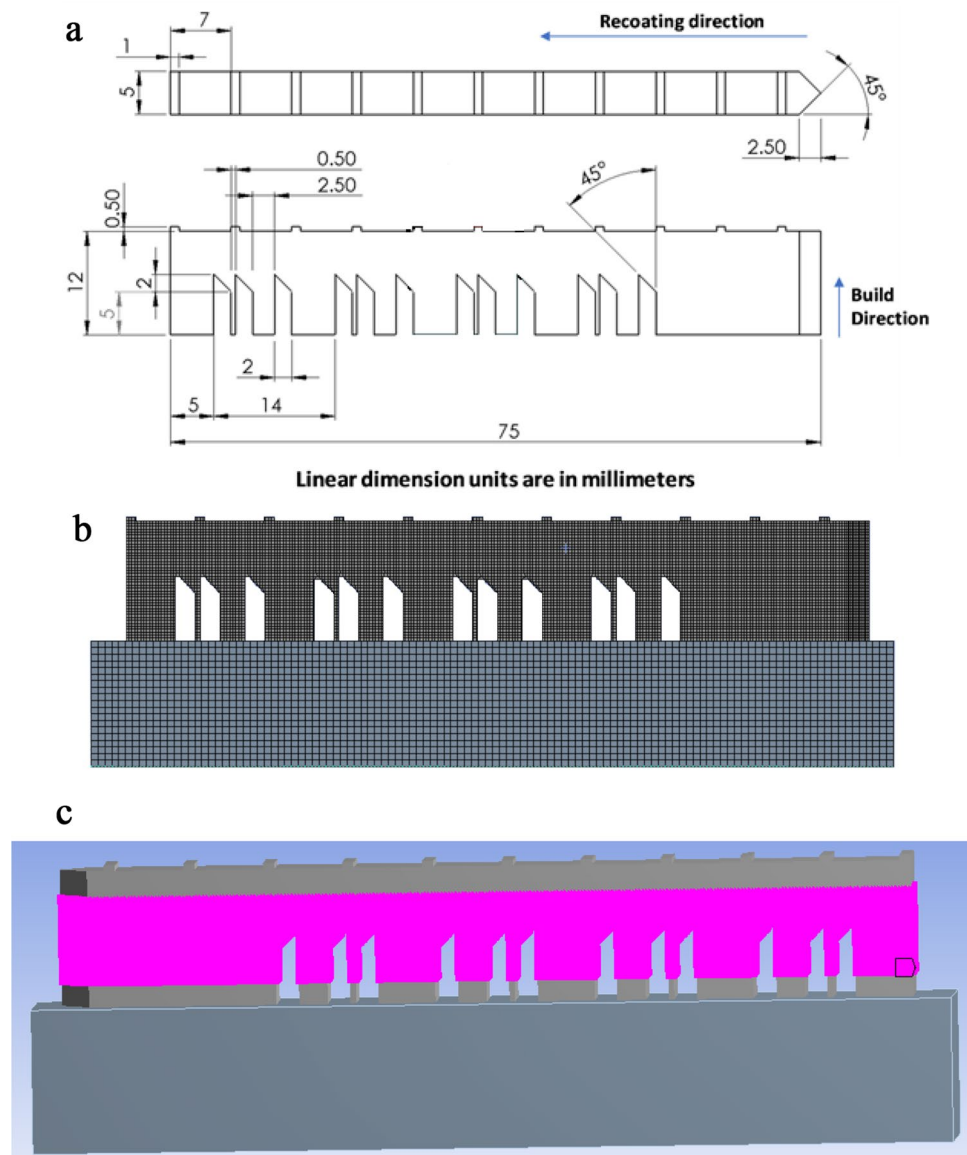
**Table 1** Comparison between the POD-ANN and the CAE-MLP

	POD mode/ Latent space dimension	Training time (sec)	Prediction Time (sec)	Relative L2 error norm (mean)	Relative L2 error norm (std)
POD-ANN	3	422.37	4.15	2.88e-07	9.02e-05
CAE-MLP	5	444.743	5.89	5.76e-08	3.05e-05

#### Additive manufacturing benchmark test

In 2018, the National Institute of Standards and Technology (NIST) released several standard benchmark tests for different AM processes to help the community verify their numerical simulations. The AMB2018-01 test is one such case, designed for the selective laser melting process using the material IN625. AMB2018-01 is a bridge structure with dimensions of  $75 \times 5 \times 12.5$  mm, built over an  $81 \times 12.7 \times 11$  mm substrate, as shown in Fig. 7a and b. The case study aimed to provide reliable data on the residual stresses, strains, and deflections in the built part. The residual stresses and strains were measured using neutron diffraction and x-ray methods, while the deflection was measured after partially cutting the bridge from the base plate. More details on the experimental process and geometry can be found on the NIST website [42]. The additive manufacturing numerical simulation is constructed using the Workbench Additive software. The validation of the workbench model was conducted in a previous publication [4]; for details on the Workbench Additive model, readers are referred to [35]. The simulation took 1 h on 32 CPUs with an Intel E5-2683 v4 processor. The layer thickness, hatch spacing, laser speed, Poisson's coefficient, and Young's modulus were considered random input parameters, while the strain across the entire domain was the output variable. The output strains were calculated over 97,360 nodes in the three-dimensional mesh, as shown in Fig. 7c. To eliminate edge or boundary errors, nodes near the boundary were excluded. This simplification reduces the number of nodes, minimizing computation costs.

