Emulating the evolution of phase separating microstructures using low-dimensional tensor decomposition and nonlinear regression

Ashif S. Iquebal,* Peichen Wu, Ali Sarfraz, and Kumar Ankit*

Impact statement

The phase-field method is an attractive computational tool for simulating microstructural evolution during phase separation, including solidification and spinodal decomposition. However, they remain computationally intensive due to the strict limits on the maximum time and length scales imposed by the numerical methods. This research presents a phase-field emulator that will predict the microstructural evolution observed in phase separating materials by leveraging the already existing data sets obtained from traditional phase-field modeling approaches. The phase-field emulator, which for the first time, couples a tensor representation and nonparametric tensor methods with microstructure modeling and representation, will enable high-throughput and facile prediction of the microstructural evolution, as opposed to solving computationally intensive phase-field simulations whenever a new simulation needs to be performed. Our data-driven microstructure emulator opens new avenues to predict the microstructural evolution by leveraging phase-field simulations and physical experimentation where the time resolution is often quite large due to limited resources and physical constraints, such as the phase coarsening experiments previously performed in microgravity.

The phase-field method is an attractive computational tool for simulating microstructural evolution during phase separation, including solidification and spinodal decomposition. However, the high computational cost associated with solving phase-field equations currently limits our ability to comprehend phase transformations. This article reports a novel phase-field emulator based on the tensor decomposition of the evolving microstructures and their corresponding two-point correlation functions to predict microstructural evolution at arbitrarily small time scales that are otherwise nontrivial to achieve using traditional phase-field approaches. The reported technique is based on obtaining a low-dimensional representation of the microstructures via tensor decomposition, and subsequently, predicting the microstructure evolution in the low-dimensional space using Gaussian process regression (GPR). Once we obtain the microstructure prediction in the low-dimensional space, we employ a hybrid input–output phase-retrieval algorithm to reconstruct the microstructures. As proof of concept, we present the results on microstructure prediction for spinodal decomposition, although the method itself is agnostic of the material parameters. Results show that we are able to predict microstructure evolution sequences that closely resemble the true microstructures (average normalized mean square of 6.78×10^{-7}) at time scales half of that employed in obtaining training data. Our data-driven microstructure emulator opens new avenues to predict the microstructural evolution by leveraging phase-field simulations and physical experimentation where the time resolution is often quite large due to limited resources and physical constraints, such as the phase coarsening experiments previously performed in microgravity.

Introduction

The phase-field method is a powerful tool to simulate microstructural evolution in materials undergoing phase transformations or degradation.[1–6] However, this method, which typically involves solving nonlinear partial differential equations over a three-dimensional (3D) computational domain, remains computationally intensive due to the strict limits on the maximum time and length scales implicitly imposed by the adopted numerical techniques.[7] For

Ashif S. Iquebal, School of Computing and Augmented Intelligence, Arizona State University, Tempe, USA; aiquebal@asu.edu
Peichen Wu, Materials Science and Engineering, School for Engineering of Matter, Transport, and Energy, Arizona State University, Tempe, USA
Ali Sarfraz, School of Computing and Augmented Intelligence, Arizona State University, Tempe, USA
Kumar Ankit, Materials Science and Engineering, School for Engineering of Matter, Transport, and Energy, Arizona State University, Tempe, USA; kankit@asu.edu
*Corresponding author

doi:10.1557/s43577-022-00443-x

instance, the computational complexity associated with simulating phase separation in those metallic melts where dendritic evolution is prevalent, scales as $\mathcal{O}(S^4)$, where S is the grid size.[8] Similarly, to simulate the microstructural evolution in materials that accompany spinodal decomposition, the permissible length and time scales are even more restrictive,[5,9] and warrant the application of supercomputers for solving fourth-order Cahn–Hilliard partial differential equations (PDEs). Depending on the spatial and time resolution required, the run times for such phase-field simulations of microstructural evolution in three dimensions, which is a four-dimensional (4D) problem (considering the time evolution), can span from several hours to weeks on a supercomputing cluster. Current strategies to accelerating phase-field simulations rely on high-performance graphical processing units (GPUs) and power-intensive computational resources as opposed to intelligently leveraging the past simulations to emulate new ones. As a result, a new simulation needs to be executed every time a simulation parameter (e.g., atomic diffusion coefficient) is changed.

Over the past two decades, the increase in computational power with the emergence of supercomputers such as TSUBAME 2.0[10] and GPUs such as NVIDIA Tesla C2050,[11] to some extent, has helped address some of the limitations.[12–15] In addition to high-performance GPUs, various numerical methods have also been developed such as adaptive meshing,[16,17] time stepping,[18,19] and spectral methods[20] to speed up the microstructure simulation. Nonetheless, phase-field simulations at realistic length and time scales still require millions of computing core hours even on parallel computing platforms,[21–23] not to mention their high installation and maintenance costs.

More recently, deep-learning strategies have been explored for modeling microstructure evolution. For instance, Yang et al.[24] implemented a recurrent neural network (RNN) with eidetic 3D long short-term memory (LSTM) cells to predict the evolution pattern in four different evolution phenomena, including plane-wave propagation, grain growth, spinodal decomposition, and dendrite growth. The methodology involved training the RNN with microstructure sequences followed by predicting the microstructure evolution for a new initial microstructure. In another work, Montes de Oca Zapiain et al.[25] implemented an LSTM network to predict the microstructure evolution in phase-field simulation after obtaining a low-dimensional representation using two-point statistics and principal component analysis. Although these approaches have demonstrated the viability of using deep learning for microstructure prediction, there are some limitations. First, deep learning approaches remain data hungry and therefore their performance is highly contingent on the availability of large data sets. For instance, the LSTM network trained by Montes de Oca Zapiain et al.[25] required 5000 high-fidelity phase-field simulation each with 60 frames. Another limitation of the existing approaches is the use of principal component analysis (PCA) for dimensionality reduction. Although PCA could be used to obtain a low-dimensional representation, they make little sense when dealing with sequence of images. Because PCA implementation requires unfolding an image data into a one-dimensional array, the spatiotemporal information is lost. This is also reflected in the poor reconstruction of the microstructures.[25,26]

In this work, we develop a phase-field emulator to predict the microstructure evolution at arbitrary time scales without solving costly PDEs. Our approach is based on representing the microstructure evolution as a tensor and obtaining a low-dimensional tensor decomposition of the two-point correlation functions representing the evolving microstructures. The tensor decomposition does not require any unfolding and simultaneously preserves the spatiotemporal relationships, therefore overcoming the limitations of PCA. With the low-dimensional representation at hand, we then employ Gaussian process regression (GPR)—a nonparametric nonlinear regression approach—to predict the microstructure evolution in the low-dimensional space at arbitrary time steps that cannot be accomplished without running multiple phase-field simulations with distinct time step widths. For training, we generate spinodal decomposition via high-fidelity phase-field simulations. After we obtain the predictions from GPR, we use a hybrid input–output phase-retrieval algorithm to reconstruct the microstructures from the predicted two-point correlation functions. An outline of the proposed methodology is presented in **Figure 1**. Finally, we demonstrate the efficacy of the methodology on spinodal decomposition of a two-phase material system.

