PHYSICAL REVIEW MATERIALS **7**, 083802 (2023)

Accelerating microstructure modeling via machine learning: A method combining Autoencoder and ConvLSTM

Owais Ahmad, Naveen Kumar, Rajdip Mukherjee, and Somnath Bhowmick[*]
Department of Materials Science and Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India

(Received 7 April 2023; revised 22 June 2023; accepted 18 July 2023; published 7 August 2023)

Phase-field modeling is an elegant and versatile computation tool to predict microstructure evolution in materials in the mesoscale regime. However, these simulations require rigorous numerical solutions of differential equations, which are accurate but computationally expensive. To overcome this difficulty, we combine two popular machine-learning techniques, autoencoder and convolutional long short-term memory (ConvLSTM), to accelerate the study of microstructural evolution without compromising the resolution of the microstructural representation. After training with phase-field-generated microstructures of 10 known compositions, the model can accurately predict the microstructure for the future *n*th frames based on the previous *m* frames for an unknown composition. Replacing *n* phase-field steps with machine-learned microstructures can significantly accelerate the *in silico* study of microstructure evolution.

DOI: 10.1103/PhysRevMaterials.7.083802

I. INTRODUCTION

In recent years, the rise of artificial intelligence (AI) in science and technology has been phenomenal. With the development of sophisticated machine-learning algorithms and the availability of vast amounts of data, AI has become an indispensable tool for solving complex problems in various fields. AI has also demonstrated great promise in materials science. One of the most active research domains is analyzing vast amounts of data by machine-learning algorithms for the accelerated discovery of materials [1,2]. AI algorithms have been trained for microstructure analysis [3,4], additive manufacturing [5], mapping materials properties to atomic-scale imaging [6], and diagnosing materials failure before they occur to reduce downtime and increase productivity [7].

The phase-field method is a powerful computational tool to model and study microstructure evolution and related properties, including solidification [8–10], precipitate growth [11,12], grain growth [13–15], coarsening [16], effect of external field [17,18], and spinodal decomposition [19,20]. Apart from materials science, it has applications in various domains [21,22,23]. Since the microstructural images in phase-field models are represented by a system of continuously evolving variables in the spatial and temporal domain, this kind of fidelitous phase-field model demands a discretized spatiotemporal representation by partial differential equations, making their implementation computationally expensive and cumbersome, which motivated researchers to minimize the computational costs by primarily leveraging advanced numerical methods [24–26] and high-performance computational architectures [27–29].

As machine learning and deep learning have touched every domain, microstructure evolution is no exception. Researchers have already started leveraging the power of AI to do faster simulations [30]. In this paper, we propose a method for accelerated prediction of microstructure evolution via a machine-learned surrogate model. Figure 1 summarizes the workflow. Like the earlier works [30,31], initially, we generate a dataset of 1000 images each for 10 different compositions, ranging from $c_{\text{avg}} = 0.25$ to 0.5 with the help of phase-field calculations. The size of each frame is $256 \times 256 \times 3$, where the last digit represents the number of channels (red, green, and blue) in the image. Since the dimensions are huge for 10 000 images ($256 \times 256 \times 3$, i.e., 196 608 per image), the required computational resources are also very high and time consuming. To overcome this problem, we need to apply some dimensionality reduction techniques. In this paper, we deploy the autoencoder method [32,33]. The encoder part reduces the dimensions to $32 \times 32 \times 8$ (i.e., 8192), and using this transformed version of data in latent space, we train the model to learn the spatial and temporal variation in the dataset. For predicting time series image data, we create a model with convolutional long short-time memory (ConvLSTM) [34]. The model can accurately predict the microstructure for the future *n*th frames based on previous *m* frames for an unknown composition. Finally, the decoder takes this predicted frame (still in latent space) and projects it into its original dimensions. Combined with the autoencoder, ConvLSTM proves to be a very robust technique to learn the spatiotemporal variation of microstructure with a minimal dataset for the binary phase. Our method is comparable with the ones carried out using principal component analysis (PCA) with a recurrent neural network and PCA with LSTM [30,31].

This paper is organized in the following manner: In Sec. II, we describe the details of the phase-field model and data generation. In Sec. III, we discuss the autoencoder method for dimensionality reduction. In Sec. IV, we talk about spatiotemporal prediction using ConvLSTM, then reconstructing the microstructure using the decoder. In Sec. V, we discuss the

---

[*] bsomnath@iitk.ac.in

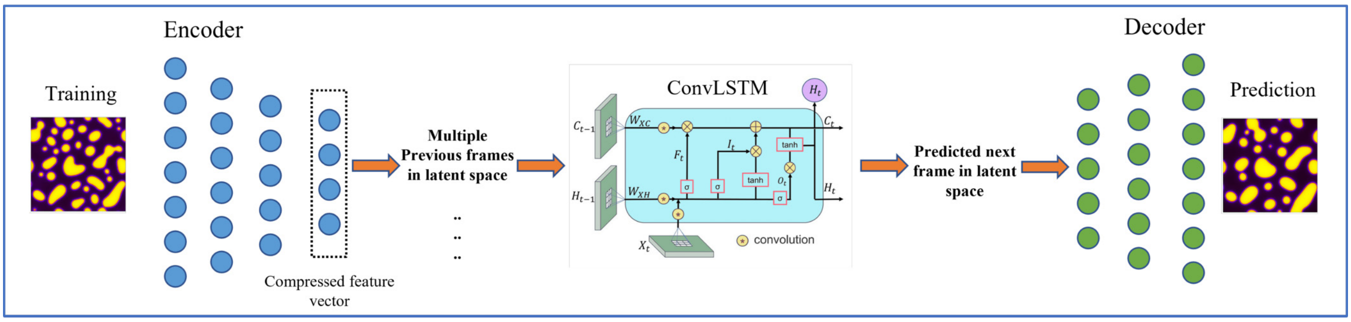

FIG. 1. Workflow of training machine-learning model with phase-field-generated microstructures and building a machine-learned surrogate model for accelerated prediction of microstructure evolution.

salient features of the method used in this paper and compare it with other possible techniques. Finally, we conclude the paper in Sec. VI.