The five input parameters were selected within the bounds listed in Table 2 to create a dataset of 360 samples randomly chosen using the LHS sampling method. Young's modulus and Poisson's ratio were scaled using variables  $c_1$  and  $c_2$ , respectively, to adjust their values. Normal directional strains over the 97,360 nodes were calculated for each input



**Fig. 7** **a** 2D representation of the AMB2018-01 bridge geometry from a plan and an elevation view. **b** Mesh of the bridge and the substrate using workbench additive software. **c** Representation of nodes considered to extract the strain results

**Table 2** Upper and lower bounds for input parameters

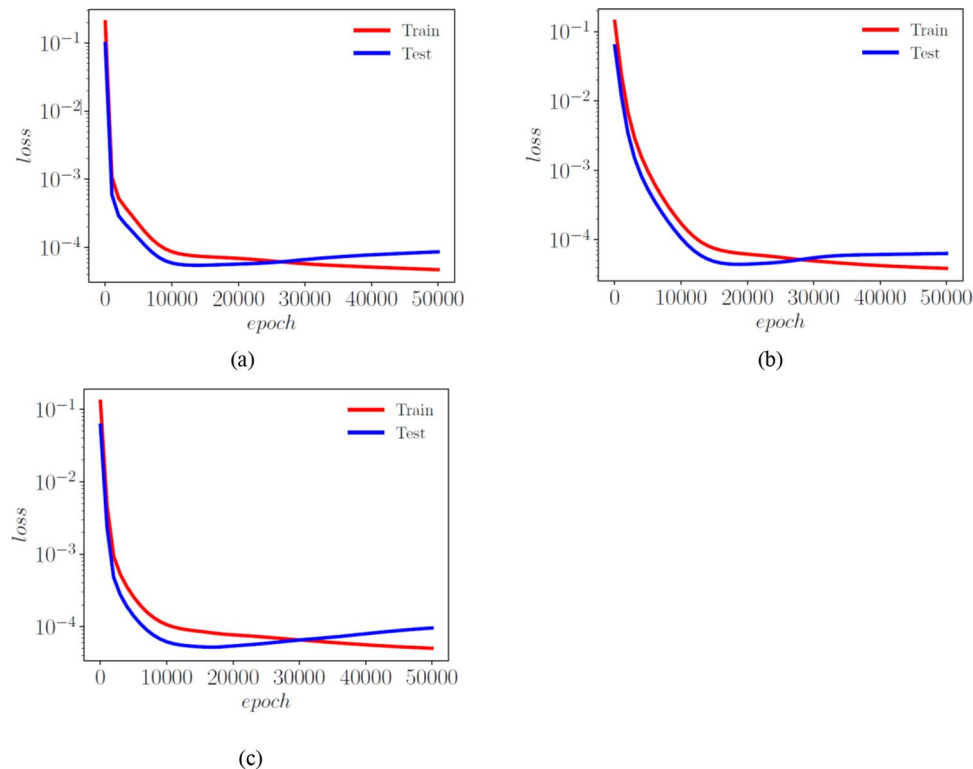
Input Parameters	Lower Bound	Upper Bound
laser speed	680 mm/s	920
layer thickness	0.085 mm	0.115
hatch spacing	0.017 mm	0.023
$c_1Y$	0.85	1.15
$c_2V$	0.85	1.15

sample, and the solutions from each sample were combined to construct a high-fidelity snapshot matrix, which was later used to train the POD-ANN and CAE-MLP models.

A total of three snapshot matrices, each with dimensions of  $360 \times 97,360$ , were generated for the x, y, and z normal strains. The POD-ANN and CAE-MLP models were

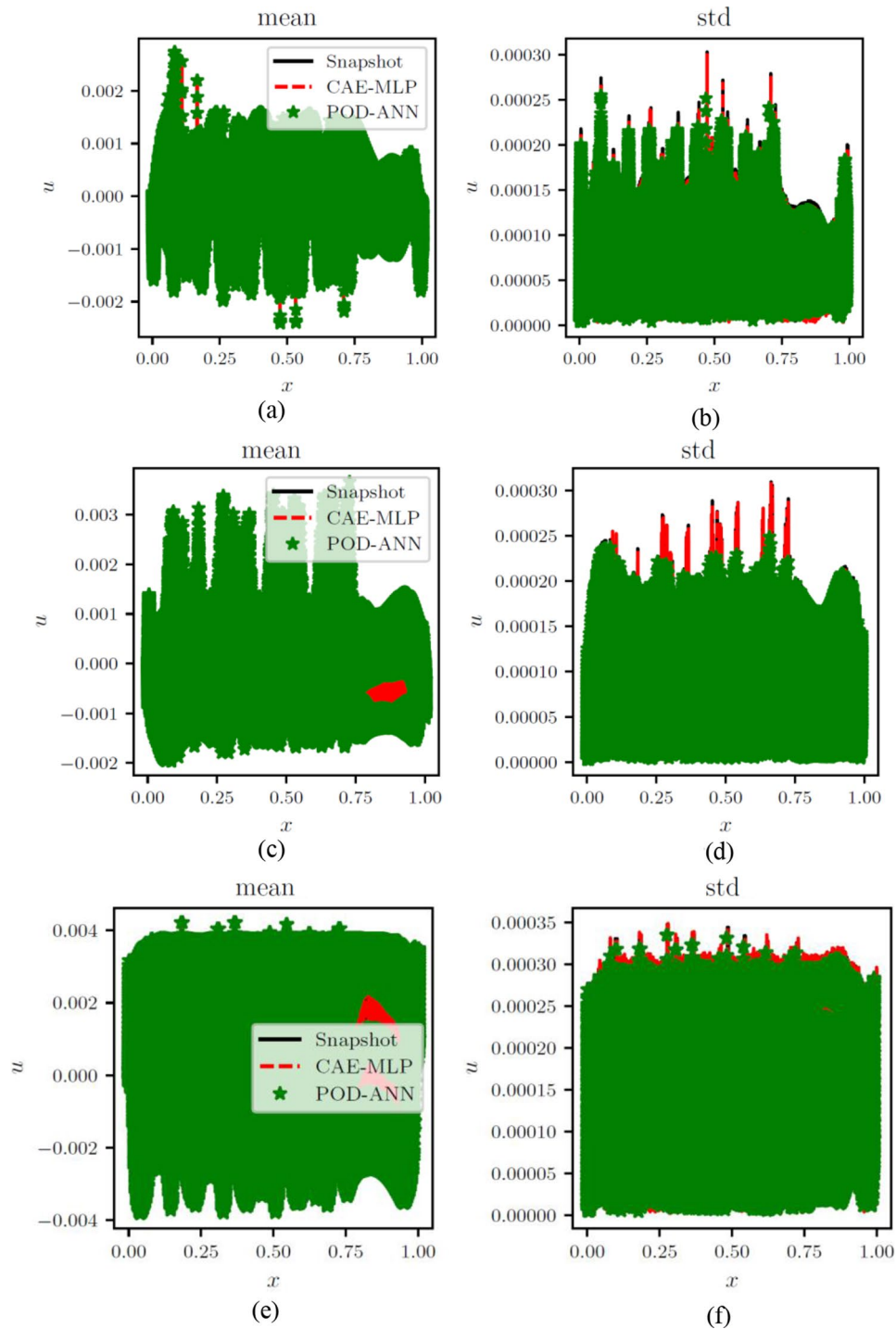
trained separately for each matrix. The performance of both models was evaluated by comparing the statistical moments with the original snapshot matrix. Figure 8 shows the training and testing of the POD-ANN models for all three normal strains, while Fig. 9 presents comparisons between the POD-ANN predictions and the snapshot matrix for the entire domain and different directional strains. The POD-ANN uses the Adam optimizer and the ReLU activation function to achieve better results during the learning and prediction phases. The number of POD modes for each normal strain are calculated using the truncation error  $\delta$ , which is  $10^{-5}$ ,  $10^{-6}$ , and  $10^{-7}$  for the x, y and z directional strains, respectively. It was determined that 19, 42, and 66 POD modes are sufficient to construct a surrogate model for the given snapshot matrix, as evidenced by the error graph in Fig. 8. Figure 9 demonstrates that the mean and standard deviation results correlate well with the original snapshot matrix. The strain graphs indicate that the body has undergone compression in certain areas and expansion in others. The comparison between the mean and standard deviation results suggests that the predictions are accurate. This model can be utilized to calculate the strain at any given point in the geometry and for unseen input parameters within the training intervals. The results along a random cross-section of the bridge are provided in Fig. 10. Additional details related to the POD-ANN structure and effects are presented in Table 3.