Results and discussion

To predict the microstructure evolution, we first generate the training data set by simulating spinodal decomposition using the Cahn–Hilliard equation given as:

$$\frac{\partial \phi}{\partial t} = M\nabla^2\left[\frac{1}{2}W\left(2\phi^3 - 3\phi^2 + \phi\right) - \varepsilon^2\nabla^2\phi\right], \qquad 1$$

where $\phi \equiv \phi(x,t)$ is the scaled concentration field of the A–B alloy, M is the mobility of diffusing species, ε is the gradient free energy coefficient, which is related to interfacial energy, and W is the well height of the double-well potential $f(\phi)$ given as:

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

A detailed description of the phase-field simulation is provided in the "Methods" section. For simulation, we fixed $\varepsilon = 0.051$ and diffusivity $D = M\partial^2 f(\phi)/\partial\phi^2 = 1$. The temporal discretization was selected using the relationship $\delta t = \delta x^4 W/2^{2d+1}M\varepsilon^2$,[9] where d equals 2 for two-dimensional (2D) simulation domains. We set the grid spacing both in x and y directions as $\delta x = 0.3$ such that the discretization in the temporal domain is $\delta t = 0.0002$. For a given set of simulation parameters, we record the microstructure evolution for 20 frames. The generated microstructures are of sizes 32 × 32,

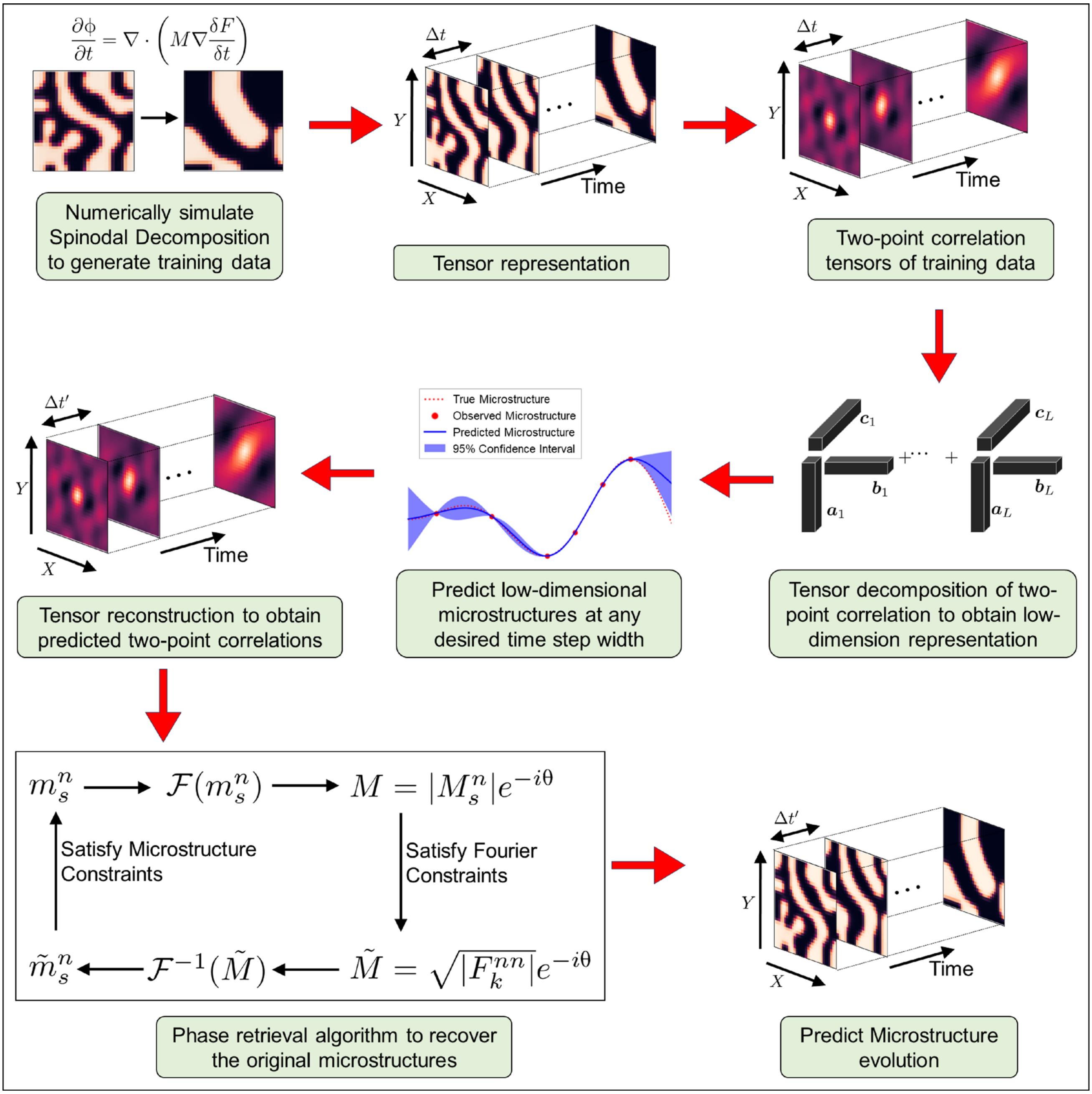
$\frac{\partial\phi}{\partial t} = \nabla \cdot \left(M\nabla\frac{\delta F}{\delta t}\right)$
Numerically simulate Spinodal Decomposition to generate training data
Δt
Y
X
Time
Tensor representation
Δt
Y
X
Time
Two-point correlation tensors of training data
c_1
b_1
a_1
$+\cdots+$
c_L
b_L
a_L
Tensor decomposition of two-point correlation to obtain low-dimension representation
True Microstructure
Observed Microstructure
Predicted Microstructure
95% Confidence Interval
Predict low-dimensional microstructures at any desired time step width
Δt′
Y
X
Time
Tensor reconstruction to obtain predicted two-point correlations
$m_s^n \longrightarrow \mathcal{F}(m_s^n) \longrightarrow M = |M_s^n|e^{-i\theta}$
Satisfy Microstructure Constraints
Satisfy Fourier Constraints
$\tilde{m}_s^n \longleftarrow \mathcal{F}^{-1}(\tilde{M}) \longleftarrow \tilde{M} = \sqrt{|F_k^{nn}|}e^{-i\theta}$
Phase retrieval algorithm to recover the original microstructures
Δt′
Y
X
Time
Predict Microstructure evolution

Figure 1. Flow sequence outlining our microstructure emulation approach.

256 × 256, 512 × 512, and 1024 × 1024 grid points. For demonstration of the methodology and results, we use the microstructures of size 32 × 32 grid points. Comparative results for other grid sizes are presented afterward.

We begin by considering a statistical representation of the microstructures. Mathematically, a microstructure function m_s^n is represented as the probability of finding some local state n at a spatial location $s \in \mathbb{R}^2$ for 2D microstructures. Given the stochastic nature of microstructures in space, we first obtained their statistical representation using two-point correlations. Intuitively, the two-point correlations, denoted as f_{np}^r, capture the likelihood of observing some combination of microstructure states (n and p) at two randomly selected locations separated by a distance of r. For a two-state microstructure (say 0 and 1), as we are concerned in this work, there are a total of four combinations of states (i.e., 00, 01, 11, and 10) and therefore, four sets of two-point correlations. However, two-point correlation functions have inherent redundancies in them, making them interdependent. For instance, the probability of observing states 1 and 0 in spatial bins separated by a distance of r

