Computational Materials Science 224 (2023) 112187

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

Full length article

Emulating microstructural evolution during spinodal decomposition using a tensor decomposed convolutional and recurrent neural network

Peichen Wu [a], Ashif Sikandar Iquebal [b], Kumar Ankit [a,*]

[a] Materials Science & Engineering, School for Engineering of Matter, Transport and Energy, Arizona State University, 551 E. Tyler Mall, Tempe AZ 85287, USA
[b] School of Computing and Augmented Intelligence, Arizona State University, 699 S. Mill Ave., Tempe AZ 85281, USA

ARTICLE INFO

Keywords:
Convolutional and recurrent neural network
Tensor decomposition
Phase-field
Microstructure evolution

ABSTRACT

Phase-field (PF) models are one of the most powerful tools to simulate microstructural evolution in metallic materials, polymers, and ceramics. However, existing PF approaches rely on rigorous mathematical model development, sophisticated numerical schemes, and high-performance computing for accuracy. Although recently developed surrogate microstructure models employ deep-learning techniques and reconstruction of microstructures from lower-dimensional data, their accuracy is fairly limited as spatiotemporal information is lost in the pursuit of dimensional reduction. Given these limitations, we present a novel data-driven emulator (DDE) for predicting microstructural evolution, which combines an image-based convolutional and recurrent neural network (CRNN) with tensor decomposition, while leveraging previously obtained PF datasets for training. To assess the robustness of DDE, we also compare the emulation sequence and the scaling behavior with phase-field simulations for several noisy initial states. Finally, we discuss the effectiveness of our microstructure emulation technique in the context of runtime speed-up while also highlighting its trade-off with accuracy.

1. Introduction

Phase-field methods are one of the most powerful and versatile tools to simulate microstructural evolution in metallic materials, polymers, and ceramics [1–4]. The popularity of the phase-field method's in modeling morphological evolution is due to the elegance with which it treats moving boundary problems by obviating the necessity to explicitly track the interfaces. However, they remain computationally intensive due to the strict limits on the maximum time and length scales imposed by the numerical methods [5]. For instance, the computational complexity associated with simulating spinodal decomposition [3,6] warrants the application of supercomputers for solving fourth-order Cahn–Hilliard partial differential equations (PDEs). The finite difference and the finite element schemes have been extensively employed, although there are challenges related to numerical implementation [7]. Various techniques have also been employed to shorten the simulation runtimes, such as adaptive mesh refinement [8,9], load balancing, [10] and random walker [11]. However, depending on the spatial and time resolution warranted, simulating the evolution of microstructures in three dimensions, which is a 4D problem, often spans from several hours to weeks on a supercomputing cluster. Current strategies to accelerate phase-field calculations rely on high-performance GPUs and power-intensive computational resources [12] as opposed to leveraging the past simulations to emulate new ones. As a result, a new simulation needs to be performed every time a simulation parameter is altered. Another limitation of the phase-field method is that it is not transferable across material systems without extensive parameter adaptation [13], even though the underlying phase transformation mechanism may be similar such as spinodally decomposing microstructures of polymers [14–16] and metallic materials [17–19], both of which entail up-hill diffusion of atoms. Merriman–Bence–Osher (MBO) threshold dynamics is another popular algorithm for simulating the mean curvature motion of interfaces [20,21]. However, this technique is only first order accurate in a two-phase setting while the accuracy is known to rapidly degrade in a multi-phase settings. A recent effort has rendered MBO dynamics to be second order accurate in problems limited to two-phases [22], however, there are currently no accurate schemes available for 3D multiphase problems.

In fact, the dilemma of efficiency, computation cost, and accuracy is not a new topic of interest within the materials modeling community. Parallel computation, which utilizes multiple CPUs or GPUs, is one of the ways by which simulation run-times can be reduced, as evidenced by its extensive use in molecular dynamics, phase-field, and finite element simulations [23]. However, the associated computation

* Correspondence to: 551 E. Tyler Mall, Tempe, AZ 85287, USA.
E-mail address: kumar.ankit@asu.edu (K. Ankit).

https://doi.org/10.1016/j.commatsci.2023.112187
Received 30 September 2022; Received in revised form 28 March 2023; Accepted 5 April 2023
Available online 12 April 2023

cost is still high and often requires powerful and expensive high-performance computing resources. Another pathway to dealing with this issue is the utilization of machine learning. Some of the recent examples include neural network-aided methodology to design modular metamaterials [24], potentials for molecular dynamics (DeePMD) [25], and Behler–Parrinello neural network (BPNN) [26]. Although the published literature is replete with techniques that focused on combining machine learning with molecular dynamics, only a handful of the works have focused on integrating machine learning with the phase-field methods [27–32]. These studies primarily rely on incorporating dimensional reduction, machine learning, and phase reconstruction, for microstructure emulations. For the dimensional reduction of microstructural images, Principle Component Analysis (PCA) is typically employed that has several limitations. Since PCA requires unfolding an image data into a one-dimensional array which causes loss of spatiotemporal information, its use is to be best avoided when dealing with image sequences for e.g. an evolving microstructure. This spatiotemporal loss manifests as poor reconstruction of microstructures. This is evident from the mismatch between the emulated and simulated spinodal microstructures reported by Zapiain et al. [27]. Another limitation is that the phase reconstruction adopted by those authors is not very effective in reconstructing the diffuse interfaces which are a characteristic of phase-field simulations. Although, reconstruction of interface may be not be an issue for a majority of solid-state transformations where the thickness of the interfaces in quite small (below 1 nm), the above noted limitation is particularly concerning for the case of solidification microstructures where the solid–liquid interface is known to be comparatively thicker. As compared to PCA, tensor decomposition is a better alternative since it does not lead to any spatiotemporal losses [33] although accurate emulations of microstructures may incur large computational costs.

The applications of Convolutional Recurrent Neural Networks (CRNN), which combines CNN (convolutional neural network) with RNN (Recurrent neural networks) have been limited to speech separation [34] and activity recognition [35], to date. However, its utility in materials science, particularly in the context of microstructure modeling, has never been previously explored. Therefore, in this work, we report a novel Data-driven Emulator (DDE) using a tensor decomposed convolutional and recurrent neural network (CRNN) to circumvent aforementioned issues [36–38], which couples machine learning with tensor decomposition (TD) and microstructure modeling to enable a high-throughput and accurate prediction for microstructure evolution. The developed architecture, as described below, leverages CNN which extracts image features and learns their spatial relationships within each image, while the coupled RNN models the temporal relationship between the subsequent images. For training and validating this algorithm, we utilize the phase-field generated datasets of spinodally decomposing microstructures.

This paper is organized as follows: In Section 2, the Cahn–Hilliard phase-field and the DDE models are explained. Section 3 compares and validates the DDE predictions with the corresponding phase-field images. Here, we also discuss the trade-off between training efficiency and prediction accuracy. Section 4 concludes this article.