Similar to the heat transfer test case, the spatial autoencoder structure consists of 1D convolution layers with 10 and 15 channels, along with max pooling layers and a non-linear activation function (ReLU). These components help reduce the spatial dimension from 97,360 to 19, 40, and 60 for the x, y, and z directional strains, respectively. The reduced dimensions represent the spatial latent space and are directly connected to the input parameters via an MLP model. Figure 3 illustrates the working flowchart



**Fig. 8** Error graphs for the POD-ANN for the x, y and z strains represented by (a), (b) and (c), respectively

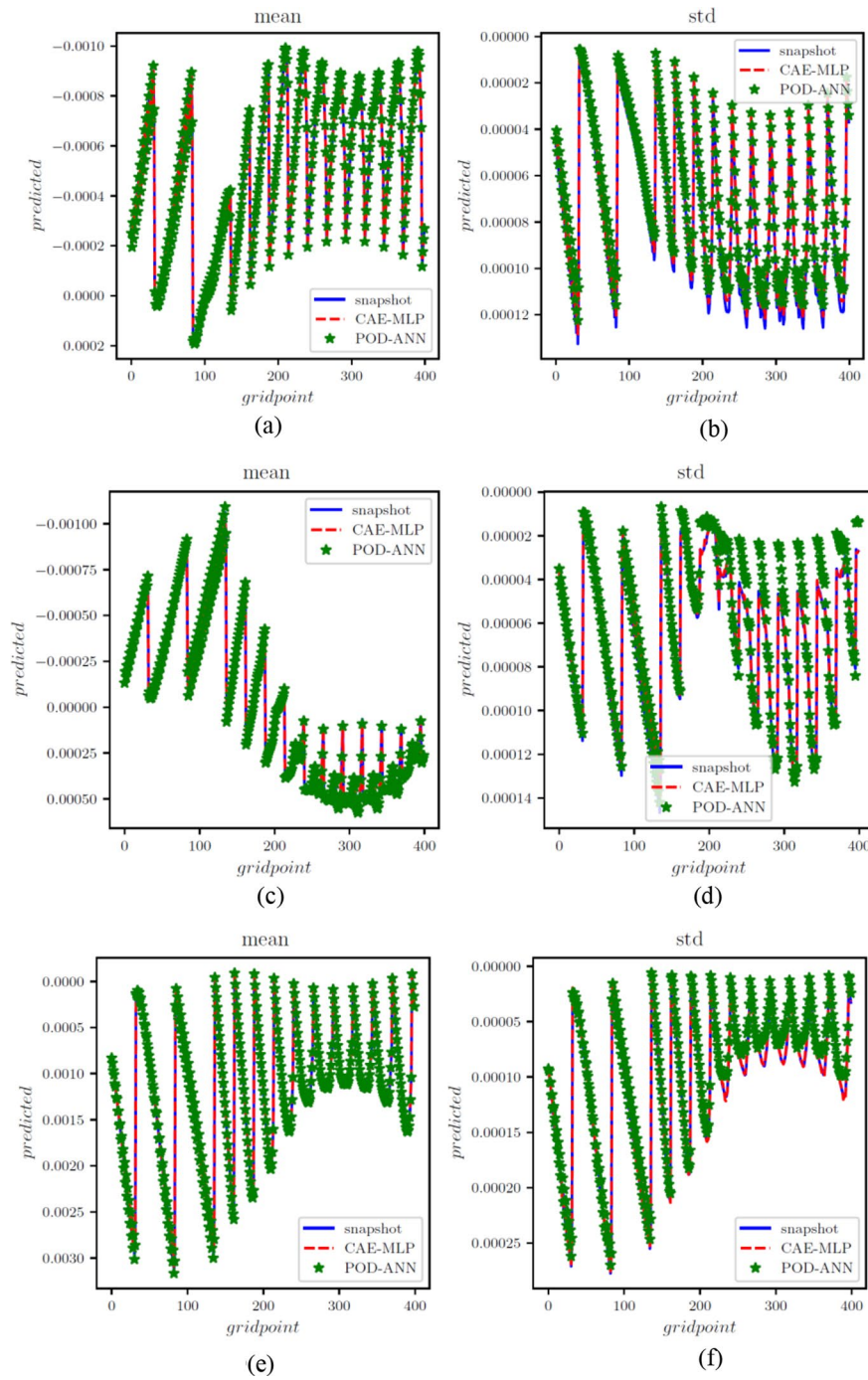




**Fig. 9** Comparison of POD-ANN and CAE-MLP with the snapshot matrices of the  $x$ ,  $y$  and  $z$  strain represented in (a-b), (c-d) and (e-f), respectively, for the whole domain