(i.e., f_{10}^{r}) is the same as observing the two states in the reverse order separated by a distance of $-r$ (i.e., f_{01}^{-r}). Considering all the redundancies, for a two-state microstructure, we only need to compute either an autocorrelation or a cross-correlation function and the remaining correlations could be recovered.[27] Because the two-point spatial correlation summarizes the pairwise distribution of local states within the microstructure, they are particularly useful in tracking microstructure evolution. Details of the two-point correlation functions are provided in the "Methods" section. To efficiently calculate the two-point correlations, we subscribed to discrete Fourier transform (DFT) of the microstructure function m_s^n. We first note that the DFT of the two-point statistics is given as:

$$\mathcal{F}_k^{np} = \mathcal{F}(f_{np}^{r}) = \frac{1}{S}|M_k^n|e^{i\theta_k^n}|M_k^p|e^{i\theta_k^p}, \qquad 3$$

where $\mathcal{F}$ is the DFT operator, $|M_k^n|$ denotes the magnitude component, and θ_k^n represents the phase component of the microstructure function m_s^n for state n. For $n = p$, the DFT of the two-point statistic gives us the autocorrelation function and for $n \neq p$, the DFT gives the cross-correlation function. As mentioned earlier, we only need either an autocorrelation or cross-correlation function for a two-phase microstructure and the rest could be recovered. In the remainder of the discussion, we work with the autocorrelation function of one of the states. **Figure 2**a–b shows representative examples of the microstructure and its corresponding autocorrelation function, respectively. We note that autocorrelation functions follow a similar evolution trend as their corresponding microstructures (i.e., they evolve rapidly during the initial stages and slowly in the later stages of coarsening).

We next look at the tensor representation and decomposition of the autocorrelation functions. Microstructure evolution lies in a high-dimensional space (e.g., the evolution of a 323232 microstructure), which is the smallest data set we are working with, for just 10 time steps (or samples) lies in a 327,680 dimensional space (**Figure 3**). To handle such high-dimensional data structures, microstructures and their relevant statistics such as two-point correlations have traditionally been represented via matrix or vector-based methods such as principal component analysis that require flattening them into vectors.[25,26,28–32] However, upon vectorization, microstructure sequences lose their topological structures and dependencies across spatial and temporal modes. The absence of an appropriate mathematical representation has precluded the development of models that can efficiently capture the higher-order nonlinear patterns in microstructure evolution. In this regard, we provide a tensor representation of microstructure evolution. More specifically, we consider microstructure evolution as a three-way tensor with dimensions of parameter × space × time. Mathematically, microstructure evolution will be represented as $\mathcal{X} : P \times S \times T$, where P, S, and T represent the dimension of parameter, spatial, and temporal modes, respectively. For the emulation results reported next, the parameter mode is kept constant. To obtain a low-dimensional representation of the microstructure evolution, we employ a tensor decomposition approach known as canonical polyadic (CP) decomposition. Tensor decomposition is an extension of conventional PCA for multiway data sets that do not require any unfolding. Several methods for tensor decomposition exist such as CP,[33] Tucker,[34] and tensor train.[35] In this work, we implement CP

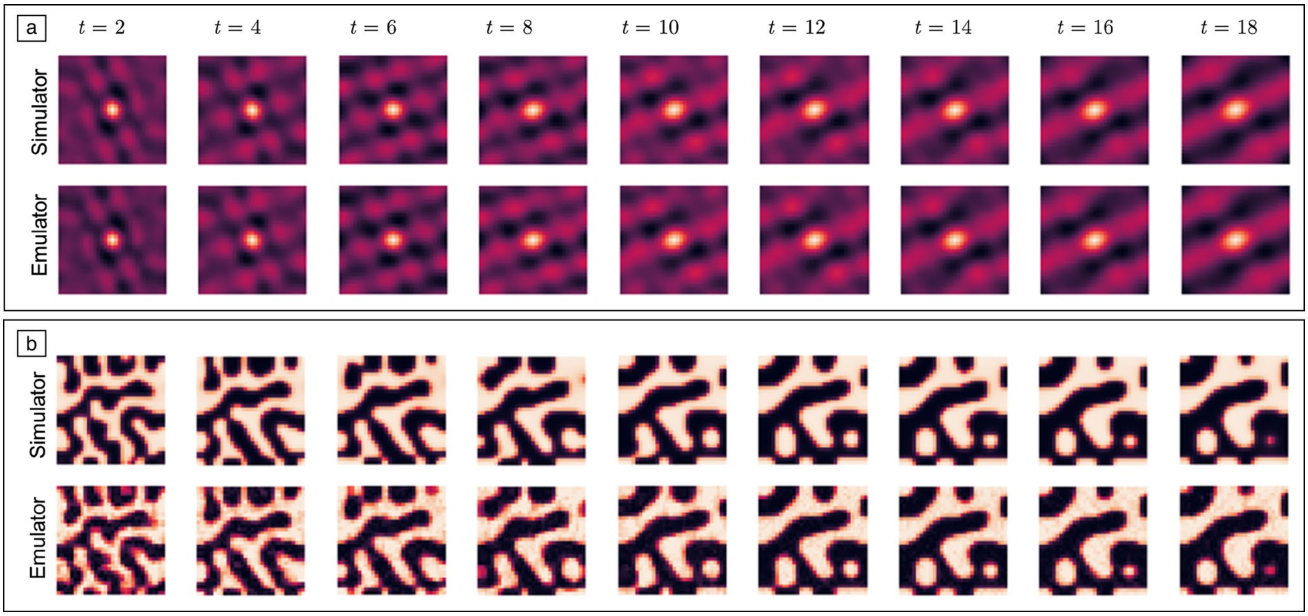

Figure 2. Frame-by-frame comparison of microstructure evolution simulated using phase-field method with the corresponding emulation. The top row shows the original sequence of the (a) two-point correlation and (b) autocorrelation functions of the phase-field generated data set while the emulations are plotted in the bottom row.