2. Methods

2.1. Phase-field (PF) model

For the sake of completeness, the well-known Cahn–Hilliard model, which is used to generate microstructure training datasets is briefly outlined in this section. Our diffuse interface approach for modeling the phase separation in binary immiscible alloy (A-B) films adopts a free energy functional consisting of distinct bulk and interfacial energy terms [39], written as

$$F = \int_{\Omega} \left[f(\phi) + \frac{1}{2}\epsilon^2 |\nabla\phi|^2 \right] d\Omega. \tag{1}$$

Here, the order parameter, $\phi(x,t)$, denotes the scaled concentration assumed to be a continuous function of position and time, ranging from equilibrium concentration of 0 at the A-rich β phase to 1 at the B-rich α phase. The bulk free energy density, $f(\phi)$, is given by

$$f(\phi) = \frac{1}{4} W \phi^2 (1-\phi)^2 \tag{2}$$

where W is the well height that puts an energy penalty to all the states other than 0 and 1. ϵ is the gradient free energy coefficient, which penalizes large gradients in the order parameter giving rise to the diffuse nature of phase boundaries. At equilibrium, the interfacial width, δ, and the interfacial energy, γ, is governed by an interplay of the two terms in the free energy functional and scales as $\delta \propto \sqrt{\epsilon^2/W}$ and $\gamma \propto \sqrt{\epsilon^2 W}$.

The kinetics of a phase-separating system can be simulated by solving the Cahn–Hilliard equation,

$$\frac{\partial \phi}{\partial t} = \nabla \cdot M \nabla \mu \tag{3}$$

where, μ is the chemical potential derived from the variational derivative of the free energy functional as $\mu = \frac{\delta F}{\delta \phi}$. M is the mobility of the diffusing species which can be related to the diffusion coefficient as $D = M \frac{\partial^2 f}{\partial \phi^2}$, assumed to be 1.0. In the present study, we assume the kinetic parameter to be independent of the order parameter, thus, the evolution equation may be rewritten as,

$$\frac{\partial \phi}{\partial t} = M \nabla^2 \frac{\delta F}{\delta \phi}. \tag{4}$$

Using Eqs. (1) and (2), we arrive at the final form of the Cahn–Hilliard equation,

$$\frac{\partial \phi}{\partial t} = M \nabla^2 \left[\frac{1}{2} W (2\phi^3 - 3\phi^2 + \phi) - \epsilon^2 \nabla^2 \phi \right]. \tag{5}$$

We non-dimensionalize Eq. (5) by using reduced variables which are defined as: $x^* = x/\Delta x$, $M^* = M/(M_0 k_B T)$, $\nabla^* = (\Delta x)^2 \nabla$, $W^* = W/(k_B T)$, $\epsilon^* = \epsilon/(\Delta x \sqrt{k_B T})$, and $t^* = M_0 t/(\Delta x)^2$, where Δx is the grid spacing, M_0 is an arbitrarily defined constant bulk mobility that is dependent on the temperature, T, and k_B is the Boltzmann constant. The final dimensionless form is given by

$$\frac{\partial \phi(x^*, t^*)}{\partial t^*} = M^* (\nabla^*)^2 \left\{ \frac{1}{2} W^* (2\phi^3 - 3\phi^2 + \phi) - (\epsilon^*)^2 (\nabla^*)^2 \phi \right\}. \tag{6}$$

We solve Eq. (3) via an explicit finite difference method (FDM) on a regular square mesh. Alternatively, one could implement a forward difference Euler scheme for the temporal derivatives and a second-order central difference for the spatial derivatives. The Cahn–Hilliard Eq. (6) is solved in two steps. First, the chemical potential μ is calculated at each grid point, while the Laplacian is calculated using a 7-point stencil. The discretized equation required to compute μ is given by

$$\mu^t_{i,j,k} = \frac{1}{2} W \phi^t_{i,j,k} (1 - \phi^t_{i,j,k})(1 - 2\phi^t_{i,j,k}) - \epsilon^2 \frac{\phi^t_{i+1,j,k} + \phi^t_{i-1,j,k} + \phi^t_{i,j+1,k} + \phi^t_{i,j-1,k} + \phi^t_{i,j,k+1} + \phi^t_{i,j,k-1} - 6\phi^t_{i,j,k}}{h^2}. \tag{7}$$

The Laplacian of μ is determined by another central difference scheme such that the concentration evolution equation in the discretized takes the following form,

$$\frac{\phi^{t+\Delta t}_{i,j,k} - \phi^t_{i,j,k}}{\Delta t} = M \left\{ \frac{\mu^t_{i+1,j,k} + \mu^t_{i-1,j,k} + \mu^t_{i,j+1,k} + \mu^t_{i,j-1,k} + \mu^t_{i,j,k+1} + \mu^t_{i,j,k-1} - 6\mu^t_{i,j,k}}{h^2} \right\}. \tag{8}$$

The square grid dimensions are given by $\Delta x = \Delta y = 0.3$, while the timestep is defined by $\Delta t = 0.0001$. Periodic boundary conditions are imposed along the x- and the y-directions. Non-dimensional $M^* = 1.0$ allows the phase separating microstructures to coarsen within a reasonable timeframe so as to limit the training time to below 1.5 h, while

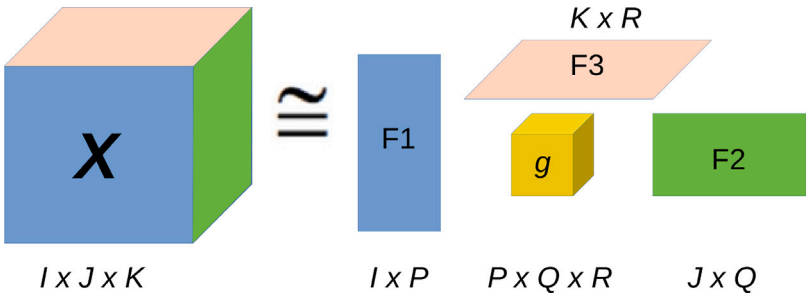

Fig. 1. Illustration of a Tucker decomposition, in which a tensor X can be decomposed as a core tensor g and factor matrices, $F1$, $F2$, and $F3$, i.e. one for every mode.

$W *= 4.0$ and $\epsilon^* = 0.051$ ensure retention of equimolar composition as the diffuse phase interfaces of width, 6 grid points ($6\Delta x$), develops. The initial condition used for generating the phase-field training data comprises an equimolar alloy composition with fluctuations, the amplitude of which varies in the range of 0.01%–0.50%. Since these composition fluctuations are seeded randomly, distinct microstructural evolution sequences are obtained from every simulation run.

2.2. Data-driven emulator (DDE)

2.2.1. Neural network architecture

The data-driven emulator or the DDE proposed in this work is based on a CRNN framework that employs a tensor decomposed convolutional neural network (CNN) for image feature extraction which is coupled to a recurrent neural network (RNN). The latter enables image sequence prediction based on the series of features extracted over time using CNN. Here, features refer to the line and curvature within the simulated micrographs generated from phase-field simulations. TD entails decomposition of a 2D convolutional layer matrix into several smaller layers. Although the number of layers as a result of decomposition increases, the total number of floating-point operations and weights will be smaller than the parent layer. The input and output tensor dimensions of a tensor decomposed convolutional layer will be the same as a regular convolutional layer.