of the CAE-MLP model, while the detailed architecture of the CAE-spatial and MLP is presented in Appendix Table A3 A and A4 A. During the training process, the CAE is trained for 500 epochs, followed by the MLP for 1,000 epochs. The convergence history of the CAE and MLP for the  $x$ ,  $y$ , and  $z$  strains is shown in Figures B1-B3 in the appendix. The CAE-MLP model is trained on the original snapshot matrices for  $x$ ,  $y$ , and





**Fig. 10** Comparison of POD-ANN and CAE-MLP with the snapshot matrices of the x, y and z strain represented in (a-b), (c-d) and (e-f), respectively, for a cross section

**Table 3** POD\_ANN structure

	POD mode	Hidden layers of ANN	Number of neurons in each layer	Training time (Sec)	Prediction time (Sec)	Relative L2 error mean	Relative L2 error std
X_strain	19	4	100	413.024	40.96	0.0003725	0.0148
Y_strain	42	5	30	226.7586	39.6269	0.0000698	0.00871
Z_strain	66	5	50	290.95	37.41	1.898 e-05	0.0040

$z$  separately, with the calculated outputs presented in terms of standard deviation and mean. Similar to the POD-ANN process, the trained CAE-MLP model is used to predict outputs for a new dataset of 5,000 samples, and the variations in the statistical moments are compared with the original snapshot matrices for each normal strain. Figures 9 and 10 illustrate the variations in the CAE-MLP predictions for the entire domain and a random cross-section of AMB2018-01, respectively, to better visualize the results. The results from the CAE-MLP are also compared with those of the POD-ANN in Figs. 9 and 10 and listed in Table 4. The complete architecture of the CAE-MLP model for different strains is shown in Appendix Table 3 A and 4 A. Both models are trained with 80% of the initial snapshot matrix, while the remaining 20% is used to test the models. Compared to the POD-ANN, the CAE-MLP model takes more time to train and predict, but the CAE-MLP predictions are more accurate. The L2 norm between the statistical moments of CAE-MLP predictions and the snapshot matrix is lower than that of the POD-ANN predictions for all strains. Table 4 lists the results of both models for easy comparison.

#### Comparison with experimental results

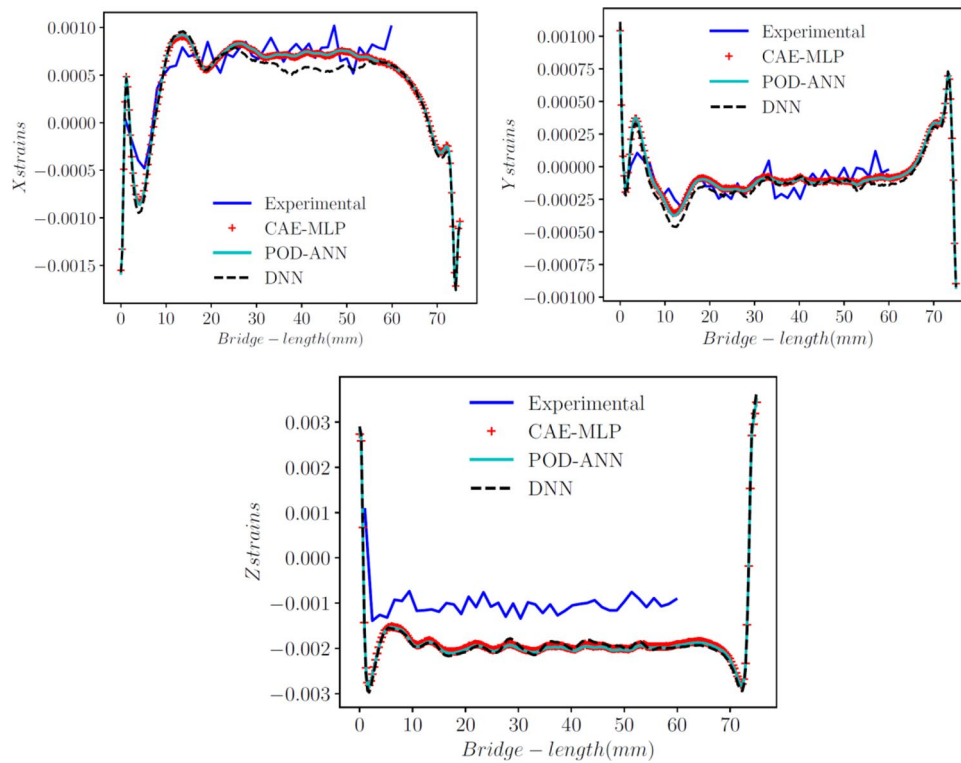
This section compares both models and the experimental outputs. The means of the 5,000 sample outputs from both models are compared with the experimental data at location  $z = 9.536$ , as shown in Fig. 11. These outcomes are also compared with the outputs generated from the DNN model in [4] over the same cross-section. Chaudhry et al. [4] constructed a surrogate model using the DNN model for the normal strains at the experimental location points.

$z = 9.536$  with only 251 nodes. Comparing the three models indicates that the reduced-order models constructed over the whole geometry provide better results than the DNN model built with just a cross-section. The new approaches improve the accuracy of the results by 60%-80%. The CAE-MLP offered significantly better results than the DNN and even improved on the POD-ANN, providing better results for all three normal strains. The relative L2 norm values for the models and the experimental data are presented in Table 5 for each directional strain.