II. PHASE-FIELD MODEL FOR SPINOIDAL DECOMPOSITION IN A BINARY ALLOY

In this paper, we employ a phase-field model to generate a training dataset of microstructure evolution during spinodal decomposition in a A-B binary alloy. A schematic phase diagram is shown in Fig. 2 (top left), which shows the chemical spinodal lines (in blue) and the miscibility gap (in red), respectively. The total free energy of the alloy is [35]

$$F = \int_V [f(c) + \kappa(\nabla c)^2] dV. \tag{1}$$

The local composition of the system is denoted by a conserved phase-field variable $c(\mathbf{r}, t)$ (space: $\mathbf{r}$, and time: t). The bulk free energy density $f(c)$ is schematically shown in Fig. 2 (top right), represented by a double well potential, and is given by

$$f(c) = Wc^2(1-c)^2, \tag{2}$$

where W is a constant determining the potential barrier height between the two equilibrium phases corresponding to the compositions $c = 0$ and 1, respectively. Spinodal decomposition occurs in the composition range between the two inflection points where $\partial^2 f/\partial c^2 < 0$. The gradients in local composition also contribute to the total free energy as given by the term $\kappa(\nabla c)^2$ in Eq. (1), where κ is the gradient energy coefficient. The spatiotemporal evolution of the conserved phase-field variable (composition c in this case) is governed by the Cahn-Hilliard equation [36]:

$$\frac{\partial c}{\partial t} = M[\nabla^2 g(c) - 2\kappa \nabla^4 c], \tag{3}$$

where M is the atomic mobility (assumed to be a constant), and $g(c) = \partial f/\partial c$. Further details about the phase-field model and its numerical implementation are provided in the Supplemental Material [37] (see also Ref. [38] therein).

Using Latin hypercube sampling, we generate 10 combinations of phase fractions values ϕ_A and ϕ_B. Since we are interested in studying the spinodal decomposition of a binary system, the ϕ_A value should lie within a range of 0.25–0.75. However, because of the symmetry of the potential, the minimum and maximum values of ϕ_A are set to 0.25 and 0.5, respectively. The phase mobilities for both components are set to 1. The simulations are performed on a two-dimensional (2D) square domain, discretized with 256 × 256 grid points, and the microstructure evolution and growth are allowed for 1000 time steps. Figure 2 shows the microstructure evolution in two different alloys, having initial compositions 0.25 and 0.5, respectively. We save the information of the microstructural state every single time step, yielding a total of 1000 time frames per composition. Since there are 10 compositions, we have a total of 10 000 microstructure evolution images of 256 × 256 resolution. The initial compositional field is distributed arbitrarily in space, and the microstructure has no discernible features from frame t_0 to t_{20}. Thus, we discard the initial part and initiate training at frame t_{20}. The subdomains develop rapidly between frames t_{20} and t_{100}, followed by a smooth and consistent agglomerating outgrowth of the microstructure from frames t_{100} to t_{1000}. We want our machine-learning model to anticipate both the rapid development and gradual growth phases of microstructure evolution. We evaluate the model using 80-20 train-test splits and verified that the outcome is similar for 60-40 and 70-30.

III. DIMENSIONALITY REDUCTION WITH ENCODER AND RECONSTRUCTION WITH DECODER

The enormous dimensionality of the phase-field data in the format of the microstructural image is precisely where the manifold hypothesis [39] can be leveraged to establish an accelerated framework for studying the microstructural evolution. However, for the microstructure-learning model to operate effectively, one must transform the 256 × 256 × 3 phase-field data per frame via a dimensionality-reduction procedure into a more compact and manageable dataset. A dimensionality-reduction algorithm aims to describe the data with fewer characteristics while retaining as much information as possible. In this paper, we use the autoencoder for dimensionality reduction and transforming phase-field microstructure data in a smaller latent space for the ConvLSTM model to learn more efficiently. The performance of another popular dimensionality reduction method, the PCA [40], is compared with the autoencoder, and the latter is found to be more accurate.

PCA is a linear transformation of the high-dimensional data that discards the insignificant modes (eigen/singular) with lower eigen/singular values, transforming the data into a low-dimensional form. Due to the nonlinear nature of the system, employing PCA to decrease the dimension of the