In this work, we have employed the most commonly used tensor decomposition approach known as tucker decomposition [38] which is shown in Fig. 1. Fig. 2 shows the architecture of the CRNN used in this work and the tensor-decomposed convolutional layer, which is referred to as the convolutional layer hereafter. The input to the layer comprises a series of images extracted from phase-field, that are processed through the convolutional and max-pooling layers of the CNN. Since we transfer the features extracted from CNN to RNN, the fully-connected layers of the CNN are removed. The convolutional layer of CNN scans using a 2×2 filter with strides of 1×1 along the image length and the width. The convolutional layer produces a different abstraction of the input image by refreshing the layer after every pass. Within this framework, the convolutional layer, in parallel, generates 256 filters for every input image. To train the CRNN, we employ a sequence of microstructure images which are simulated by phase-field simulations. For every such dataset which corresponds to a set of microstructures obtained at regular time intervals, the total time is chosen to be 3000 additional timesteps beyond the end of transient state i.e. when average domain size starts to scale linearly.

In the current study, we generated datasets of size $32\Delta x \times 32\Delta y$ as well as $128\Delta x \times 128\Delta y$ to train the DDE and assess its performance w.r.t. finite difference solver as the domain size increases. For a digital micrograph of $32\Delta x \times 32\Delta y$, no down-sampling is performed, as opposed to higher resolution data where the micrograph may need to be down-sampled. Therefore, for an input image of dimension $32\Delta x \times 32\Delta y$, the input to the convolutional layer is a tensor of $10 \times 32 \times 32$, while the output has dimension of $10 \times 32 \times 32 \times 256$. This is represented as "Tensor decomposed Conv2d-1" in the CRNN architecture (see Fig. 1). In this algorithmic sequence, the convolutional layer is followed by a max-pooling layer which is responsible for downsampling the output produced from the previous layer. A pooling operation is performed to reduce the CNN framework's computational cost, which causes dimensional reduction from $10 \times 32 \times 32 \times 256$ to $10 \times 16 \times 16 \times 256$. A combination of convolutional and max-pooling layers is useful in extracting the low-level features, such as the presence of phase boundaries. We additionally include convolutional "Tensor decomposed Conv2d-2" and max-pooling layers to extract higher-level features, such as phase boundary curvature, that enable complete feature extraction and training. Also, the padding for all the convolutional and max-pooling layers ensures that the convolved features retain their dimensions for an entire sequence.

It is relatively well-known that CNN is not very effective in handling sequential data within a time series. Therefore, to emulate microstructural evolution, we have combined it with a deep-learning approach. The output from CNN is fed into a recurrent neural network (RNN) – a deep learning approach for handling sequential data – to capture the temporal relationships within image sequences. RNN comprises a hidden state containing and transmitting the information from the previous timesteps, acting similar to a data buffer [40,41]. Traditional RNNs are known to be inefficient in capturing the temporal information for long sequences, particularly due to the vanishing gradient problem where information from past time steps decreases exponentially. To circumvent this issue, Long Short Term Memory (LSTM), which is a variant of RNN, is incorporated into the present workflow. A detailed description of the LSTM architecture has previously been reported in the literature [42].

Within the proposed workflow shown in Fig. 2, the LSTM layer will output a tensor of size 10 times the total number of LSTM's repeat modules. Next, the dense layer will output 10×1024 values for every pixel point. LSTM's repeat modules consist of hidden and cell states, while the hidden state carries information from immediately previous events, the cell state stores and loads information of penultimate events, which prevent the LSTM gradients from vanishing. Finally, the 10×1024 output matrix is converted to the original input size of $10 \times 32 \times 32 \times 1$ before comparing with the input tensor using a mean square error (MSE) function given by

$$\mathrm{MSE} = 1/n \sum_{i}^{n} (Y_i - \bar{Y}_i)^2, \tag{9}$$

where $\bar{Y}_i$ is the prediction value, Y_i is the actual value, and n is the total number of predictions.

For microstructure emulation using CRNN, there are several parameters and hyperparameters that need to be tuned, which requires prior experience and rounds of trial. Therefore, in the present work, we incorporated 32, 64, 128, 256, or 512 filters without altering other parameters and hyperparameters to optimize the losses in the validation database.

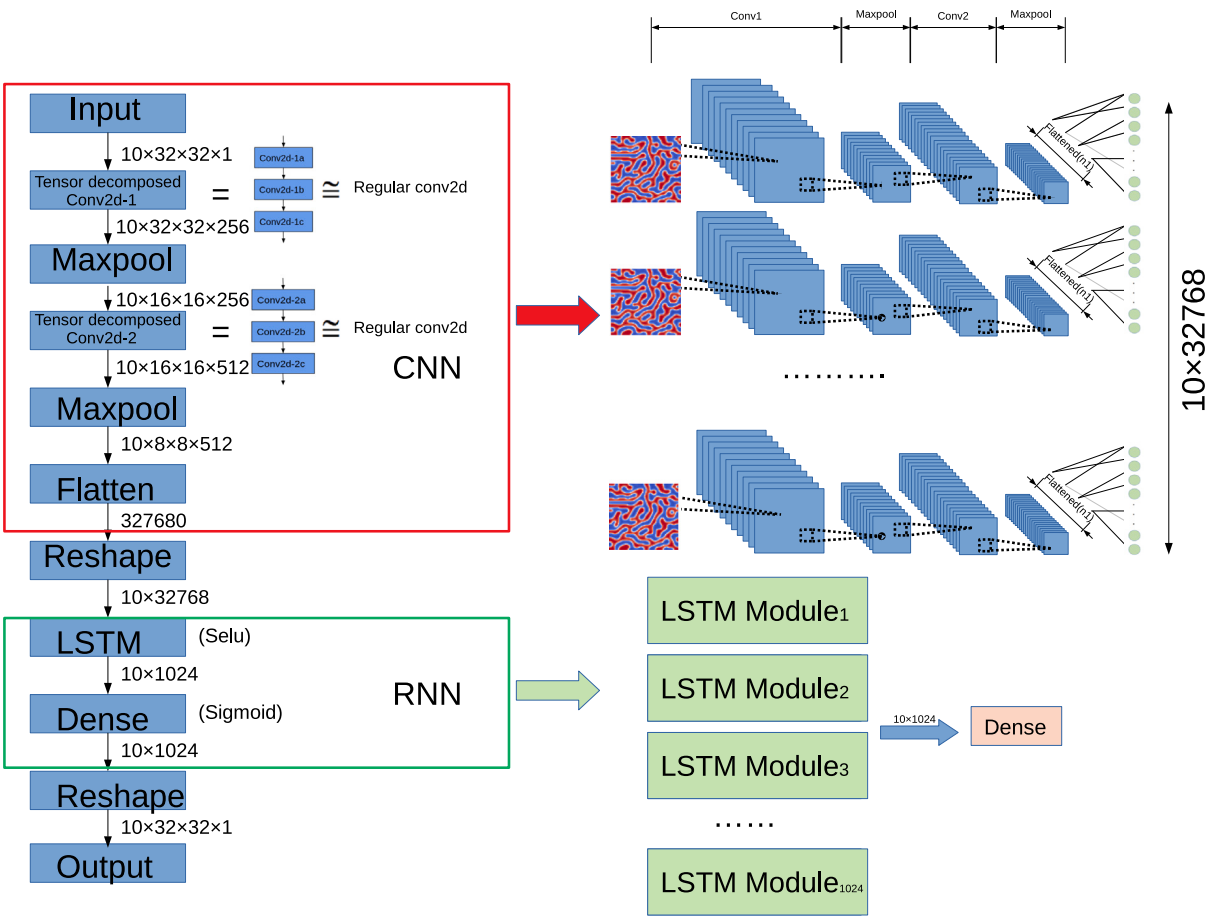

Fig. 2. CRNN layer structure and the corresponding tensor dimension. The activation function and padding type are also indicated.