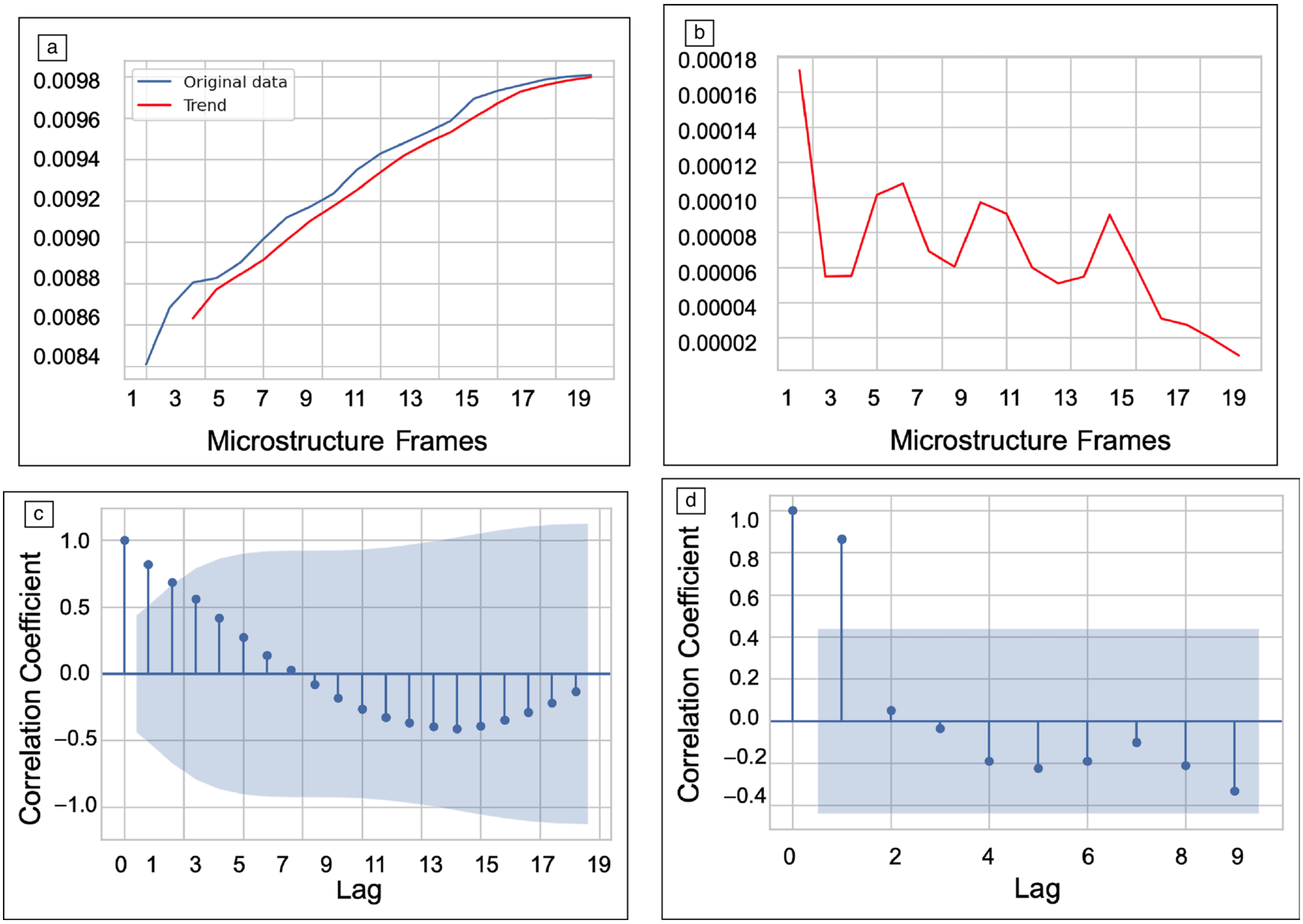

Figure 3. Exploratory analysis of the microstructure data in the low-dimensional space. (a) The first rank-one tensor across the temporal dimension (referred as original data) and the corresponding trend, (b) seasonality in the original data, (c) autocorrelation function, and (d) partial autocorrelation function.

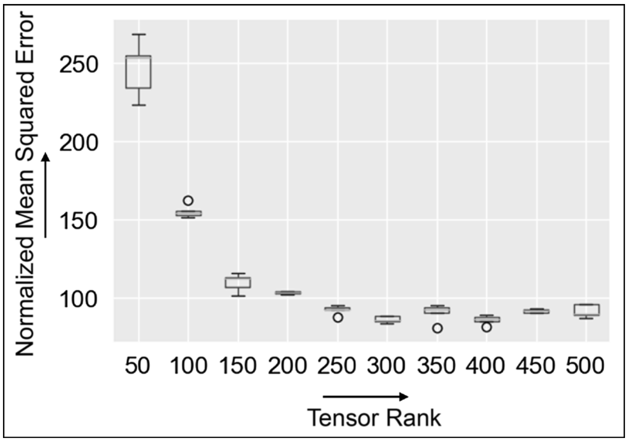

Figure 4. Variation in the NMSE for microstructure reconstruction with different tensor ranks.

decomposition as it factorizes the microstructure tensor into an outer product of rank-one tensors that are simple to handle using GPR. The number of such rank-one tensors represents the rank of the original microstructure tensor. Intuitively, rank of the microstructure tensor increases as the complexity in the spatiotemporal makeup of the microstructures increases. However, computing the rank parameter is NP hard and therefore, we perform a sensitivity analysis to determine the optimal rank. Essentially, we check the reconstruction error for varying ranks and select the rank for which the reconstruction error is within 5% of the lowest reconstruction error. We measure the reconstruction error using normalized mean squared error (see Equation 10 in the "Methods" section). **Figure 4** shows the reconstruction error for different ranks. We obtain the lowest average normalized mean squared error (NMSE) of 86.17 for a rank of 400. However, for a lower rank of 300, we obtain the average NMSE of 86.32, which is within 5% of the lowest NMSE. Therefore, to reduce computational complexity, we fix the tensor rank to 300 for rest of the discussion. Additional details on CP decomposition and rank estimation are provided in the "Methods" section.

Before we present the results on GPR fitting, we provide an overview of the microstructure data from a low-dimensional perspective. Figure 3a shows the first rank-one tensor across the temporal domain after performing CP decomposition and its corresponding trend is shown in red. Clearly, the rank-one tensor component shows a gradual increase as the microstructure evolves. In Figure 3b, we notice some seasonality effect that is obtained after removing the trend from the original data. Because the magnitude of seasonality is almost one-tenth of the magnitude of the original data, we can assume these variations to be noise. Furthermore, in Figure 3c–d, we present the autocorrelation (i.e., correlation between the data and its lagged version) and the partial autocorrelation function that is the same as autocorrelation, but without the correlation effect from data in between. From the autocorrelation plot, we can infer that successive data points are highly correlated, especially for smaller lag values. From a microstructure standpoint, this indicates that the microstructure sequences at successive time instants are highly correlated, which is also evident from Figure 2. The partial autocorrelation plot in Figure 3d shows a strong correlation if lag is 1 (i.e., for successive data points), but it drops rapidly afterward. This again reaffirms our observation that the microstructure images are strongly correlated.

After we obtain a low-dimensional representation of the microstructure tensor, we train the data-driven emulator such that it can predict the microstructure evolution. To demonstrate the efficacy of the proposed emulator, we first train the emulator using the microstructure evolution obtained at a large time step and predict the microstructure evolution at smaller time steps. In particular, we construct the training data by considering the microstructures at time steps $2\delta t$, including the first and the last microstructure frames and predict the microstructures at time steps δt. Every microstructure sequence contains $N = 20$ frames and therefore, the training data are comprised of 11 frames such that the dimension of the microstructure tensor is $32 \times 32 \times 11$. The input space after CP decomposition is 11×300 in the temporal dimension and 32×300 across each of the two spatial modes. To predict the microstructure evolution at a smaller time step, we now merely have to perform training and testing over the temporal mode.

After we obtain the low-dimensional representation of the microstructures, we use GPR to emulate microstructural evolution. Details of the GPR and its posterior predictive distribution are provided in the "Methods" section. To model the covariance structure, we use a Matérn kernel function given as:

$$K(t, t') = \left(1 + \frac{\sqrt{3}(t - t')^2}{\sigma^2}\right) \exp\left(-\frac{\sqrt{3}(t - t')^2}{\sigma^2}\right), \quad 4$$

where σ^2 is the length scale parameter. We train one GPR for each of the rank-one tensors across the temporal dimension obtained from training frames $0, 1, 3, 5, \ldots, 19$ and subsequently, predict the rank-one tensors for frames $0, 1, 2, \ldots 18, 19$. For every training microstructure sequence, we perform hyperparameter (σ^2) optimization by maximizing the log-marginal likelihood function (see "Methods" section for details). We tune the hyperparameters individually for every rank-one tensor. After obtaining the predictions, we reconstruct the autocorrelations by using the outer product of the predicted one-dimensional tensors across every mode. The top row in Figure 2a shows the two-point correlation functions obtained from the phase-field simulator while the corresponding predictions obtained from the emulator are plotted in the bottom row. To demonstrate the advantage of using GPR over other nonlinear regression methods, we compare the corresponding predictions in a representative low-dimensional representation with that of support vector regression with two different kernel functions: (1) radial basis kernel and (2) polynomial kernel. The corresponding predictions are shown in **Figure 5**. From the figure, it is evident that GPR is able to accurately fit the data points, whereas support vector regression fails to capture the nonlinear behavior.