Additionally, the prediction results for normal directions with the POD-ANN and CAE-MLP are compared with the experimental results at three different locations. Figure 12 shows the good correlation between the experimental outputs and the predictions at  $z = 8.25$  mm, 8.75 mm, and 9.25 mm. A total of 40 experimental data points for each location were considered and compared up to a length of 60 mm of the bridge. These results demonstrate the ability of the proposed technique to accurately predict strain values across the entire domain.

**Table 4** Comparison of POD-ANN and CAE-MLP results

	Modes	Relative L2 error for mean	Relative L2 error for std
POD-ANN x strain	19	0.00037	0.0149
CAE-MLP x strain	19	0.0001	0.0038
POD-ANN y strain	44	0.0006	0.0088
CAE-MLP y strain	40	0.0001	0.0014
POD-ANN z strain	66	0.000017	0.0039
CAE-MLP z strain	60	0.0002	0.00049



**Fig. 11** Comparison of the POD-ANN and CAE-MLP models with the experimental results for normal strains in the x, y, and z directions

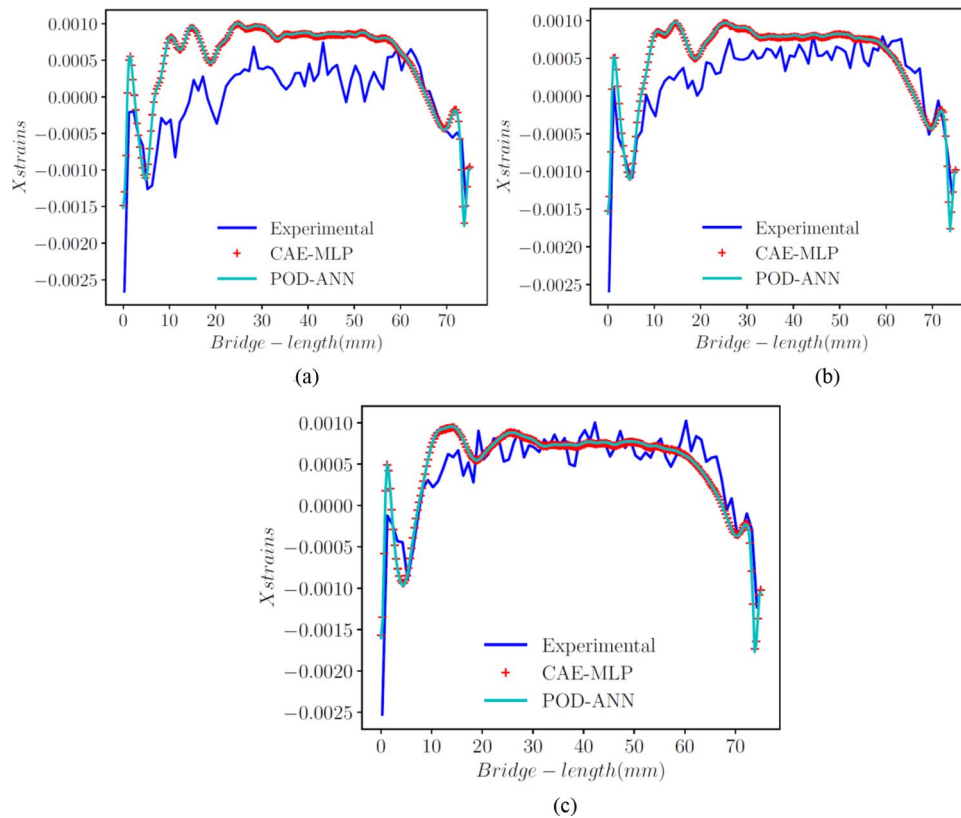
**Table 5** Comparison of relative L2 error norm values of 3 ML approaches with the experimental results

	POD-ANN	CAE-MLP	DNN
X strain	0.040	0.039	0.271
Y strain	0.234	0.212	0.658
Z strain	0.789	0.74	0.873

## Conclusion

This study introduced non-intrusive reduced-order models—POD-ANN and CAE-MLP—to analyze the selective laser melting (SLM) process. By leveraging proper orthogonal decomposition and deep convolutional autoencoders, the proposed frameworks demonstrated both efficiency and effectiveness in constructing data-driven surrogate models that combine techniques from reduced-order modeling and machine learning.

Both approaches operate in two stages: an offline phase and an online phase. During the offline phase, a high-fidelity snapshot matrix ( $97,360 \times 360$ ), representing normal strains computed using Workbench Additive software, is used. In this stage, the POD-ANN employs proper orthogonal decomposition to reduce the dimensionality of the snapshot matrix, while the CAE-MLP utilizes an autoencoder to encode the spatial dimension into a latent space. The decoder subsequently reconstructs the original dimension from the latent representation. The reduced latent space is mapped to the input variables using a multilayer perceptron (MLP). In the online phase, new input datasets are provided, and the trained models predict the corresponding outputs. For



**Fig. 12** Comparison of the x strains found with the POD-ANN and CAE-MLP models with the experimental results at  $z=8.25, 8.75$  and  $9.25$ , in (a), (b) and (c), respectively

CAE-MLP, the predicted latent vectors are decoded back into the original spatial dimension, enabling reconstruction of the strain fields.

The efficiency and accuracy of these approaches were validated on two test cases: a steady-state heat transfer problem and the AMB2018-01 SLM benchmark from NIST. Both models achieved excellent agreement with high-fidelity simulations within  $\pm 15\%$  variation of input parameters. In the benchmark SLM case, CAE-MLP provided a superior approximation of statistical moments relative to the original snapshot matrix and outperformed POD-ANN in modeling nonlinear complexities. The CAE-MLP exhibited lower relative error norms and demonstrated enhanced capability for representing intricate physical phenomena.

Moreover, predictions from both models were compared to experimental data and results from a deep neural network (DNN) model reported in [40]. Notably, both POD-ANN and CAE-MLP showed improved correlation with experimental observations, with CAE-MLP achieving prediction accuracy gains of 60–80% over the prior DNN approach. These findings underscore the potential of the proposed methods for generating high-fidelity surrogates capable of capturing the essential dynamics of additive manufacturing processes.