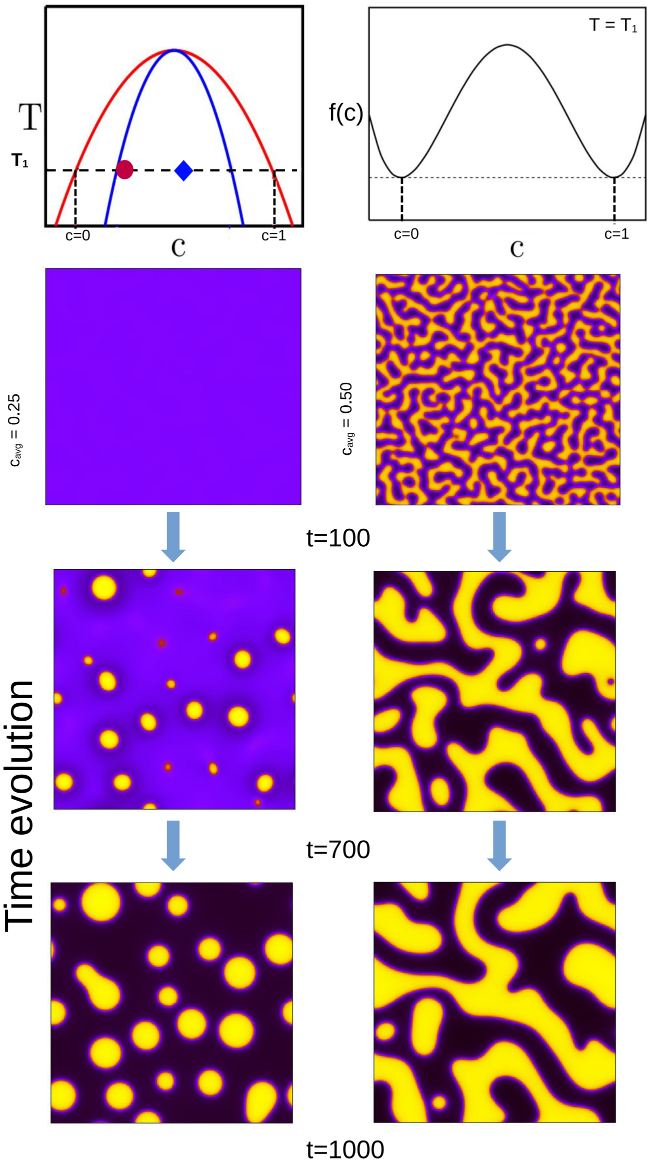

FIG. 2. Phase diagram with miscibility gap (top left) and the corresponding bulk free energy $f(c)$ versus composition c diagram at temperature $T = T1$ (top right). The microstructures show the time evolution (spinodal decomposition) during isothermal ($T = T1$) aging of alloys with two different initial compositions 0.25 and 0.5, respectively, at time $t = 100$, 700, and 1000. Red (circle) and blue (diamond) points on the phase diagram show these two compositions, respectively. Total 10 000 such images (10 different compositions, 1000 time frames per composition) are used as training set.

microstructure representation may result in the loss of vital information if only a few appropriate principle components are examined. We need a nonlinear mapping technique from a high-dimensional spatial version to a low-dimensional latent space while avoiding loss of information, as in the case of PCA. An autoencoder does precisely this. The encoder reduces the dimension to lower latent dimensions via a nonlinear mapped version of high-dimensional microstructure data (p, q, t) into a low-dimensional version but in latent space represented by (r). At the same time, the decoder learns the reverse mapping from low-dimensional latent space to high-dimensional microstructure. Mathematically, we can write this as

$$\alpha_{\theta_{\mathrm{enc}}} : \phi(\mathrm{p, q, t}) \rightarrow \tilde{\phi}(r), \tag{4}$$

$$\beta_{\theta_{\mathrm{dec}}} : \tilde{\phi}(r) \rightarrow \phi(\mathrm{p, q, t}). \tag{5}$$

Here, α and β represent the mapped versions as transformed by the encoder and the decoder, respectively. In Eq. (4), the encoder takes ϕ (p,q,t) $\in \mathbb{R}^{256\times256\times3}$ as input and maps it to a latent space $\tilde{\phi}(r) \in \mathbb{R}^{l_d}$ with l_d dimensions. The error $\mathcal{L}_{\mathrm{ae}}$ given below is minimized while training the autoencoder:

$$\mathcal{L}_{\mathrm{ae}} = \min_{\theta_{\mathrm{ae}}=(\theta_{\mathrm{enc}},\theta_{\mathrm{dec}})} \|\phi(\mathrm{p,q,t}) - \tilde{\phi}(\mathrm{p,q,t};\theta_{\mathrm{ae}})\|_2^2. \tag{6}$$

Here, θ_{ae} represents the parameters for training the autoencoder. This transformed data in latent space is the training data for the ConvLSTM model. We observe a faster convergence in training ConvLSTM while feeding the dimensionally reduced data than actual data. The predicted output from ConvLSTM is then transformed back to its original dimensions (in the form of a microstructure) with the decoder, the second part of the autoencoder.

The optimization of the autoencoder (both encoder and decoder) architecture is initially undertaken to preserve the majority of the features in latent space. A series of tests are conducted, starting with adding two cells (single cell refers to a single layer of convolutional filters) and gradually increasing to four. As depicted in Fig. 3, it is observed that architectures featuring $\leqslant$3 encoder layers performed much better than $\geqslant$4 encoder layers. The transformed data lose their correlation with the features in $>$3-layer encoders, and the decoder cannot accurately reconstruct the data from the transformed features. In the first row of Fig. 3, 2 cells are taken for encoding; the reconstructed image is very close to the original image, and the heat map shows tiny red regions, implying very little difference between the original and reconstructed image. In the second row, Fig. 3, 3 cells are taken for encoding; the reconstructed image is close to the original image but not as good as the case of 2 cells. As we increase the number of cells to four in the third row of Fig. 3, the reconstructed image gets distorted, and the difference in certain regions reaches yellow in the heat map, implying a significant difference in actual and reconstructed pixel values. Finally, we choose to use a 3-cell encoder, as it offers a higher dimensionality reduction with minimal data loss. To quantify the difference between the original and reconstructed microstructure, we have chosen mean point-wise error (MPE). It quantifies the average discrepancy between corresponding points in the microstructures. MPE provides a measure of similarity or dissimilarity by calculating the absolute difference between the values at each point in the two microstructures and then averaging these differences. This metric is used in various applications, such as image registration, object recognition, and computer vision, enabling effective comparison.

To optimize the speed of the model training, experimentation is conducted using a different number of images in the dataset. The model is trained using datasets of 500, 1000, 2000, 3000, 4000, and 5000 images for 1000 epochs. The results indicate that, while the architecture can learn the