2.2.2. Dataset pre-processing

Using the phase-field constitutive equations outlined in Section 2.1, we first generate the training, validation, and test datasets that are required to train and validate the proposed CRNN-based algorithm. The validation dataset is used to tune the neural network while maintaining the associated losses to a minimum during this process. After achieving the lowest validation loss, CRNN outputs an image corresponding to every new microstructural data in the test set, which does not containing any dataset that was used for training or validation. Fig. 3(a) shows an exemplary training dataset which comprises a series of 40 snapshots of phase separating microstructures obtained every 1000 timesteps using the phase-field method. As shown, the first ten images represent the first training sample, the next 10, the second sample, and so on. Therefore, the output corresponding to the first training sample is overall the eleventh image in sequence. Thus, we have a total of 30 training samples and 30 responses from every image sequence. We generate 500 such image sequences using different initialization of the phase-field i.e. seed number, with other material-specific parameters consistent, such that we have a total of 15,000 training samples. In a similar fashion, we generate a validation dataset with 300 validation samples.

In total, we obtain 10 testing datasets, one of which is shown in Fig. 3(b). The first ten phase-field images are used as input to the DDE for predicting the microstructure corresponding to $t = 11$. Then the phase-field images from $t \in \{2, 10\}$ in addition to the predicted image are rendered as the second input in sequence for predicting the microstructure corresponding to the $t = 12$. In this manner, we continue to predict the microstructures sequentially until $t = 40$.

The phase-field model that was used to generate microstructures for the training dataset are coded in C++. The DDE model is constructed using Keras library in Anaconda Python 3. In the spirit of establishing a fair comparison between the DDE and phase-field method, both the codes are serially-coded or executed on a single CPU running Linux OS.

3. Results and discussion

In this section, we discuss the DDE predictions corresponding to domains of size, $32\Delta x \times 32\Delta y$ and $128\Delta x \times 128\Delta y$, while highlighting the trade-off between accuracy and training efficiency of the novel CRNN approach for predicting microstructural evolution. Fig. 4 compares the simulated temporal evolution of microstructures of size $32\Delta x \times 32\Delta y$, with the emulated ones at the representative timesteps, starting from distinct seed numbers. The adjacent plots compare the simulated with emulated average domain sizes computed based on the scaling approach reported in the literature [43,44]. At steady state, the average domain size of microstructures undergoing spinodal decomposition scales as

$$S = At^n \tag{10}$$

where, S is the domain size, n is the power exponent, and A is the pre-factor. The same approach has been traditionally used to obtain the scaling behavior of phase separating microstructures, one of the more recent examples being that of vapor co-deposited alloy films [45,46]. It is found that when the volume of the training database equals 500 image series, the emulated microstructure although visually comparable shows minor discrepancies with respect to phase-field results, the most notable being the thinning of the vertical ligament in Fig. 4d. Similar discrepancies can also be noted in Fig. 4f, where the globule spanning the right edge does not pinch off unlike in the corresponding phase-field

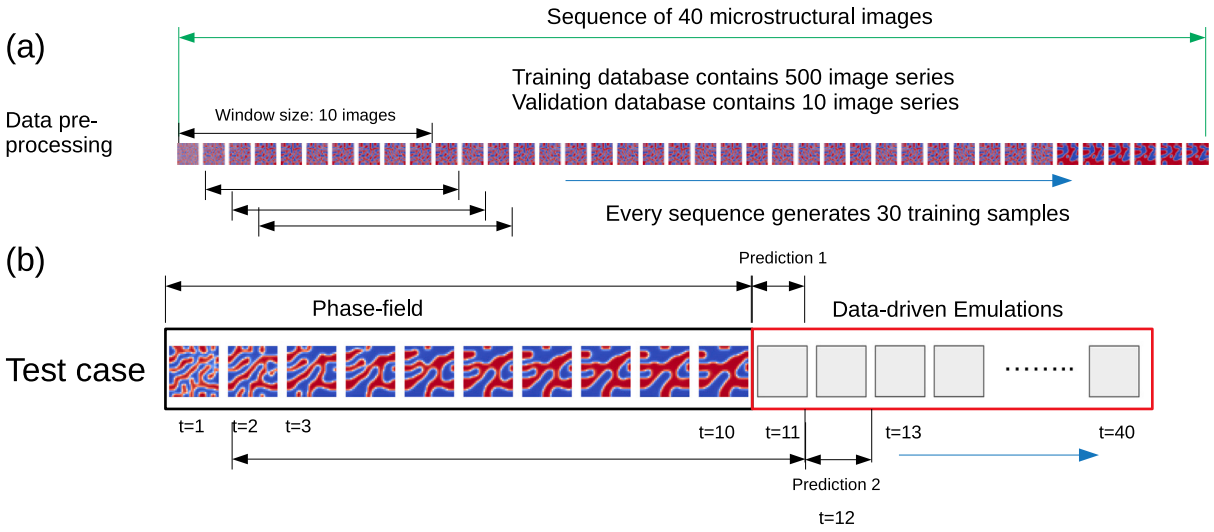

Fig. 3. (a) Structure of the training and validation databases. (b) Schematic diagram showing our data-driven emulation methodology.

simulated evolution. However, in either case, we were able to correctly emulate the microstructural evolution by doubling the training database volume. Based on a favorable comparison of scaling dynamics as well as the morphological evolution in Fig. 4, we conclude that the accuracy of the CRNN-based emulation approach is independent of the initial condition. At this juncture, we highlight relevant differences between our approach with respect to recently reported deep learning approaches [27–29]. For instance, Yang et al. [29] implemented an RNN with eidetic 3D LSTM cells to predict the evolution pattern in four different evolution phenomena including plane-wave propagation, grain coarsening, spinodal decomposition, and dendrite growth. Their methodology involved training the RNN with microstructure sequences followed by predicting the microstructure evolution for a new initial microstructure. In another work, Zapiain et al. [27] implemented a LSTM network to predict the microstructure evolution by obtaining a low-dimensional representation using two-point statistics and principal component analysis. While both these previous approaches have demonstrated the viability of using deep learning for microstructure prediction, their application is limited by the following factors: first, deep learning approaches are known to be data-hungry and therefore their performance is highly contingent on the availability of large datasets. For instance, the LSTM network trained by Zapiain et al. [27] warranted 5000 high-fidelity phase-field simulations, each of which comprised 60 sequential microstructure datasets. Another limitation of the above approaches is the use of principal component analysis (PCA) for dimensionality reduction that has limited applicability in emulating microstructure sequences, since it entails unfolding images into one-dimensional arrays. In this operation, the spatiotemporal information related to phase boundaries, particularly the width and the curvature, is lost which ultimately manifests as poor reconstruction of emulated microstructures [27,28]. We emphasize that unlike the work of Zapiain et al. [27], where the microstructure prediction is limited to the final timestep, the CRNN approach reported here has predicted the complete microstructure evolution sequence with a reasonable accuracy. Nonetheless, one of the limitations of our approach is that minor errors are accumulated as the microstructure prediction is extrapolated for future time steps. This is an expected outcome since we have utilized the phase-field generated microstructures from the first 10 time steps to emulate microstructural evolution for the subsequent 30 time steps.