While these models exhibit promising performance, their reliance on training data limits generalization beyond the sampled parameter space. Future work will focus on incorporating uncertainty quantification techniques—such as ensemble methods or Bayesian neural networks [43–45], to provide confidence intervals, enhance robustness, and ensure reliability in extrapolative scenarios. Such developments will make

these reduced-order models even more applicable to real-world industrial applications where variability and unknowns are inevitable. Furthermore, recent advances in Physics-Informed Neural Networks (PINNs) for additive manufacturing—including thermomechanically augmented PINNs for stress prediction [19] and transfer learning-enhanced PINNs for melt pool morphology estimation [25] suggest promising avenues for hybrid approaches. Combining physics-based constraints with data-driven ROMs could enhance generalization, interpretability, and reliability, paving the way for their deployment in real-world SLM workflows and optimization pipelines.

Thus, the proposed POD-ANN and CAE-MLP models present powerful tools for studying highly nonlinear, complex physical systems by reducing computational costs without sacrificing accuracy. They offer a compelling foundation for advancing surrogate modeling in additive manufacturing and beyond.

## Appendix A

This section provides the convergence history and the architecture for the CAE-MLP and POD-ANN models.

Table A.1: CAE space architecture for the heat transfer test case:

**Table A.1a** Encoder-Space

	Filters	Activation function	Kernel shape
Input	-	-	-
Conv-pooling	32	Prelu	$3 \times 2$
Conv-pooling	68	Prelu	$3 \times 2$
Conv-pooling	128	Prelu	$3 \times 2$
Flatten	-	-	-
Dense	-	Prelu	-
Dense (output - $L_x$ )	-	Prelu	-

**Table A.1b** Decoder- space

	Filters	Activation function	Kernel shape
Input ( $L_x$ )	-	-	-
Dense	-	Prelu	-
Reshape	-	Prelu	-
Conv-upsamp	128	Prelu	$3 \times 2$
Conv-upsamp	68	Prelu	$3 \times 2$
Conv-upsamp	32	Prelu	$3 \times 2$
Output	1	Prelu	3

**Table A.2** MLP architecture for heat transfer case

Layer type	Output size	Activation function
Input layer	2	Prelu
Dense layer	128	Prelu
Dense layer	128	Prelu
Dense layer	128	Prelu
Output layer	$L_x = 5$	Linear

Table A3: CAE space architecture for the AMB2018-01 benchmark case:

**Table A 3a** Encoder-Space

	Filter size	Kernel shape	Activation function
Input	-	-	-
Con-pooling	10	$3 \times 2$	Prelu

**Table A.3a** Encoder-Space

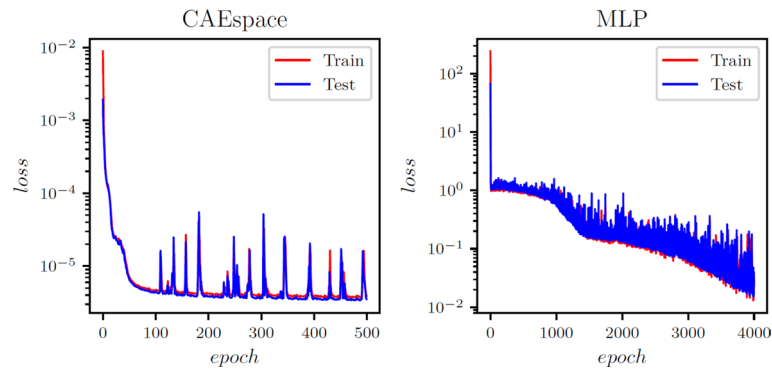
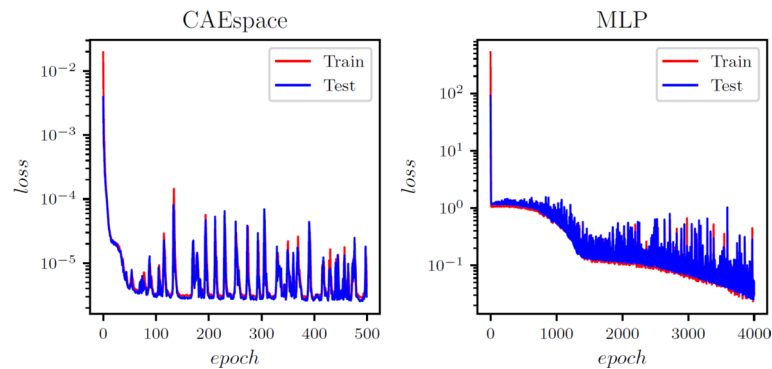
	Filter size	Kernel shape	Activation function
Con-pooling	15	$3 \times 2$	Prelu
Flatten	-	-	-
Dense	-	-	Prelu
Dense (output - $L_x$ )	-	-	Prelu

**Table A.3b** Decoder- Space

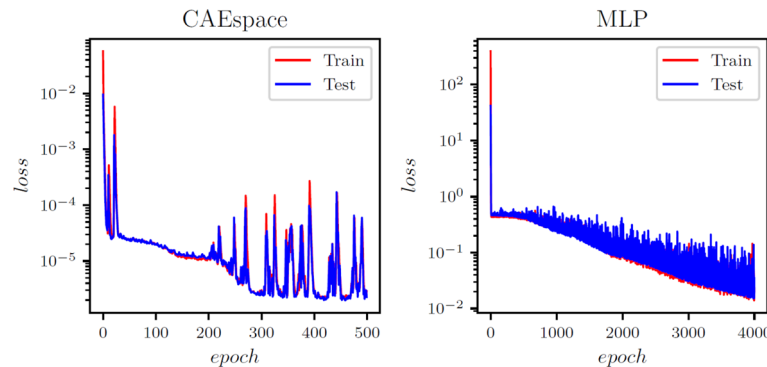
	Filter size	Kernel shape	Activation function
Input ( $L_x$ )	-	-	-
Dense	-	-	Prelu
Reshape	-	-	Prelu
Con-upsam	15	$3 \times 2$	Prelu
Con-upsam	10	$3 \times 2$	Prelu
Output layer	1	3	Prelu

**Table A.4** MLP architecture for the AMB2018-01 cases

Layer type	Output size	Activation function
Input layer	5	Prelu
Dense layer	128	Prelu
Dense layer	128	Prelu
Dense layer	128	Prelu
Output layer	$L_x = 19,40,60$	Linear

**Fig. B1** Evolution of the training and validation error for the x-strain in the AMB2018-01 benchmark case**Fig. B2** Evolution of the training and validation error for the y-strain in the AMB2018-01 benchmark case





**Fig. B3** Evolution of the training and validation error for the z-strain in the AMB2018-01 benchmark case

#### Author contributions

S.C.: Writing – original draft, Visualization, Validation, Software, Investigation. A.A.: Software, Validation. A.S. Soulaïmani: Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

#### Data availability

Data cannot be shared openly but are available on request from authors.