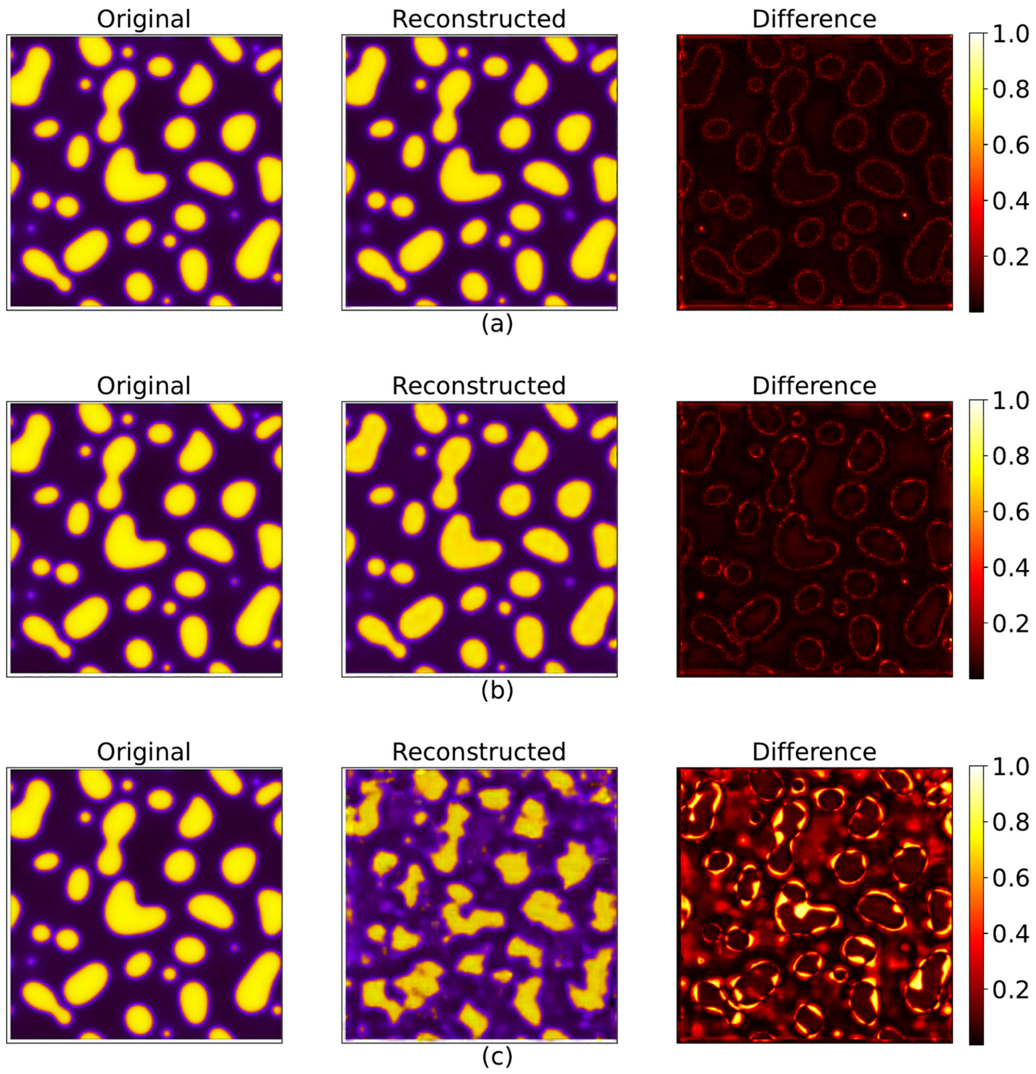

FIG. 3. Images are reduced in dimensions using the autoencoder. We find that 2000 images are sufficient for training the autoencoder; a heat map comparison of the original (encoded) and reconstructed (decoded) images using (a) 2 cells (MPE = 0.06), (b) 3 cells (MPE = 0.08), and (c) 4 cells (MPE = 0.4) autoencoders are presented. MPE stands for mean point-wise error (see text for further details). Reconstructed images are comparable with the original ones when 2–3 cells are used.

microstructure evolution profile using 500 images, it struggles to track the phases within that profile accurately. However, as the number of images in the training dataset increases, the model performs with an accuracy of 98%. However, its performance in terms of loss and accuracy shows minimal improvement after 2000 images.

IV. TRAINING CONVLSTM, PREDICTION AND RECONSTRUCTING MICROSTRUCTURE WITH DECODER

ConvLSTM is a neural network architecture that combines the advantages of a convolutional neural network [41] with LSTM networks [34]. The ConvLSTM architecture can analyze and learn spatial information with its temporal dependencies, such as in video or time series data. Inputs $\mathcal{X}_1$, $\mathcal{X}_2$,...$\mathcal{X}_t$, cell outputs $\mathcal{C}_1$, $\mathcal{C}_2$...$\mathcal{C}_t$, hidden states $\mathcal{H}_1$, $\mathcal{H}_2$...$\mathcal{H}_t$, and gates i_t, f_t, o_t, all of these in the case of ConvLSTM are three-dimensional (3D) tensors, the last two dimensions being spatial. This is the advantage of ConvLSTM over LSTM because the spatial information is lost in the latter. The following equations represent the fundamental structure of a ConvLSTM cell. Here, $*$ represents the convolution operator, and $\circ$ represents the Hadamard product:

$$i_t = \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i), \quad (7)$$

$$f_t = \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f), \quad (8)$$

$$\mathcal{C}_t = f_t \circ \mathcal{C}_(t-1) * i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c), \quad (9)$$

$$o_t = \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o), \quad (10)$$

$$\mathcal{H} = o_t \circ \tanh(\mathcal{C}_t). \quad (11)$$

Here, W_{xi}, W_{hi}, W_{ci}, W_{xf}, W_{hf}, W_{cf}, W_{xc}, W_{hc}, W_{xo}, W_{ho}, and W_{co} represent weights for respective variables while b_i, b_f, b_c, and

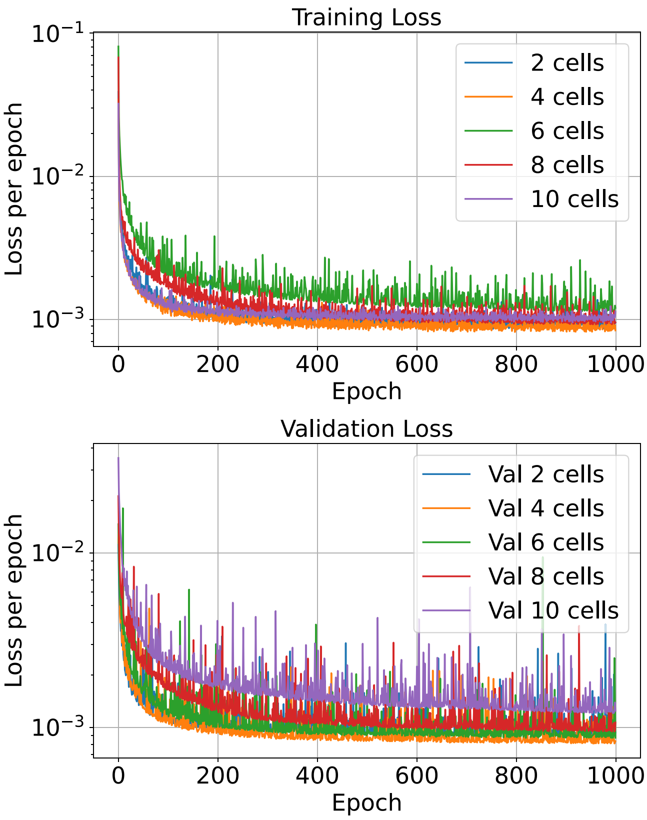

FIG. 4. The ConvLSTM model is trained with a compressed-image dataset in latent space. The architecture of the ConvLSTM model comprises 2, 4, 6, 8, and 10 cells. The illustration compares training and validation loss for different ConvLSTM architectures in log scale.