The microstructural evolution emulated using the DDE in a domain of size, $128\Delta x \times 128\Delta y$ is found to be visually comparable to the corresponding phase-field simulations, as observed in Fig. 5, and exemplified by the disappearance of the globular phase separating domains within the encircled region. To further validate the DDE in accordance with respect to the benchmark problems reported by Jokisaari et al. [47], we estimated the total free energy from the emulated micrographs. Depending on the value of the order parameter specific to every grid point in the emulated microstructure, the corresponding bulk and the interfacial energy contributions are added to obtain the total free energy for the entire microstructure. The temporal evolution trend of the emulated free energy minimization shows reasonable agreement with the phase-field free energy minimization as shown in Fig. 6. Minor discrepancies in emulations of the 1-D interface profiles of ϕ are also observed when the training database volume is smaller. However, the emulated interface profile is found to better converge with the phase-field diffuse interface when the training volume is doubled, as shown in Fig. 7. Both these findings indicate the importance of selecting a large training database that ultimately facilitates accuracy in microstructure emulations.

An another advantage of DDE is the minuscule runtimes as compared to phase-field which are tabulated in Table 1.

This acceleration in run time comes at the cost of accuracy which is quantified by a metric that measures the average gain per epoch

$$\text{Average gain} = \frac{t - t_{TD}}{t} \times 100, \tag{11}$$

where t refers to the time needed for running one epoch without applying tensor decomposition while t_{TD} corresponds to the runtimes when tensor decomposition is incorporated. In the present context, the epoch refers to the number of times that datasets pass in the forward or in the backward directions per the workflow shown in Fig. 2. Corresponding gains in runtime, the MSE, and the overall emulation runtimes are listed in Table 1 for computational domains of size, 32×32 and 128×128 grid points. We note that the MSE and the prediction times increase with the computational domain size, since the latter entails handling a larger input variable matrix that increases the processing time. Clearly, a trade-off between prediction accuracy and overall runtime exists in order to predict the microstructural evolution within a reasonable timeframe. Our current technique of accelerating the CNN by deploying tensor decomposition, although novel in the context of emulating microstructural evolution, is essentially motivated from previous findings where more than eight times improvement in the training efficiency could be achieved without any significant decrease in accuracy [49]. One of the ways by which the performance of the data-driven emulator can be further improved is by lowering the rank of tensor decomposition based on the prediction accuracy threshold. Image downsampling, which is known to be compatible with CRNN approaches, can also be employed for this purpose.

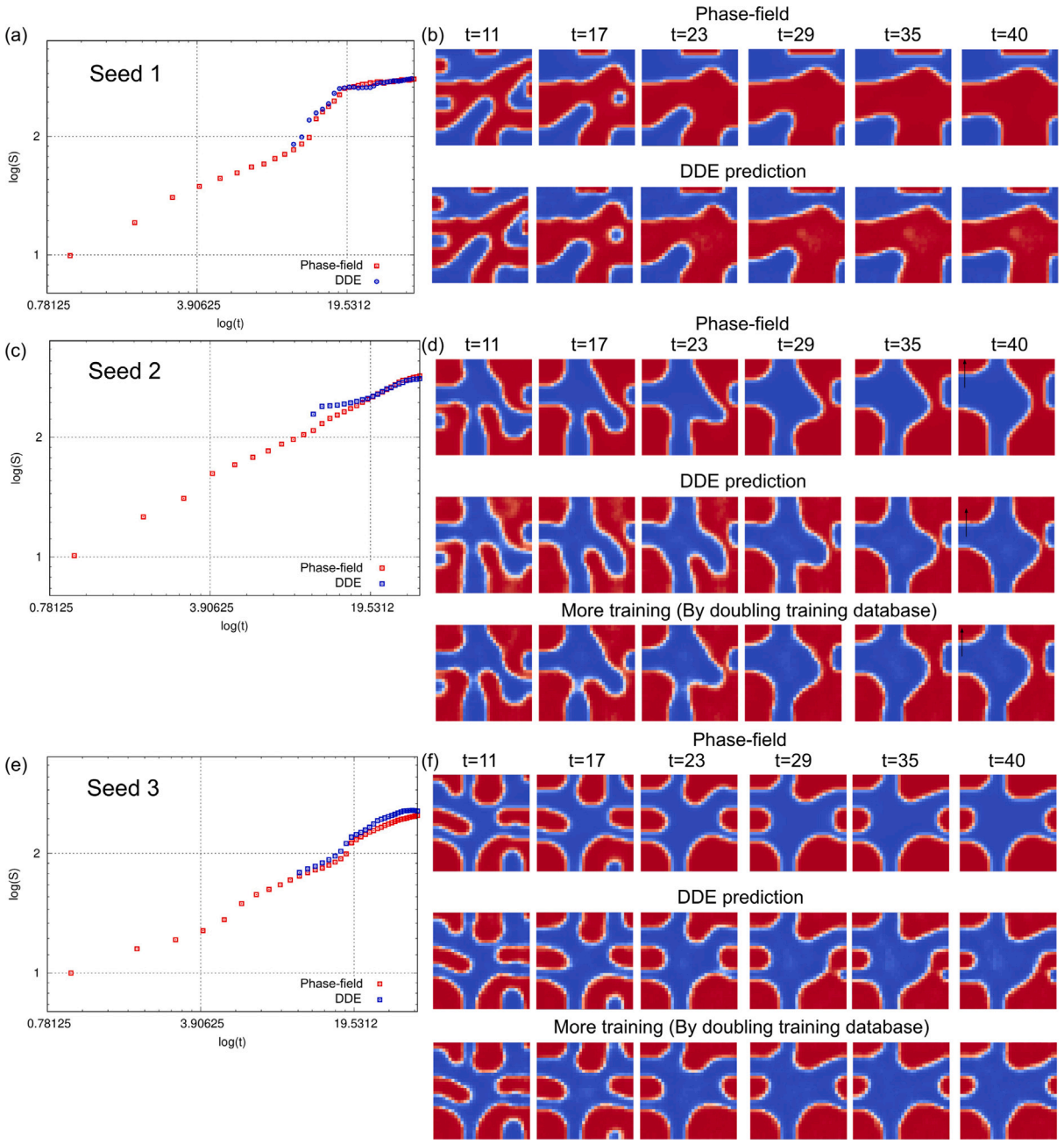

Fig. 4. Comparing the simulated and the emulated evolution of microstructural evolution at representative timesteps for distinct initial conditions generated by random seeds, as discussed in Section 2.1. The scaling of the domain size which is obtained by both these techniques is compared in the adjacent plots. Doubling the training volume increases the accuracy of emulations as evident from (d) and (f). '1x training' refers to training dataset comprising of 500 phase-field simulations whereas '2x training' implies twice as much i.e. 1000. The black arrows in (d) indicate the regions where the 1D diffuse interface profiles shown in Fig. 7 were plotted.