#### Declarations

##### Competing interests

The authors declare no competing interests.

Received: 20 May 2025 / Accepted: 26 July 2025

Published online: 05 August 2025

#### References

- Meng L, et al. Machine learning in additive manufacturing: a review. *JOM*. June 2020;72(6):2363–77. <https://doi.org/10.1007/s11837-020-04155-y>.
- Wang C, Tan XP, Tor SB, Lim CS. Machine learning in additive manufacturing: State-of-the-art and perspectives. *Addit Manuf*. Dec. 2020;36:101538. <https://doi.org/10.1016/j.addma.2020.101538>.
- Ravichander BB, Rahimzadeh A, Farhang B, Shayesteh Moghaddam N, Amerinatanzi A, Mehrpouya M. A prediction model for additive manufacturing of inconel 718 superalloy. *Appl Sci*. Aug. 2021;11(17):8010. <https://doi.org/10.3390/app11178010>.
- Chaudhry S, Soulaïmani A. A comparative study of machine learning methods for computational modeling of the selective laser melting additive manufacturing process. *Appl Sci*. 2022;12(5):2324.
- Francis J, Bian L. Deep Learning for Distortion Prediction in laser-based additive manufacturing using big data. *Manuf Lett*. Apr. 2019;20:10–14. <https://doi.org/10.1016/j.mfglet.2019.02.001>.
- Mozaffar M, et al. Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks. *Manuf Lett*. Oct. 2018;18:35–9. <https://doi.org/10.1016/j.mfglet.2018.10.002>.
- Rong-Ji W, Xin-hua L, Qing-ding W, Lingling W. Optimizing process parameters for selective laser sintering based on neural network and genetic algorithm. *Int J Adv Manuf Technol*. June 2009;42:11–2. <https://doi.org/10.1007/s00170-008-1669-0>.
- Li Z, Zhang Z, Shi J, Wu D. Prediction of surface roughness in extrusion-based additive manufacturing with machine learning. *Robot Comput - Integr Manuf*. June 2019;57:488–95. <https://doi.org/10.1016/j.rcim.2019.01.004>.
- Meng L, Zhang J. Process design of laser powder bed fusion of stainless steel using a Gaussian process-based machine learning model. *JOM*. Jan. 2020;72(1):420–8. <https://doi.org/10.1007/s11837-019-03792-2>.
- Pearson K. LIII. On lines and planes of closest fit to systems of points in space, Lond. *Edinb. Dublin Philos. Mag J Sci*, Nov. 1901;2(11):559–572. <https://doi.org/10.1080/14786440109462720>.
- Xiao D, Fang F, Pain CC, Navon IM. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Comput Methods Appl Mech Eng*. Apr. 2017;317:868–89. <https://doi.org/10.1016/j.cma.2016.12.033>.
- Chatterjee A. An introduction to the proper orthogonal decomposition. *Curr Sci*. 2000;78(7):808–17.
- Moçayd NE, Shadi Mohamed M, Ouazar D, Seaid M. Stochastic model reduction for polynomial chaos expansion of acoustic waves using proper orthogonal decomposition. *Reliab Eng Syst Saf*. Mar. 2020;195:106733. <https://doi.org/10.1016/j.res.s.2019.106733>.
- Abdedou A, Soulaïmani A. A non-intrusive reduced-order modeling for uncertainty propagation of time-dependent problems using a B-splines Bézier elements-based method and proper orthogonal decomposition: Application to dam-break flows. *Comput Math Appl*. Nov. 2021;102:187–205. <https://doi.org/10.1016/j.camwa.2021.10.006>.
- Jacquier P, Abdedou A, Delmas V, Soulaïmani A. Non-intrusive reduced-order modeling using Uncertainty-Aware deep neural networks and proper orthogonal decomposition: application to flood modeling. *J Comput Phys*. Jan. 2021;424:109854. <https://doi.org/10.1016/j.jcp.2020.109854>.
- Sun X, Choi J-I. Non-intrusive reduced-order modeling for uncertainty quantification of space-time-dependent parameterized problems. *Comput Math Appl*. Apr. 2021;87:50–64. <https://doi.org/10.1016/j.camwa.2021.01.015>.