b_o represent the bias for each gate. ConvLSTM predicts the future state of a grid cell based on the inputs and previous states of its immediate neighbors. Utilizing a convolution operator for the state-to-state and with input-to-state transitions makes this simple to implement. Before conducting the convolution operation, padding is required to ensure that the output states and the inputs have the same number of rows and columns. Using the state of the outer environment for computational purposes might be interpreted as padding for the hidden states at the boundary points. Before the first input, we typically set the LSTM states to zero, eliminating the dependency on the future. If we set padding as zero on the hidden states, it would set the state of the outside world to zero, assuming that it is unaware of the outside world. As the microstructures can be periodic, the padding has been set the same for this paper.

After reducing the microstructure dataset to latent dimensions using the optimized autoencoder model, the ConvLSTM architecture is applied to learn the changes in the microstructure with respect to time. In the process of optimizing the ConvLSTM architecture, the number of cells is gradually increased in steps of two as 2, 4, 6, 8, and 10. Figure 4 compares the training loss and the validation loss for the different numbers of cells. The loss for ConvLSTM training is defined as the mean squared error (MSE):

$$\mathrm{MSE}_{a_j} = \frac{1}{KN}\sum_{k=1}^{K}\sum_{i=1}^{N}\left[\hat{a}_j^{(k)}(t_i) - \tilde{a}_j^{(k)}(t_i)\right]^2. \qquad (12)$$

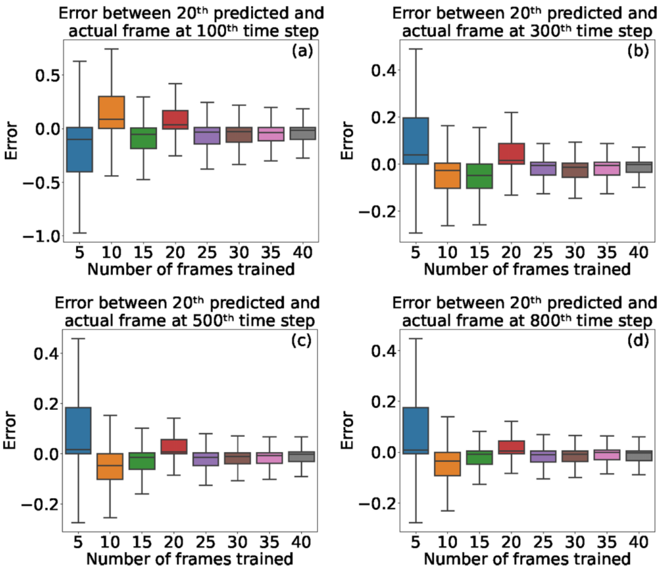

FIG. 5. This graph compares the relative error between the actual and predicted frame. For predicting the 20th frame, the number of previous frames used are 5, 10, 15, 20, 25, 30, 35, and 40. The comparison is made at different stages of the microstructure evolution: (a) 120th, (b) 320th, (c) 520th, and (d) 820th time steps.

Here, N is the number of time frames for which the error is calculated, and K represents the total number of microstructure evolution predictions for which the error is calculated. Also, $\hat{a}_j^{(k)}$ and $\tilde{a}_j^{(k)}$ represent the actual and predicted values of the pixel in latent space, respectively. We find that 2 and 4 ConvLSTM cells outperform all other architectural configurations when trained and compared over 1000 epochs. Since 4 ConsLSTM cells offer minimum loss at the expense of minimum computational cost, we use the same for the rest of this paper. Though 100 epochs are sufficient for making the predictions, we show up to 1000 epochs to illustrate that the model stays stable over 1000 epochs.

The final optimization step for the ConvLSTM model involves varying the number of previous frames used for predicting the next frame. The model is trained to predict the next frame based on the previous 5, 10, 15, 20, 25, 30, 35, and 40 frames. The predicted frame from ConvLSTM is in the latent space and is projected to its actual dimension with the decoder. As explained previously, the decoder is the second half of the autoencoder, the same model we optimized earlier for dimensionality reduction. After reconstructing the predicted image with the decoder, we compare it with the actual microstructure (directly obtained from phase-field simulations). Panel (a) of Fig. 5 compares the error in the 20th predicted frame based on 5, 10, and up to 40 preceding frames prior to the 100th time step; it is evident that, as the number of frames grows, the inaccuracy in the predicted frame reduces. Additionally, we find that, after 25 frames, the error in the predicted frame does not vary significantly. Similar conclusions can be drawn while predicting the 300th, 500th, and 800th time steps, as shown in panels (b), (c), and (d) of Fig. 5, respectively. Thus, one can conclude that the absolute relative error produced in the predicted frame decreases as the number of previous frames used for prediction increases, and >25 frames would be a