Table 1
Comparison of the computation costs associated with phase-field calculations employing finite difference and Fourier spectral solvers (for numerical scheme, refer [48]) with DDE. Training efficiency improved by tensor decomposition (TD) per epoch is also listed. CPU runtime improvement is calculated as the time used for generating the training data×3600/24. An improvement of 62.55 implies that DDE is around 63 times faster than the finite difference solver.

Simulation domain size	Numerical scheme	Total number of image series	Total number of training samples	Time used for generating training data	Training time without TD	Average gain for each epoch with TD	Average loss of validation set (MSE)	DDE prediction time	CPU runtime improvement
32 × 32	Finite difference	500	15000	0.417 h	13.42 h	44.78%	0.0056	24 s	62.55
32 × 32	2nd order Fourier spectral	500	15000	0.549 h	13.13 h	43.67%	0.0034	22 s	89.84
128 × 128	Finite difference	400	3000	1.321 h	288.8 h	26.98%	0.0725	79 s	60.20

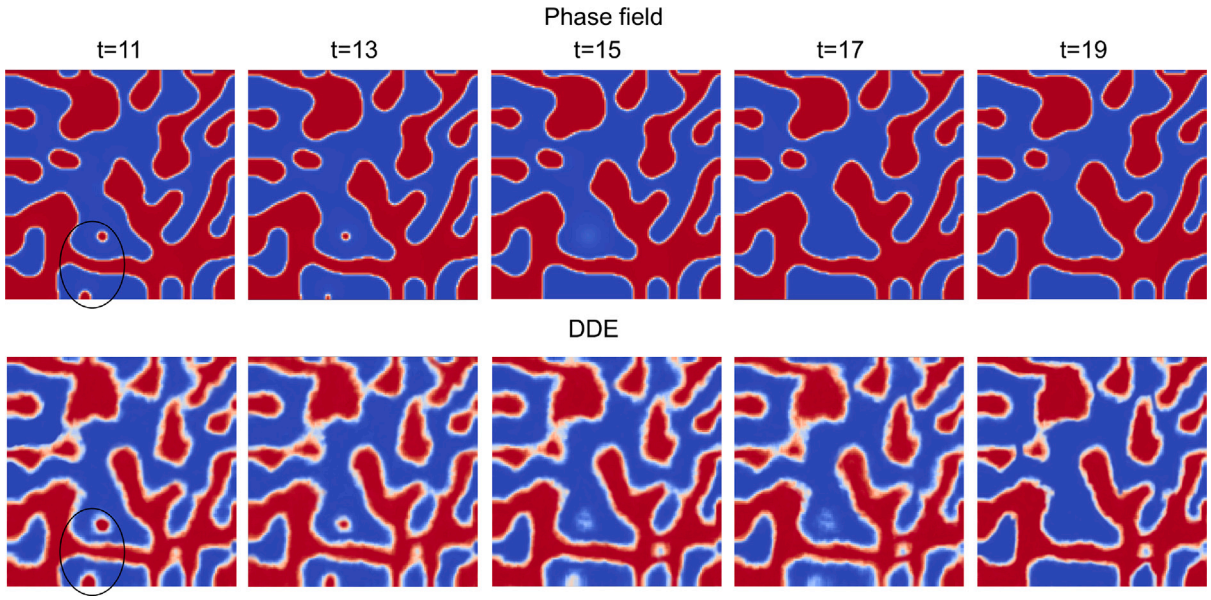

Fig. 5. The comparison between the phase-field simulation and the data-driven emulation of microstructural evolution at representative timesteps for a simulation box size of size 128 × 128 grid points. Encircled regions compare an instance when the disappearance of globular features seen in phase-field simulations are satisfactorily emulated.

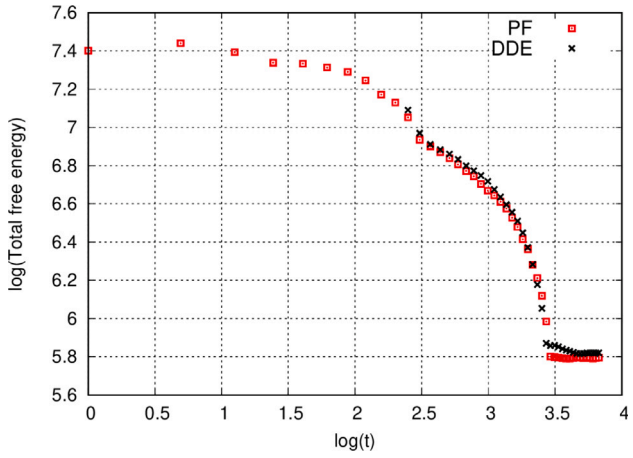

Fig. 6. Comparison of the temporal minimization of the total free energy obtained from phase-field simulations and DDE. Both trends conform with the benchmark laid by Jokisaari et al. [47]. The total free energy for DDE is estimated from the emulated microstructures by accounting for the bulk and interfacial energy contributions specific to every grid point depending on the value of the order parameter.

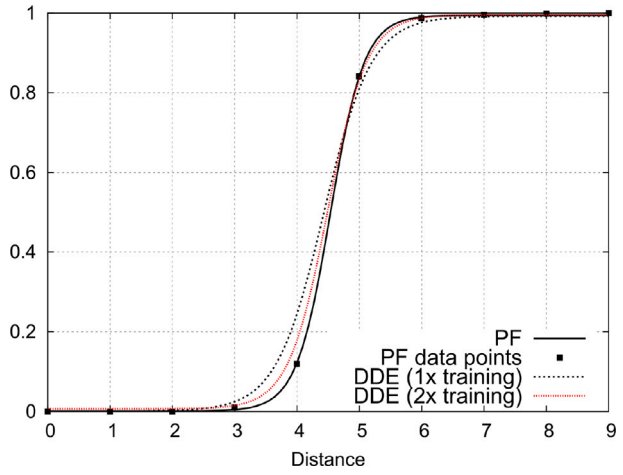

Fig. 7. Comparing the simulated and the emulated 1-D interface profiles of the conserved order parameter, ϕ, plotted along the black arrow (at t=40) in Fig. 4d.

Finally, our study does not discount the importance of phase-field methodology which still remains the best technique for simulating microstructure evolution. Machine learning based surrogate models of microstructure evolution would more often than not completely rely on phase-field models for the generation of training datasets, therefore, mastering the latter which entails acquiring well-rounded knowledge of materials thermodynamics and kinetics, computer programming, and numerical techniques, remains crucial and irreplaceable.

4. Conclusions

In conclusion, we have introduced a novel CRNN-based approach for emulating microstructural evolution that accompanies spinodal decomposition. Our data-driven approach, which leverages phase-field generated microstructure datasets as input, combines tensor decomposition and deep learning to predict the morphological evolution efficiently. Based on a systematic study, we found that a fully trained DDE emulator can predict the microstructural evolution 60 times faster when compared to the phase-field method.