Once we have predicted the autocorrelation functions, the last step involves recovery of microstructures from the autocorrelation functions. This implies extracting phases from the amplitude information because the phase information is typically lost during the convolutions performed for determining autocorrelation functions. Thus, we employ an iterative hybrid input–output phase-retrieval algorithm,[36] based on the widely known Gerchberg–Saxton algorithm,[37] the details of which are provided in the "Methods" section and **Figure 6**. Traditionally, phase-retrieval methods are known to suffer from poor reconstruction inaccuracies that depend on the initial conditions.[25,26] To circumvent this issue, we add a padding around the microstructures before extracting the autocorrelation functions during the training phase. The padding around each of the microstructures, although increases the computational complexity, significantly improves the reconstruction accuracy. In this work, we find the optimal padding on every side to be equal to the image dimensions such that a computational microstructure of size 32×32 upon padding scales up to 96×96. The value of pixels in the padding is set to zero. Because the phase-retrieval algorithm is iterative, if the NMSE of the reconstructed microstructure does not change beyond 5% for five consecutive iterations, the flow sequence exits. The top row in Figure 2b shows the microstructure frames that were not included in the training data set, while the corresponding predictions obtained from the phase-field emulator are shown at the bottom. We note that the predicted microstructures closely resemble the true microstructures.

The emulated domain size is plotted as a function of time for distinct δt, as shown in **Figure 7**a. See the "Methods" section (Equation 11) for details on the calculation of average feature size. As indicated by these plots, the scaling dynamics are found to nearly overlap with the phase-field results. To further quantify these emulations, we calculate the NMSE (see Equation 10) of the predicted microstructures for different values of δt as shown in Figure 7b. The average NMSE for distinct δt is equal to 0.0314, 0.0280, 0.0236, 0.0207, and 0.0128.

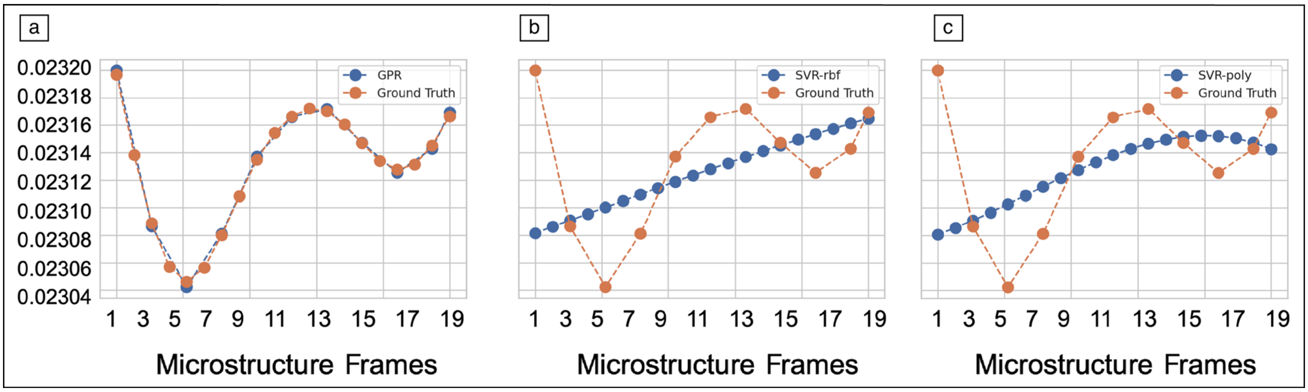

Figure 5. Comparison of prediction in a representative low-dimensional space obtained from (a) Gaussian process regression (GPR) with support vector regression (SVR) using (b) radial basis function (rbf) kernel and (c) polynomial kernel. The *y*-axis is unlabeled as it may not have physical significance in the low-dimensional space.

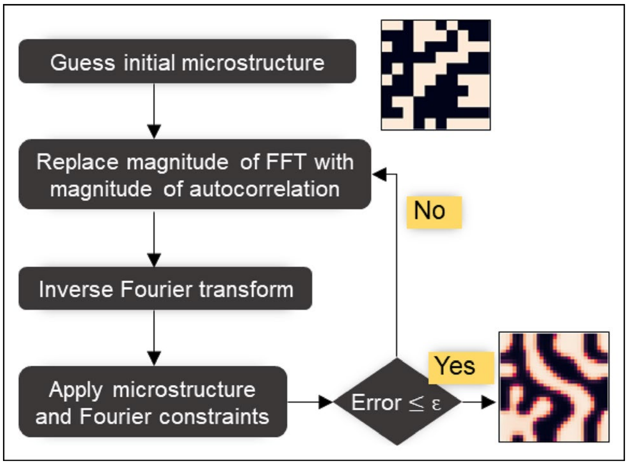
Guess initial microstructure
Replace magnitude of FFT with magnitude of autocorrelation
Inverse Fourier transform
Apply microstructure and Fourier constraints
Error ≤ ε
No
Yes

Figure 6. Flowchart of the hybrid input–output algorithm for microstructure reconstruction from the predicted autocorrelation functions.

At this point, some important observations are in order: First, the reconstruction of the microstructures obtained from our methodology is statistically accurate (with average NMSE $= 6.78 \times 10^{-7}$) as evident from Figure 2a. Next, we notice that the proposed method is able to predict the microstructure evolution both in the transient stages initially where the phase separating domain size increases rapidly as well as later during the slow coarsening stage with a sufficiently high accuracy (see Figure 2b). It is worth emphasizing that predicting microstructural evolution during the transient stages has proved to be challenging for recently published algorithms[24,25] as the microstructures evolve rapidly in the initial stages, making it difficult to capture the evolving microstructural features. In contrast, the tensor-based phase-field emulator, developed in this research, preserves the spatiotemporal relationships even in the low dimensions; therefore, we are able to accurately predict the microstructural evolution and recover the corresponding scaling dynamics. These results show that the proposed approach is not only statistically accurate, but is also able to capture the complex microstructural features. Finally, we notice that the NMSE of the predicted microstructures decreases as the spinodal decomposition proceeds toward the late-stage coarsening regime given that the feature complexity is higher when the average phase separated domain size is small. We also observe that the average NMSEs decrease as the time step width, δt, for the training data decrease, which is owing to the larger interpolations that are performed by the emulator at larger δt, causing a relative loss in accuracy.