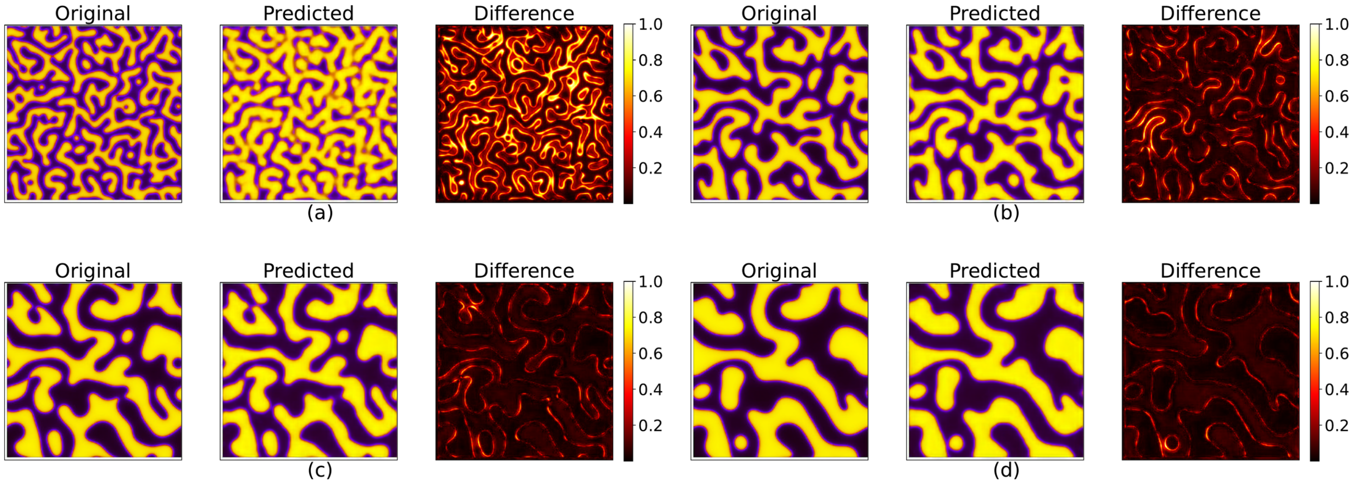

FIG. 6. A model trained on 40 previous frames to predict the next frame is used to make the final predictions. This figure illustrates the difference between the actual and predicted frame at the 120th (MPE = 0.4), 320th (MPE = 0.09), 520th (MPE = 0.07), and 820th (MPE = 0.04) time step as a heat map.

safe choice for the method to predict the 20th frame with reasonable accuracy. For the rest of the discussion, we consider microstructures predicted with 40 preceding frames. Figure 5 also illustrates the error in predicted frames at different stages of the microstructure evolution. Comparing the four panels of Fig. 5, one can further conclude that error in predicted frames is smaller in the later stages of the microstructure evolution than initial stages.

V. DISCUSSION

So far, we have discussed a method that starts with microstructure generation using a phase-field model. Next, we use the autoencoder, which has two parts. The encoder part carries out the dimensionality reduction of the microstructures to a latent space, and the decoder part reconstructs the original microstructure back from the latent space. Using the data in latent space (obtained from the encoder), we train a ConvLSTM model, which predicts spatiotemporal evolution in latent space itself. Finally, using the predicted data in latent space, the decoder reconstructs the predicted microstructure.

While the principle is straightforward, one must still optimize the parameters to obtain the highest accuracy at a minimal computation cost. We need to optimize at two levels; first, while training the autoencoder and, next, while training the ConvLSTM. For example, 2000 images are sufficient to train the autoencoder, and a 3-cell encoder offers maximum dimensionality reduction at the expense of minimum data loss (see Fig. 3). Next, ConvLSTM, trained with 4 cells and up to 100 epochs, is sufficient for prediction (see Fig. 4). Since the model is trained to predict the next frame based on the previous frames, optimizing the number of previous frames used for prediction is also essential. We find that the errors between the actual and predicted frame can be minimized by using ∼40 previous frames to predict the 20th frame, although ∼25 previous frames should be sufficient for this purpose (see Fig. 5).

Interestingly, the prediction quality also depends on the microstructure evolution stage, evident from heat maps shown in Fig. 6. These are generated based on predictions using the previous 40 frames. In each case of 100th, 300th, 500th, and 800th time steps during the microstructure evolution, 40 previous frames are taken, and the subsequent 20 frames are predicted, yielding the 120th, 320th, 520th, and 820th frames, respectively. The pixel-by-pixel error heat map shows that, although the model can capture the overall profile at any stage of microstructure evolution, the error is higher during the initial stage (at the 120th time step) and decreases significantly at later stages. Thus, one needs to be cautious, particularly with the predictions during the initial stage of the microstructure evolution.

The autoencoder-ConvLSTM model can accelerate *in silico* study of microstructure evolution, as it can predict the nth frame based on the previous m frames. Essentially, we are replacing n phase-field steps with machine-learned microstructures. To accelerate, one needs to maximize n and minimize m, keeping the error within an acceptable limit. For example, if the number of previous frames used is $m = 40$, how far can we predict before the errors blow up? A comparison, in terms of the heat map and autocorrelation [42], is shown in Fig. 7 for the 10th, 20th, 30th, and 40th frames, predicted from the previous 40 frames. Evidently, the predicted spatial domain starts to differ from the actual state as we move further ahead in the time domain. Figure 7 also presents errors in a predicted microstructure, where PCA is used for the dimensionality reduction instead of the autoencoder. Even the 10th predicted frame has unacceptably high errors.

The autoencoder has proven to be a more robust technique than PCA to learn the spatiotemporal variation of microstructure with a minimal dataset for the binary phase. PCA presumes the linear embedding of microstructure in higher-dimensional space. The binary phase microstructure evolution exhibits a nonlinear trajectory that PCA failed to capture. Some researchers have already shown the advantage of nonlinear dimensionality-reduction techniques such as isometric feature mapping or Isomap [43], uniform manifold approximation and projection or UMAP [44].

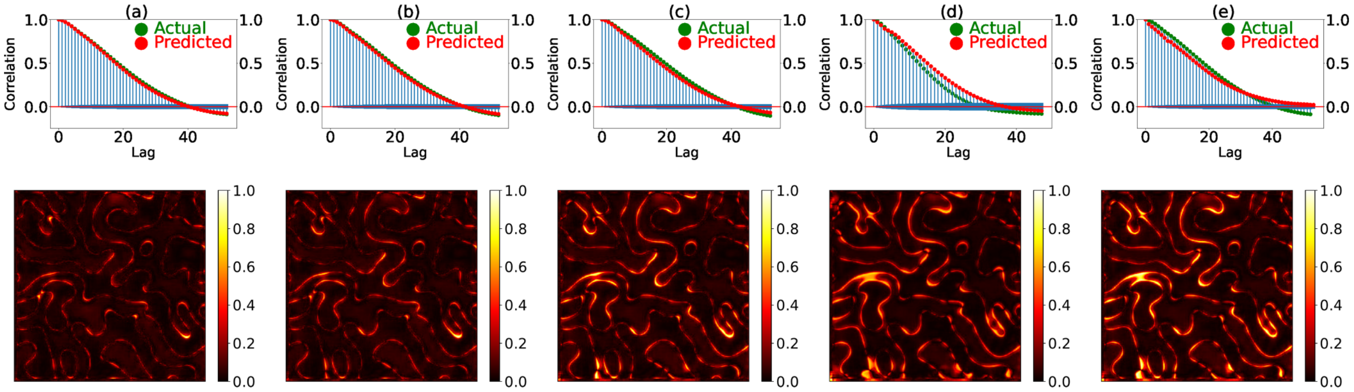

FIG. 7. Comparative autocorrelation and heat map of the (a) 10th (MPE = 0.03), (b) 20th (MPE = 0.06), (c) 30th (MPE = 0.16), and (d) 40th (MPE = 0.25) predicted frame after the model is trained on the preceding 40 frames. The final figure (e) is the 10th (MPE = 0.28) frame predicted, using principal component analysis (PCA) as the dimensionality reduction technique. Comparing (a) and (e) reveals that the autoencoder fares significantly better than PCA.