The convolution and recurrent neural network (CRNN) follows an image-based, nonparametric algorithm, which is capable of extracting microstructural features and their evolution pattern from phase-field simulated sequences. The proposed CRNN is material as well as process agnostic, and therefore could potentially be used for predicting microstructural evolution in distinct scenarios as long as the training data is available. Unlike previous studies [27–29], the present DDE model, in addition to reproducing the correct scaling dynamics and the morphological evolution, conserves the characteristic diffuse nature of the interfaces. However, in order to reproduce the smooth hyperbolic tangent-like interface profiles which the phase-field predicts, the training volume needs to be increased. The time required for a trained DDE to predict microstructural evolution typically spans a minute, which is minuscule when compared to phase-field simulation run times.

While our work is not the first one to develop a data-driven simulator for phase field methods, our novel implementation of tensor decomposition into CNN instead of using low-dimensional projections

of two-point correlation functions [27,28], accelerates the neural network without significant accuracy losses. In closing, we would like to emphasize that the development of a microstructure emulation method such as the one reported here is not aimed at replacing the phase field models. Rather it is meant to aid in scenarios where obtaining the solution of partial differential equations is non-trivial, or in those cases where the complexities related to multiphysics make the formulation of phase field models, difficult. In numerous scenarios, phase-field models could be the only means for generating the training data for DDE. Apparently, the present emulation technique heavily relies on the development of phase-field models without which the former cannot be applied.

It is also equally important to note that DDE, unlike numerical schemes, does not involve solving any partial differential equations. It is agnostic of the complex multiphysics associated with phase transformations as the morphological evolution that is emulated is entirely based on image processing and machine learning. Therefore, the full potential of DDE lies in scenarios where phase-field models that could possibly encapsulate all the relevant physics are yet to be developed, for e.g., additive manufacturing. In such cases, training datasets for DDE can be generated using in-situ experiments.

CRediT authorship contribution statement

Peichen Wu: Methodology, Investigation, Writing – original draft, Validation. **Ashif Sikandar Iquebal:** Conceptualization, Investigation, Writing – original draft. **Kumar Ankit:** Conceptualization, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Kumar Ankit reports financial support was provided by National Science Foundation.

Data availability

Data will be made available on request.

Acknowledgments

KA acknowledges funding from the National Science Foundation (NSF) under Grant Nos. CMMI-1763128 (Drs. Alexis Lewis and Thomas Kuech, Program Managers) and CAREER-2145812 (Dr. Jonathan Madison, Program Manager).

References

[1] W.J. Boettinger, J.A. Warren, C. Beckermann, A. Karma, Phase-field simulation of solidification, Annu. Rev. Mater. Res. 32 (1) (2002) 163–194, http://dx.doi.org/10.1146/annurev.matsci.32.101901.155803.

[2] L.-Q. Chen, Phase-field models for microstructure evolution, Annu. Rev. Mater. Res. 32 (1) (2002) 113–140, http://dx.doi.org/10.1146/annurev.matsci.32.112001.132041.

[3] N. Moelans, B. Blanpain, P. Wollants, An introduction to phase-field modeling of microstructure evolution, CALPHAD 32 (2) (2008) 268–294, http://dx.doi.org/10.1016/j.calphad.2007.11.003.

[4] I. Singer-Loginova, H. Singer, The phase field technique for modeling multiphase materials, Rep. Progr. Phys. 71 (10) (2008) 106501, http://dx.doi.org/10.1088/0034-4885/71/10/106501.

[5] E.J. Kautz, Predicting material microstructure evolution via data-driven machine learning, Patterns (2021) 100285, http://dx.doi.org/10.1016/j.patter.2021.100285.

[6] N. Provatas, K. Elder, Phase-Field Methods in Materials Science and Engineering, Wiley, 2010, http://dx.doi.org/10.1002/9783527631520.

[7] K. Cheng, W. Feng, C. Wang, S. Wise, An energy stable fourth order finite difference scheme for the Cahn–Hilliard equation, J. Comput. Appl. Math. 362 (2019) 574–595, http://dx.doi.org/10.1016/j.cam.2018.05.039.

[8] N. Provatas, N. Goldenfeld, J. Dantzig, Efficient computation of dendritic microstructures using adaptive mesh refinement, Phys. Rev. Lett. 80 (1998) 3308–3311, http://dx.doi.org/10.1103/PhysRevLett.80.3308.

[9] M. Greenwood, K. Shampur, N. Ofori-Opoku, T. Pinomaa, L. Wang, S. Gurevich, N. Provatas, Quantitative 3D phase field modelling of solidification using next-generation adaptive mesh refinement, Comput. Mater. Sci. 142 (2018) 153–171, http://dx.doi.org/10.1016/j.commatsci.2017.09.029.

[10] W.L. George, J.A. Warren, A parallel 3D dendritic growth simulator using the phase-field method, J. Comput. Phys. 177 (2) (2002) 264–283, http://dx.doi.org/10.1006/jcph.2002.7005.

[11] M. Plapp, A. Karma, Multiscale random-walk algorithm for simulating interfacial pattern formation, Phys. Rev. Lett. 84 (8) (2000) 1740, http://dx.doi.org/10.1103/PhysRevLett.84.1740.

[12] T. Shimokawabe, T. Aoki, T. Takaki, T. Endo, A. Yamanaka, N. Maruyama, A. Nukada, S. Matsuoka, Peta-scale phase-field simulation for dendritic solidification on the TSUBAME 2.0 supercomputer, in: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, 2011, pp. 1–11, http://dx.doi.org/10.1145/2063384.2063388.

[13] S. DeWitt, S. Rudraraju, D. Montiel, W.B. Andrews, K. Thornton, PRISMS-PF: A general framework for phase-field modeling with a matrix-free finite element method, NPJ Comput. Mater. 6 (1) (2020) 1–12, http://dx.doi.org/10.1038/s41524-020-0298-5.

[14] S. Glotzer, Computer simulations of spinodal decomposition in polymer blends, Annu. Rev. Comput. Phys. II (1995) 1–46, http://dx.doi.org/10.1142/9789812831149_0001.

[15] F. Bruder, R. Brenn, Spinodal decomposition in thin films of a polymer blend, Phys. Rev. Lett. 69 (4) (1992) 624, http://dx.doi.org/10.1103/PhysRevLett.69.624.

[16] A. Mukherjee, R. Mukherjee, K. Ankit, A. Bhattacharya, B. Nestler, Influence of substrate interaction and confinement on electric-field-induced transition in symmetric block-copolymer thin films, Phys. Rev. E 93 (2016) 032504, http://dx.doi.org/10.1103/PhysRevE.93.032504.

[17] K. Rundman, J. Hilliard, Early stages of spinodal decomposition in an aluminum-zinc alloy, Acta Metall. 15 (6) (1967) 1025–1033, http://dx.doi.org/10.1016/0001-6160(67)90266-0.

[18] J. Langer, Theory of spinodal decomposition in alloys, Ann. Phys. 65 (1) (1971) 53–86, http://dx.doi.org/10.1016/0003-4916(71)90162-X.

[19] M. Miller, J. Hyde, M. Hetherington, A. Cerezo, G. Smith, C. Elliott, Spinodal decomposition in Fe–Cr alloys: Experimental study at the atomic level and comparison with computer models–I. Introduction and methodology, Acta Metall. Mater. 43 (9) (1995) 3385–3401, http://dx.doi.org/10.1016/0956-7151(95)00040-3.