The phase-field emulator proposed in this work involves four steps as outlined in Figure 1. The first three steps comprise the training phase of the emulator, which includes extraction of the autocorrelation functions, tensor decomposition, and GPR training. Testing phase involves prediction using GPR and hybrid input–output algorithm for phase retrieval. For a microstructure with S grids evolving for T time periods and a tensor rank of R, the computation cost during the training phase for each of the steps are $\mathcal{O}(S)$, $\mathcal{O}(STR)$, and $\mathcal{O}(RT^3)$, respectively. Note that the computational complexity for tensor decomposition is determined for CP decomposition with alternating least squares.[38] For the testing phase, the computational complexity for a single microstructure image is $\mathcal{O}(R)$ for GPR prediction and $\mathcal{O}(MS \log S)$ for the hybrid input–output algorithm considering M iterations. If the rank R of CP decomposition is fixed, then the computational complexity during the testing phase scales as $\mathcal{O}(MS \log S)$, making phase retrieval as the most resource-intensive step. **Table I** shows the breakdown of computational cost for each of the steps. All computations presented in this work were performed on an Intel(R) Core(T) i9-10900K CPU with 32 GB of RAM without any GPU acceleration. The run time for obtaining the microstructure sequence from phase-field simulation required approximately 0.19 sec for microstructure of size 32 × 32. For every microstructure sequence generated from the phase-field simulation, we used $n/2 + 1$ of the frames to construct the training data set. The distribution for each of the steps involved in the phase-field emulator are as follows: In the training phase, the calculation

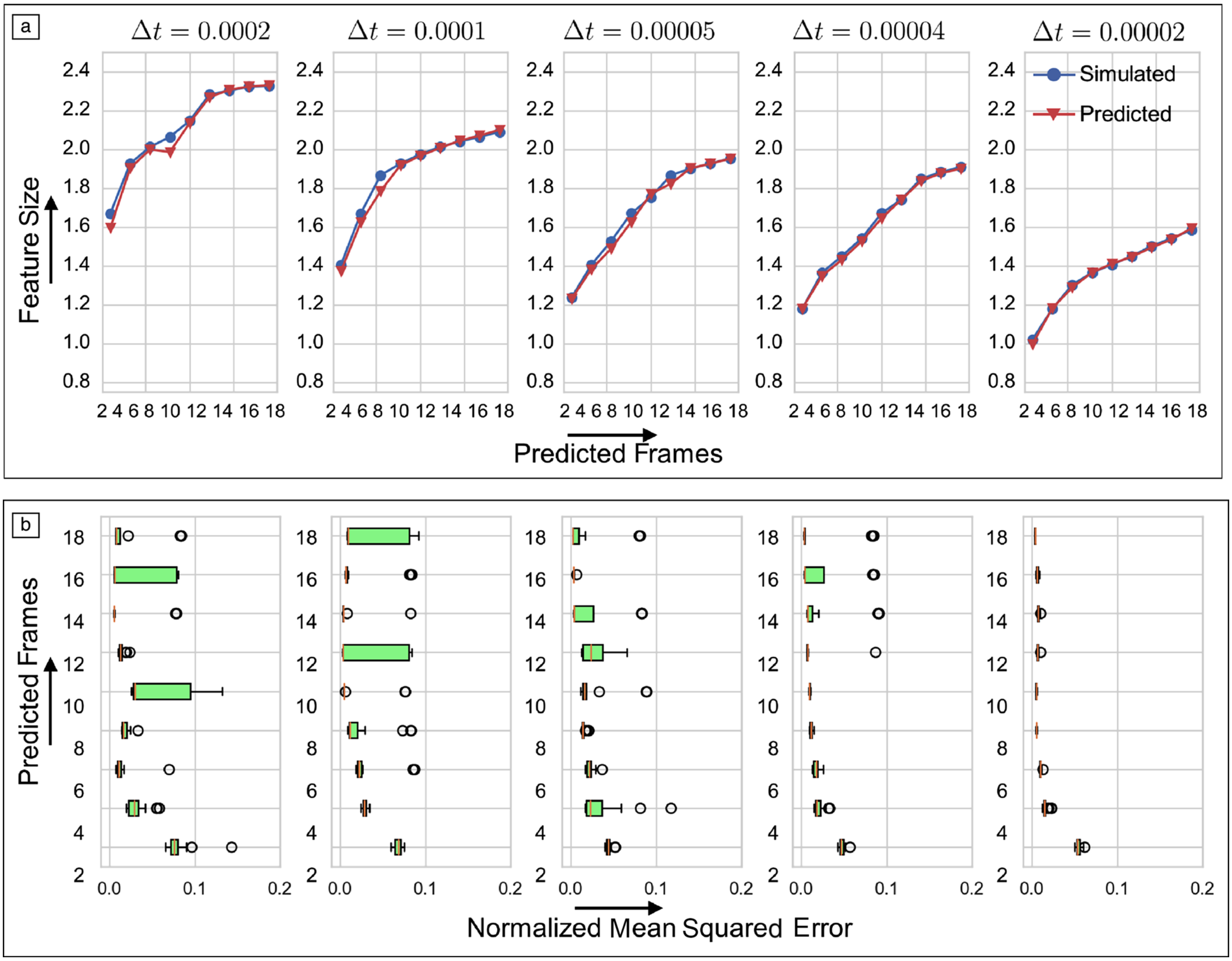

Figure 7. (a) Comparison of the microstructure feature size of the predicted microstructure and the simulated microstructure for different values of the time step, δt. (b) Normalized mean squared error for predicted microstructures for different values of time step, δt.

Table I. Computational run times of emulator and phase-field simulations (in s).

Domain	Training Time	Testing Time	PFS Time	PFS to Emulator
32 × 32	0.24	0.12	0.19	1.58
256 × 256	0.78	12.98	68	5.24
512 × 512	3.18	64.5	588.97	9.13
1024 × 1024	198.88	289.58	5057.3	17.46

of autocorrelation function consumed 44.4%, CP decomposition consumed 41.8%, while training the GPR required 13.8% of the total training time. In the prediction phase, GPR predictions accounted for merely 0.4% and the microstructure reconstruction from GPR predictions consumed more than 99.6% of the total run time. We also compare the computational costs of microstructure prediction with an increase in domain size. Table I shows the time (in s) for generating one microstructure in the sequence for 32 × 32, 256 × 256, 512 × 512, and 1024 × 1024. The last column shows the ratio of time taken by the phase-field simulator for simulating one microstructure to the time taken by phase-field emulator for predicting one microstructure. We note that the proposed emulator is at least 5 × faster as compared to the phase-field simulator. Interestingly, the phase-field emulator performs faster as the domain size increases. In fact, for the largest domain size of 1024 × 1024, the phase-field emulator is more than 17× faster as compared to the phase-field simulations.

Although the foregoing discussions have employed phase-field simulations of spinodal decomposition, the proposed approach is generalizable to other microstructural evolution scenarios, such as coarsening of precipitates and dendritic growth. Particularly, the proposed approach could be attractive for two applications. The first application involves obtaining the microstructure evolution at arbitrarily small time scales via phase-field methods. Traditionally, the computational complexity limits the temporal resolution of phase-field simulations. With the proposed phase-field emulator, it is possible to obtain the microstructure evolution at a smaller time step (i.e., higher temporal resolution using just the microstructure evolution obtained at a larger time scale). Therefore, the emulator would preclude the need for running multiple simulations whenever there is a change in the time step parameter. The second application will be in physical experiments, such as those that were performed under microgravity by NASA, where physical restrictions and resource constraints limit the number of experiments that could be performed.[39] Similar limitations also exist when observing microstructure evolution *in situ* using costly high-fidelity measurement techniques such as a transmission electron microscope[40] that often limit the time steps at which the microstructures could be recorded. Under such circumstances, the proposed emulator could provide an opportunity to observe the microstructure evolution at intermediate time steps that could not be recorded by the measurement system.

We also note some limitations of the proposed approach. In the current implementation, we first obtain a tensor decomposition followed by a separate GPR for each rank-one tensor. This two-step approach could be computationally intensive when complex microstructural features are involved that require higher rank CP decomposition. In our future works, we aim to develop nonlinear tensor regression methods that would not require any decomposition. Second, limitation of the approach originates from the CP decomposition itself. CP decomposition results in rank-one tensor across each of the temporal and spatial modes. Because we only predict across the temporal mode, the information across the spatial mode is smeared across the temporal mode. As a result, we notice that the reconstruction errors are higher in the initial stages of spinodal decomposition when the microstructures are rapidly evolving (see Figure 2b). Finally, we also notice some limitations with the two-point correlations. For two-phase microstructures (as considered in this work), it was possible to handle two-point correlations because we can summarize all the information with just either autocorrelation or cross-correlation. However, for a microstructure with multiple phases, two-point correlations could be cumbersome and could require higher-order correlation functions.