The calculations are carried out on a server utilizing an Intel Xeon Gold 6226R CPU operating at 2.90 GHz. The system is supported by 128 GB of RAM, allowing efficient data management and processing. It takes 30 min to train the autoencoder model, while the ConvLSTM model requires 60 min. These durations highlight the reasonable time taken by the system to complete the training tasks, facilitating effective model development and analysis. We want to emphasize that the machine-learning model needs to be trained only once before it can be used to predict microstructure evolution for an unknown composition without further retraining. Our ML-assisted phase-field model is faster than standard numerical methods for solving the Cahn-Hilliard equation. While the standard numerical method for phase-field calculation takes 1.8 s for evolving 20 steps, the ML-assisted approach requires only 0.3 s for the same number of steps.

VI. CONCLUSIONS AND FUTURE SCOPE

In conclusion, the autoencoder-ConvLSTM model provides an accelerated framework for microstructure evolution predictions. The performance depends on two parts of the model. The first part is the autoencoder model, which efficiently reduces dimensions to a compact dataset. The computational cost of the autoencoder is relatively inexpensive. The autoencoder can be trained for as small as 1000 images for the microstructure evolution during spinodal decomposition in a binary system. It performs very well, both in terms of moderate computational resource requirements and relatively less time taken to reconstruct the image in the original spatial dimension. The second part is the ability of the ConvLSTM neural network to learn spatial information with its temporal dependencies, which can predict the microstructure evolution. The model can be further implemented for more complicated cases, e.g., multiphase micorstructure evolution, microstructure evolution under the influence of external magnetic field, and strain field.

Recently, several researchers studied distinct methods for utilizing the underlying correlations between datasets derived from diverse data sources with varying accuracy and obtained optimal predictions [45]. One can consider implementing the multifidelity implementation of the proposed method by including experimental data derived from processes with similar characteristics. In this perspective, data derived from phase-field models using a numerical solver can be regarded as comprising a low-fidelity dataset, while the experimental microstructures and imaging data from analogous processes can be termed high-fidelity datasets.

ACKNOWLEDGMENTS

The authors acknowledge the National Supercomputing Mission for providing computing resources of Param Sanganak at IIT Kanpur, which is implemented by Center for Development of Advanced Computing (C-DAC) and supported by the Ministry of Electronics and Information Technology and Department of Science and Technology, Government of India. The authors are also thankful for the high-performance computing facility provided by Computer Center at IIT Kanpur. RM and SB are thankful for financial support received from C-DAC Project No. Meity/R&D/HPC/2(1)/2014.

O.A. and N.K. contributed equally to this paper.

---

[1] Z. Xiong, Y. Cui, Z. Liu, Y. Zhao, M. Hu, and J. Hu, Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation, Comput. Mater. Sci. **171**, 109203 (2020).

[2] P. C. Jennings, S. Lysgaard, J. S. Hummelshøj, T. Vegge, and T. Bligaard, Genetic algorithms for computational materials discovery accelerated by machine learning, npj Comput. Mater. **5**, 46 (2019).

[3] B. L. DeCost and E. A. Holm, A computer vision approach for automated analysis and classification of microstructural image data, Comput. Mater. Sci. **110**, 126 (2015).

[4] E. A. Holm, R. Cohn, N. Gao, A. R. Kitahara, T. P. Matson, B. Lei, and S. R. Yarasi, Overview: Computer vision and

machine learning for microstructural characterization and analysis, Metallurgical and Materials Transactions A **51**, 5985 (2020).

[5] C. Herriott and A. D. Spear, Predicting microstructure-dependent mechanical properties in additively manufactured metals with machine- and deep-learning methods, Comput. Mater. Sci. **175**, 109599 (2020).

[6] J. Han, K.-J. Go, J. Jang, S. Yang, and S.-Y. Choi, Materials property mapping from atomic scale imaging via machine learning based sub-pixel processing, npj Comput. Mater. **8**, 196 (2022).

[7] R. Li, M. Jin, and V. C. Paquit, Geometrical defect detection for additive manufacturing with machine learning models, Mater. Des. **206**, 109726 (2021).

[8] S. Chatterjee, T. A. Abinandanan, and K. Chattopadhyay, Phase-field simulation of fusion interface events during solidification of dissimilar welds: Effect of composition inhomogeneity, Metall. Mater. Trans. A **39**, 1638 (2008).

[9] J. Hötzer, M. Jainta, P. Steinmetz, B. Nestler, A. Dennstedt, A. Genau, M. Bauer, H. Köstler, and U. Rüde, Large scale phase-field simulations of directional ternary eutectic solidification, Acta Mater. **93**, 194 (2015).

[10] Y. Zhao, B. Zhang, H. Hou, W. Chen, and M. Wang, Phase-field simulation for the evolution of solid/liquid interface front in directional solidification process, J. Mater. Sci. Technol. **35**, 1044 (2019).

[11] R. Mukherjee, T. A. Abinandanan, and M. P. Gururajan, Phase field study of precipitate growth: Effect of misfit strain and interface curvature, Acta Mater. **57**, 3947 (2009).

[12] R. Mukherjee, T. A. Abinandanan, and M. P. Gururajan, Precipitate growth with composition-dependent diffusivity: Comparison between theory and phase field simulations, Scr. Mater. **62**, 85 (2010).