17. Abdedou A, Soulaïmani A. A non-intrusive B-splines Bézier elements-based method for uncertainty propagation. *Comput. Methods Appl Mech Eng.* Mar. 2019;345:774–804 <https://doi.org/10.1016/j.cma.2018.10.047>
18. Zhao X, et al. Enhancing standard finite element codes with POD for reduced order thermal analysis: application to electron beam melting of pure tungsten. *Mater Today Commun.* 2021;29:102796.
19. Sharma R, Guo YB. Thermal-mechanical physics informed deep learning for fast prediction of thermal stress evolution in laser metal deposition. Dec. 25, 2024, arXiv: arXiv:2412.18786. <https://doi.org/10.48550/arXiv.2412.18786>
20. Fic A, Bialecki RA, Kassab AJ. Solving transient nonlinear heat conduction problems by proper orthogonal decomposition and the finite-element method, *Numer. Heat Transf. Part B Fundam.* Aug. 2005;48(2):103–124 <https://doi.org/10.1080/10407790590935920>
21. Bialecki RA, Kassab AJ, Fic A. Proper orthogonal decomposition and modal analysis for acceleration of transient FEM thermal analysis. *Int. J Numer Methods Eng.* Feb 2005;62(6):774–797 <https://doi.org/10.1002/nme.1205>
22. Liu X, Liu X. Regression trees on grassmann manifold for adapting reduced-order models, arXiv, arXiv:2206.11324, June 2022. 2022. [Online]. Available: <http://arxiv.org/abs/2206.11324>. Accessed: Oct. 31, 2022.
23. Liao S, Xue T, Jeong J, Webster S, Ehmann K, Cao J. Hybrid thermal modeling of additive manufacturing processes using physics-informed neural networks for temperature prediction and parameter identification, *Comput Mech.* Sept. 2023;72(3):499–512. <https://doi.org/10.1007/s00466-022-02257-9>
24. Li S, A physics-informed neural network framework to predict 3D temperature field without labeled data in process of laser metal deposition. *Eng Appl Artif Intell.* Apr. Intell. 2023;120(C)<https://doi.org/10.1016/j.engappai.2023.105908>
25. Zhu Q, Lu Z, Hu Y. Transfer learning-enhanced physics informed neural network for accurate melt pool prediction in laser melting. *Adv Manuf.* Jan. 2025;2(1): Art. no. 1 <https://doi.org/10.55092/am20250001>
26. Abdedou A, Soulaïmani A. Reduced-order modeling for stochastic large-scale and time-dependent problems using deep spatial and temporal convolutional autoencoders, arXiv, arXiv:2208.03190, Aug. 2022. [Online]. Available: <http://arxiv.org/abs/2208.03190>. Accessed: Oct. 31, 2022.
27. Zhu J, Shi H, Song B, Tao Y, Tan S. Information concentrated variational auto-encoder for quality-related nonlinear process monitoring. *J Process Control.* Oct. 2020;94:12–25. <https://doi.org/10.1016/j.jprocont.2020.08.002>.
28. Zhu J, Shi H, Song B, Tan S, Tao Y. Deep neural network based recursive feature learning for nonlinear dynamic process monitoring. *Can J Chem Eng.* Apr. 2020;98(4):919–933 <https://doi.org/10.1002/cjce.23669>
29. Tan Y. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA: IEEE, Dec. 2019, pp. 1008–1015. <https://doi.org/10.1109/ICMLA.2019.00171>
30. He Y, Shi H, Tan S, Song B, Zhu J. Multiblock temporal convolution network-based temporal-correlated feature learning for fault diagnosis of multivariate processes. *J Taiwan Inst Chem Eng.* May 2021;122:78–84. <https://doi.org/10.1016/j.jtice.2021.04.062>.
31. Gonzalez FJ, Balajewicz M. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems, arXiv, arXiv:1808.01346, Aug. 2018. [Online]. Available: <http://arxiv.org/abs/1808.01346>. Accessed: Jan. 09, 2023.
32. Shi Z, Mamun AA, Kan C, Tian W, Liu C. An LSTM-autoencoder based online side channel monitoring approach for cyber-physical attack detection in additive manufacturing. *J Intell Manuf.* Jan. 2022. <https://doi.org/10.1007/s10845-021-01879-9>.
33. Siddalingappa R, Kanagaraj S. Anomaly detection on medical images using autoencoder and convolutional neural network. *Int J Adv Comput Sci Appl.* 2021;12(7):10.
34. Tang W, Vian CM, Tang Z, Yang B. Anomaly detection of core failures in die casting X-ray inspection images using a convolutional autoencoder. *Mach Vis Appl.* July 2021;32(4):102. <https://doi.org/10.1007/s00138-021-01226-1>.
35. Workbench additive manufacturing analysis guide. 2021. [Online]. Available: <https://www.ansys.com/products/additive>
36. Phan TQ. Elastic residual strain and stress measurements and corresponding part deflections of 3d additive manufacturing builds of IN625 AM-Bench artifacts using neutron diffraction, Synchrotron X-Ray Diffraction, and Contour Method. Integrating Mater Manuf Innov. Sept. 2019;8(3):318–334 <https://doi.org/10.1007/s40192-019-00149-0>
37. Lumley JL. The Structure of Inhomogeneous Turbulence In: Yaglom AM and Tatarski VI, Eds., *Atmospheric Turbulence and Wave Propagation*. Nauka, Moscow 1967, pp. 166–178.
38. Weiss J. A tutorial on the proper orthogonal decomposition. in AIAA aviation 2019 forum. Dallas, Texas: American Institute of Aeronautics and Astronautics; June 2019. <https://doi.org/10.2514/6.2019-3333>.
39. Maulik R, Lusch B, Balaprakash P. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Phys Fluids.* Mar. 2021;33(3):037106. <https://doi.org/10.1063/5.0039986>
40. Abadi M et al. TensorFlow: A system for large-scale machine learning.
41. Saliby E, Pacheco F. An empirical evaluation of sampling methods in risk analysis simulation: quasi-Monte Carlo, descriptive sampling, and latin hypercube sampling, in Proceedings of the Winter Simulation Conference, San Diego, CA, USA: IEEE, 2002:1606–1610. <https://doi.org/10.1109/WSC.2002.1166440>
42. torey.liepa@nist.gov, Benchmark Test Data, NIST. [Online]. Available: <https://www.nist.gov/ambench/benchmark-test-data>. Accessed: Dec. 22, 2021.
43. Kabir HMD, Khosravi A, Hosen MA, Nahavandi S. Neural network-based uncertainty quantification: A survey of methodologies and applications. *IEEE Access.* 2018;6:36218–34. <https://doi.org/10.1109/ACCESS.2018.2836917>.
44. Jacobsen C. Enhancing physical modeling with interpretable physics-aware machine learning, Thesis, 2024. <https://doi.org/10.7302/23070>
45. Yong BX, Brintrup A. Bayesian autoencoders with uncertainty quantification: Towards trustworthy anomaly detection. Feb. 25, 2022, arXiv: arXiv:2202.12653. <https://doi.org/10.48550/arXiv.2202.12653>

## Publisher's note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.