Conclusions

In closing, we have presented a tensor-based phase-field emulator to predict the microstructure evolution at arbitrary time scales by leveraging the existing phase-field simulations. Traditionally, the computational complexity of phase-field simulations limit the smallest time scales that are tractable. In light of this, the proposed approach provides an alternative to obtain phase-field simulations at smaller time scales without the need to run costly simulations. By casting microstructure evolution as a tensor, the proposed approach preserves the spatiotemporal relationships in the low-dimensional representations obtained via tensor decomposition. Overall, the proposed phase-field emulator is able to predict the microstructures with average NMSE of 6.78×10^{-7}. We also highlight that the phase-field emulator is material agnostic and could be applied to a wide array of evolving microstructures. Our current and future works are focused on predicting the microstructure evolution for different phase-field parameters such as mobility and gradient free energy. We are also exploring advanced tensor-on-tensor regression approaches that could be a viable option, especially when microstructural features are complex.

Methods

Phase-field simulation

For simulating the microstructural evolution during spinodal decomposition, the total free energy of the system (F) can be formulated as:

$$\frac{F}{k_B T} = \int_V \left\{ \underbrace{\phi^2(1-\phi)^2}_{F_{\text{bulk}}} + \underbrace{\kappa|\nabla\phi|^2}_{F_{\text{interface}}} \right\} dr, \qquad 5$$

where $\phi \equiv \phi(\mathbf{r}, t)$ is the composition field of the A–B alloy, which decomposes into an A-rich and a B-rich stable phase. F_{bulk} represents the bulk free energy that accounts for the two stable phases, $\phi = \{0, 1\}$ below the critical temperature, T is the deposition temperature, and κ is the gradient energy coefficient, which penalizes gradients in the order parameter. The kinetic evolution of the conserved order parameter ϕ follows Cahn–Hilliard dynamics:

$$\frac{\partial\phi}{\partial t} = \nabla \cdot M\nabla\mu, \qquad 6$$

where M is the composition-dependent mobility proposed by Bray and Emmott,[41] $M = M_0\left(1-\phi^2\right)^{\alpha}$ and μ is the chemical potential given as $\mu = \delta F/\delta\phi$. The exponent equals 0.0 for volume diffusion-limited and 1.0 for interfacial diffusion-dominated coarsening.[41] We nondimensionalize the model parameters previously described by selecting characteristic energy scale F', length scale L', and time t', where $F' = k_B T$, $L' = \delta x$, and $t' = M_0$. We solve the dynamic Equation 6 using an explicit finite-difference method, where the spatial derivatives are discretized using central difference, which is second-order accurate in space, and temporal discretization is done using a first-order Euler method.

Two-point correlation functions

We first begin with a mathematical description of the microstructures. For a discrete microstructure as described in the foregoing ("Results and discussion" section), we denote m_s^n as the probability of finding the local state $n = 1, 2, \ldots, N$ at spatial location $s = 1, 2, \ldots, S$ such that $\sum_n m_s^n = 1$ and $m_s^n \geq 0$. Note that for a two-phase microstructure, $N = 2$. Such a mathematical representation allows one to characterize the microstructure using various statistical measures such as n-point statistics, chord length, and nearest-neighbor functions.[42] One of

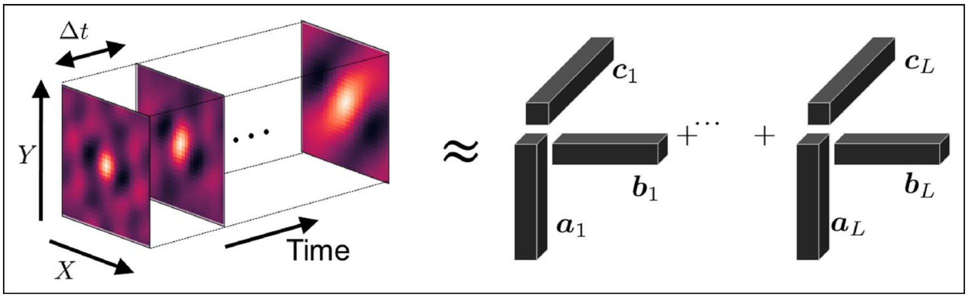

Figure 8. Schematic of tensor decomposition.

the most commonly employed statistic is the two-point spatial correlation that captures the conditional probability of finding a local state p at location $s + r$ given that a local state n is present at location s and is given as:[43]

$$f_{\boldsymbol{r}}^{np} = \frac{1}{S_r}\frac{1}{J}\sum_{\boldsymbol{s}}^{S_r}\sum_{j=1}^{J} {}^{(j)}m_{\boldsymbol{s}}^{n\,(j)}m_{\boldsymbol{s+r}}^{p}, \tag{7}$$

where S_r represents the number of admissible values of r for a given s and is upper bounded by S. Note that superscript $j = 1, 2, \ldots J$ represents different realizations of the microstructure.

Low-dimensional representation

CP decomposition also known as PARAFAC decomposition is a generalization of singular value decomposition to multiway tensor data sets. Originally proposed by Carroll and Chang,[44] CP decomposition aims to decompose a tensor $\mathcal{X}$ of dimension $n \times n \times T$ into an outer product of rank-one tensors given as:

$$\mathcal{X} \approx \sum_{i=1}^{R} \lambda_i \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i = [[\boldsymbol{\lambda}, \mathbf{A}, \mathbf{B}, \mathbf{C}]], \tag{8}$$

where R is a positive number representing the number of rank-one tensors needed to represent $\mathcal{X}$, λ_i represents the weights, $\boldsymbol{a}_i \in \mathbb{R}^n$, $\boldsymbol{b}_i \in \mathbb{R}^n$, and $\boldsymbol{c}_i \in \mathbb{R}^T$ represent the rank-one tensors, $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_R]$ and lastly, $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ are the factor matrices containing the rank-one tensors as columns (i.e., $\mathbf{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_R]$ and likewise for $\mathbf{B}, \mathbf{C}$). Ideally, the number of rank-one tensors required to recover the tensor $\mathcal{X}$ is equal to the rank of $\mathcal{X}$. However, there is no finite algorithm for computing the rank of a tensor, leading to the approximation in Equation 8. To identify the minimum number of rank-one tensors needed for CP decomposition, we start with an arbitrary rank and gradually increase until a good fit (say 95%) is obtained. With the rank R specified, we use the alternating least squares (ALS) to identify the rank-one tensors by minimizing the Frobenius norm between the original $\mathcal{X}$ and the reconstructed tensor $\hat{\mathcal{X}}$ (i.e., $\hat{\mathcal{X}} = \arg\min_{\hat{\mathcal{X}}} ||\hat{\mathcal{X}} - \mathcal{X}||_F$). For a three-way tensor as considered in our case, the ALS approach fixes one of the factor matrices to solve for the other two and continues iteratively until some convergence criterion is reached. Exact steps of ALS could be found elsewhere.[45] A schematic of the CP decomposition approach is presented in **Figure 8**. In this work, we implement the PARAFAC function from the TensorLy package in Python.[46]

Gaussian process regression

Once the microstructure evolution is projected in the rank-one tensor space, we can learn the microstructural variations using statistical and machine learning techniques. In this work, we choose Gaussian process regression, a nonparametric regression model that is well suited to capture the nonlinear function variability. The flexibility of Gaussian process regression arises from the assumption that the underlying data—in this case rank-one tensors, $\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i, i = 1, 2, \ldots, R$—are drawn from a Gaussian process with mean function $\vartheta(x)$ and covariance function $k(x, x')$ (i.e., $f(x) \sim \mathcal{GP}(\vartheta(x), k(x, x'))$, where x is the domain for each of the rank-one tensors. For simplicity, we assume a zero mean Gaussian process prior in this work.