[20] B. Merriman, J. Bence, S. Osher, Diffusion Generated Motion by Mean Curvature, Department of Mathematics, University of California, Los Angeles, 1992.

[21] B. Merriman, J.K. Bence, S.J. Osher, Motion of multiple junctions: A level set approach, J. Comput. Phys. 112 (2) (1994) 334–363.

[22] A. Zaitzeff, S. Esedoḡlu, K. Garikipati, Second order threshold dynamics schemes for two phase motion by mean curvature, J. Comput. Phys. 410 (2020) 109404.

[23] E.N. Millán, C.A. Ruestes, N. Wolovick, E.M. Bringa, Boosting materials science simulations by high performance computing, Mec. Comput. 35 (10) (2017) 467–482.

[24] L. Wu, L. Liu, Y. Wang, Z. Zhai, H. Zhuang, D. Krishnaraju, Q. Wang, H. Jiang, A machine learning-based method to design modular metamaterials, Extreme Mech. Lett. 36 (100657) (2020) http://dx.doi.org/10.1016/j.eml.2020.100657.

[25] H. Wang, L. Zhang, J. Han, W. E., DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics, Comput. Phys. Comm. 228 (2018) http://dx.doi.org/10.1016/j.cpc.2018.03.016.

[26] J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett. 98 (146401) (2007) http://dx.doi.org/10.1103/PhysRevLett.98.146401.

[27] D. Zapiain, J. Stewart, R. Dingreville, Accelerating phase-field-based microstructure evolution predictions via surrogate models trained by machine learning methods, NPJ Computat. Mater. 7 (1) (2021) 1–11, http://dx.doi.org/10.1038/s41524-020-00471-8.

[28] E. Herman, J. Stewart, R. Dingreville, A data-driven surrogate model to rapidly predict microstructure morphology during physical vapor deposition, Appl. Math. Model. 88 (2020) http://dx.doi.org/10.1016/j.apm.2020.06.046.

[29] K. Yang, Y. Cao, Y. Zhang, S. Fan, M. Tang, D. Aberg, B. Sadigh, F. Zhou, Self-Supervised Learning and Prediction of Microstructure Evolution with Convolutional Recurrent Neural Networks, Patterns Cell Press, 2021, http://dx.doi.org/10.1016/j.patter.2021.100243.

[30] C. Hu, S. Martin, R. Dingreville, Accelerating phase-field predictions via recurrent neural networks learning the microstructure evolution in latent space, Comput. Methods Appl. Mech. Engrg. 397 (2022) 115128, http://dx.doi.org/10.1016/j.cma.2022.115128.

[31] V. Oommen, K. Shukla, S. Goswami, R. Dingreville, G. Karniadakis, Learning two-phase microstructure evolution using neural operators and autoencoder architectures, NPJ Computat. Mater. 8 (1) (2022) 190, http://dx.doi.org/10.1038/s41524-022-00876-7.

[32] S. Desai, R. Dingreville, Learning time-dependent deposition protocols to design thin films via genetic algorithms, Mater. Des. 219 (2022) 110815, http://dx.doi.org/10.1016/j.matdes.2022.110815.

[33] A.S. Iquebal, P. Wu, A. Sarfraz, K. Ankit, Emulating the evolution of phase separating microstructures using low-dimensional tensor decomposition and nonlinear regression, MRS Bull. (2023) 1–12., http://dx.doi.org/10.1557/s43577-022-00443-x.

[34] C. Sun, M. Zhang, R. Wu, J. Lu, G. Xian, Q. Yu, X. Gong, R. Luo, A convolutional recurrent neural network with attention framework for speech separation in monaural recordings, Sci. Rep. 11 (1) (2021) 1–14, http://dx.doi.org/10.1038/s41598-020-80713-3.

[35] V. Hernandez, T. Suzuki, G. Venture, Convolutional and recurrent neural network for human activity recognition: Application on American sign language, PLoS One 15 (2) (2020) e0228869, http://dx.doi.org/10.1371/journal.pone.0228869.

[36] Y. Kim, Y. Jernite, D. Sontag, A.M. Rush., Character-aware neural language models, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016, http://dx.doi.org/10.48550/arXiv.1508.06615.

[37] Z. Zhang, Y. Dong, Temperature forecasting via convolutional recurrent neural networks based on time-series data, Complexity 2020 (3536572) (2020) http://dx.doi.org/10.1155/2020/3536572.

[38] Y.D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, D. Shin, Compression of deep convolutional neural networks for fast and low power mobile applications, in: The International Conference on Learning Representations, 2016, http://dx.doi.org/10.48550/arXiv.1511.06530.

[39] J.W. Cahn, On spinodal decomposition, Acta Metall. Mater. 9 (9) (1961) 795–801, http://dx.doi.org/10.1016/0001-6160(61)90182-1.

[40] K. Kamijo, T. Tanigawa., Stock price pattern recognition-a recurrent neural network approach, in: IJCNN International Joint Conference on Neural Networks, Vol. 1, 1990, pp. 215–221, http://dx.doi.org/10.1109/IJCNN.1990.137572.

[41] L. Medsker, L. Jain, Recurrent Neural Networks: Design and Applications, CRC Press, 2001.

[42] B. Shi, X. Bai, C. Yao, An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 2298–2304, http://dx.doi.org/10.1109/TPAMI.2016.2646371.

[43] R. Toral, A. Chakrabarti, J.D. Gunton, Numerical study of the Cahn–Hilliard equation in three dimensions, Phys. Rev. Lett. 60 (1988) 2311–2314, http://dx.doi.org/10.1103/PhysRevLett.60.2311.

[44] A. Bray, C. Emmott, Lifshitz–Slyozov scaling for late-stage coarsening with an order-parameter-dependent mobility, Phys. Rev. B 52 (2) (1995) R685.

[45] K. Ankit, B. Derby, R. Raghavan, A. Misra, M.J. Demkowicz, 3-D phase-field simulations of self-organized composite morphologies in physical vapordeposited phase-separating binary alloys, J. Appl. Phys. 126 (075306) (2019) http://dx.doi.org/10.1063/1.5110410.

[46] R. Raghavan, A. Mukherjee, K. Ankit, Nanostructural evolution in vapor deposited phase-separating binary alloy films of non-equimolar compositions: Insights from a 3D phase-field approach, J. Appl. Phys. 128 (17) (2020) 175303, http://dx.doi.org/10.1063/5.0007385.

[47] A. Jokisaari, P. Voorhees, J. Guyer, J. Warren, O. Heinonen, Benchmark problems for numerical implementations of phase field models, Comput. Mater. Sci. 126 (2017) 139–151, http://dx.doi.org/10.1016/j.commatsci.2016.09.022, URL https://www.sciencedirect.com/science/article/pii/S0927025616304712.

[48] L.-Q. Chen, J. Shen, Applications of semi-implicit Fourier-spectral method to phase field equations, Comput. Phys. Comm. 108 (2–3) (1998) 147–158.

[49] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, V. Lempitsky, Speeding-up convolutional neural networks using fine-tuned CP-decomposition, in: The International Conference on Learning Representations, 2015, http://dx.doi.org/10.48550/arXiv.1412.6553.