[13] C. E. Krill, L. Helfen, D. Michels, H. Natter, A. Fitch, O. Masson, and R. Birringer, Size-Dependent Grain-Growth Kinetics Observed in Nanocrystalline Fe, Phys. Rev. Lett. **86**, 842 (2001).

[14] K. Chang, L.-Q. Chen, C. E. Krill, and N. Moelans, Effect of strong nonuniformity in grain boundary energy on 3-D grain growth behavior: A phase-field simulation study, Comput. Mater. Sci. **127**, 67 (2017).

[15] M. Verma and R. Mukherjee, Grain growth stagnation in solid state thin films: A phase-field study, J. Appl. Phys. **130**, 025305 (2021).

[16] D. Molnar, R. Mukherjee, A. Choudhury, A. Mora, P. Binkele, M. Selzer, B. Nestler, and S. Schmauder, Multiscale simulations on the coarsening of Cu-rich precipitates in α-Fe using kinetic Monte Carlo, molecular dynamics and phase-field simulations, Acta Mater. **60**, 6961 (2012).

[17] M. P. Gururajan and T. A. Abinandanan, Phase field study of precipitate rafting under a uniaxial stress, Acta Mater. **55**, 5015 (2007).

[18] R. Chafle, S. Bhowmick, and R. Mukherjee, Effect of co-existing external fields on a binary spinodal system: A phase-field study, J. Phys. Chem. Solids **132**, 236 (2019).

[19] S. Bhattacharyya and T. A. Abinandanan, A study of phase separation in ternary alloys, Bull. Mater. Sci. **26**, 193 (2003).

[20] H. Ramanarayan and T. A. Abinandanan, Spinodal decomposition in fine grained materials, Bull. Mater. Sci. **26**, 189 (2003).

[21] H. Abels, H. Garcke, and G. Grün, Thermodynamically consistent, frame indifferent diffuse interface models for incompressible two-phase flows with different densities, Math. Models Methods Appl. Sci. **22**, 1150013 (2012).

[22] H. Garcke, K. F. Lam, and A. Signori, On a phase field model of Cahn-Hilliard type for tumour growth with mechanical effects, Nonlinear Anal.: Real World Appl. **57**, 103192 (2021).

[23] J. E. Guyer, W. J. Boettinger, J. A. Warren, and G. B. McFadden, Phase field modeling of electrochemistry. I. Equilibrium, Phys. Rev. E **69**, 021603 (2004).

[24] D. J. Seol, S. Y. Hu, Y. L. Li, J. Shen, K. H. Oh, and L. Q. Chen, Computer simulation of spinodal decomposition in constrained films, Acta Mater. **51**, 5173 (2003).

[25] T. Muranushi, Paraiso: An automated tuning framework for explicit solvers of partial differential equations, Comput. Sci. Discov. **5**, 15003 (2012).

[26] K. Jiao, J. Xuan, Q. Du, Z. Bao, B. Xie, B. Wang, Y. Zhao, L. Fan, H. Wang, Z. Hou *et al.*, Designing the next generation of proton-exchange membrane fuel cells, Nature (London) **595**, 361 (2021).

[27] A. Hunter, F. Saied, C. Le, and M. Koslowski, Large-scale 3D phase field dislocation dynamics simulations on high-performance architectures, Int. J. High Perform. Comput. Appl. **25**, 223 (2011).

[28] A. Vondrous, M. Selzer, J. Hötzer, and B. Nestler, Parallel computing for phase-field models, Int. J. High Perform. Comput. Appl. **28**, 61 (2014).

[29] E. Miyoshi, T. Takaki, M. Ohno, Y. Shibuta, S. Sakane, T. Shimokawabe, and T. Aoki, Ultra-large-scale phase-field simulation study of ideal grain growth, npj Comput. Mater. **3**, 25 (2017).

[30] C. Hu, S. Martin, and R. Dingreville, Accelerating phase-field predictions via recurrent neural networks learning the microstructure evolution in latent space, Comput. Methods Appl. Mech. Eng. **397**, 115128 (2022).

[31] D. Montes de Oca Zapiain, J. A. Stewart, and R. Dingreville, Accelerating phase-field-based microstructure evolution predictions via surrogate models trained by machine learning methods, npj Comput. Mater. **7**, 3 (2021).

[32] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science **313**, 504 (2006).

[33] Y. Wang, H. Yao, and S. Zhao, Auto-encoder based dimensionality reduction, Neurocomputing **184**, 232 (2016).

[34] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, in *Advances in Neural Information Processing Systems*, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Curran Associates, Inc., Red Hook, 2015), Vol. 28.

[35] J. W. Cahn and J. E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, J. Chem. Phys. **28**, 258 (1958).

[36] J. W. Cahn, On spinodal decomposition, Acta Metall. **9**, 795 (1961).

[37] See Supplemental Material at https://link.aps.org/supplemental/10.1103/PhysRevMaterials.7.083802 for details on the phase-field model and its numerical implementation.

[38] L. Q. Chen and J. Shen, Applications of semi-implicit fourier-spectral method to phase field equations, Comput. Phys. Commun. **108**, 147 (1998).

[39] R. Bostanabad, Y. Zhang, X. Li, T. Kearney, L. C. Brinson, D. W. Apley, W. K. Liu, and W. Chen, Computational

microstructure characterization and reconstruction: Review of the state-of-the-art techniques, Prog. Mater. Sci. **95**, 1 (2018).

[40] H. Abdi and L. J. Williams, Principal component analysis, WIREs Comp. Stat. **2**, 433 (2010).

[41] M. M. Taye, Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions, Computation **11**, 52 (2023).

[42] S. Pfleiderer, D. Ball, and R. C. Bailey, AUTO: A computer program for the determination of the two-dimensional autocorrelation function of digital images, Comput. Geosci. **19**, 825 (1993).

[43] J. B. Tenenbaum, V. de Silva, and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science **290**, 2319 (2000).

[44] L. McInnes, J. Healy, and J. Melville, UMAP: Uniform manifold approximation and projection for dimension reduction, arXiv:1802.03426 (2020).

[45] L. Lu, R. Pestourie, S. G. Johnson, and G. Romano, Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport, Phys. Rev. Res. **4**, 023210 (2022).