Let us represent the time step and the target variable for the ith rank-one tensor be $(T, \boldsymbol{c}_i) = \{(t_1, c_{i1}), (t_2, c_{i2}), \ldots, (t_o, c_{io})\}$. For any $t_* \notin \{t_1, t_2, \ldots, t_o\}$, the posterior predictive distribution is given as $c_{i*}|T, \boldsymbol{c}_i, t_* \sim \mathcal{N}(\overline{c}_{i*}, \mathrm{cov}(c_{i*}))$, where:

$$\overline{c}_{i*} = K(t_*, T)[K(T, T) + \sigma^2 I]^{-1} \boldsymbol{c}_i$$

$$\mathrm{cov}(c_{i*}) = K(t_*, t_*) - K(t_*, t_*)'[K(T, T) + \sigma^2 I]^{-1} K(T, t_*)$$

where $K(\cdot, \cdot)$ is the covariance matrix.

Microstructure recovery

After the microstructure predictions are obtained from GPR, we reconstruct the microstructure using a two-step procedure. First, we reconstruct the two-point correlations corresponding to the predicted values of the rank of tensors. With the corresponding rank-one tensors, the reconstructed two-point correlation is given as:

$$\hat{\mathcal{X}} = \sum_{l=1}^{R} \lambda_l \boldsymbol{a}_l \circ \boldsymbol{b}_l \circ \boldsymbol{c}_l. \tag{9}$$

To recover the microstructure from reconstructed two-point correlation $\hat{\mathcal{X}}$, we use a hybrid input output (hIO) phase-retrieval algorithm also known as the Gerchberg–Saxton algorithm.[37] Note that traditionally, phase-retrieval methods have been reported to suffer from poor reconstruction accuracy and are subject to the initial conditions.[25,26] To overcome these challenges, we embed a padding with zero pixel value around the microstructures, which has been shown to significantly improve the reconstruction accuracy.[47] For a microstructure of dimension $n \times n$, we set a padding of n on each of the four sides. We now present the algorithm that consists of the following steps:

(a) Initialize the microstructure with random 0–1 values, $m_s^n(0)$.

(b) For a given microstructure $m_s^n(\tau)$ at iteration τ, obtain the corresponding Fourier transform, that is, $\mathcal{F}(m_s^n) = |M_k^n| e^{i\theta_k^n}$, where $|M_k^n|$ and θ_k^n are, respectively, the amplitude and phase of the Fourier transform. Here, we removed the iteration number τ to simplify the notations.

(c) Replace the amplitude of the Fourier transform in step (b) with the amplitude of the autocorrelation function ($\hat{\mathcal{X}}$) predicted in the previous step such that $\widetilde{\mathcal{F}(m_s^n)} = \sqrt{|\mathcal{F}(\hat{\mathcal{X}})|} e^{i\theta_k^n}$. Obtain the resulting microstructure using inverse Fourier transform, that is, $\widetilde{m_n^s} = \mathcal{F}^{-1}(\widetilde{\mathcal{F}(m_s^n)})$.

(d) Finally, we apply the constraints in the real space. Let us consider that Γ contains all the spatial locations where the local state violates the allowable values (e.g., the local state is negative). We update the microstructure in the next iteration as:

$$m_s^n(\tau+1) = \begin{cases} m_s^n(\tau) - \vartheta \widetilde{m_n^s}(\tau), & s \in \Gamma \\ \widetilde{m_n^s}(\tau), & s \notin \Gamma \end{cases},$$

where $0 < \vartheta < 1$ controls the rate of convergence. Additionally, if any of the local states exceed 1, then we reassign the state to 1.

(e) Repeat steps (b–d) until the reconstruction error defined as the Frobenius norm, $\|m_s^n(\tau) - m_n^s(\tau-1)\|_F$ between successive steps falls below a prespecified threshold.

Note that the amplitude substitution in step (b) is based on the property of autocorrelation and is a crucial step in ensuring that the guess microstructure gradually converges to the true microstructure. In particular, it says that the Fourier transform of autocorrelation is equal to the square of the magnitude of the corresponding microstructure function. The final step enforces the nonnegativity constraint before updating the microstructure function (i.e., $m_s^n(\tau) \geq 0$ at any iteration τ). A schematic of the hIO algorithm is presented in Figure 6.

Performance metrics

Microstructures are stochastic, and therefore, any predictions obtained from the phase-field emulator not only needs to be accurate, but also statistically correct. To assess the prediction accuracy of a microstructure, X, we will employ normalized mean squared error (NMSE) given as:

$$\text{NMSE} = \frac{||X - \hat{X}||^2}{\sqrt{||X||}}. \qquad 10$$

The average feature size[48] was measured by calculating the inverse of the first moment, $k_1(t)$,[49] which in turn, is obtained from the structure factor, $s(k, t)$ given as:

$$k_1(t) \equiv \frac{\sum_k k(t) s(k,t)}{\sum_k s(k,t)}. \qquad 11$$

See Reference 41 for details on the structure factor and the average feature size.

Acknowledgments

K.A. acknowledges funding from the National Science Foundation (NSF) under Grant Nos. CMMI-1763128 (A. Lewis and T. Kuech, Program Managers) and CAREER-2145812 (J. Yang and J. Madison, Program Managers).

Author contributions

A.S.I. conceived the project, developed the algorithm, and drafted and supervised the manuscript. P.W. performed the phase-field simulations and drafted the manuscript. A.S. implemented the algorithm for phase retrieval. K.A. conceived the project, drafted, and supervised the manuscript.

Competing interests

The authors declare no competing interests.

References

1. W.J. Boettinger, J.A. Warren, C. Beckermann, A. Karma, *Annu. Rev. Mater. Res.* **32**, 163 (2002)
2. Y. Jin, Y. Wang, A. Khachaturyan, *Appl. Phys. Lett.* **79**, 3071 (2001)
3. H. Henry, H. Levine, *Phys. Rev. Lett.* **93**, 105504 (2004)
4. D. Rodney, Y. Le Bouar, A. Finel, *Acta Mater.* **51**, 17 (2003)
5. N. Moelans, B. Blanpain, P. Wollants, *CALPHAD* **32**, 268 (2008)
6. Y. Wang, Y. Jin, A. Cuitino, A. Khachaturyan, *Appl. Phys. Lett.* **78**, 2324 (2001)
7. E.J. Kautz, *Patterns* **2**(7) 100285 (2021)
8. W.L. George, J.A. Warren, *J. Comput. Phys.* **177**, 264 (2002)
9. N. Provatas, K. Elder, *Phase-Field Methods in Materials Science and Engineering* (Wiley, Hoboken, 2010)
10. 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 *SC ’11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis* (Association for Computing Machinery, New York, 2011), pp. 1–11
11. X. Shi, H. Huang, G. Cao, X. Ma, *AIP Adv.* **7**(10), 105216 (2017)
12. E. Miyoshi, T. Takaki, M. Ohno, Y. Shibuta, S. Sakane, T. Shimokawabe, T. Aoki, *NPJ Comput. Mater.* **3**, 25 (2017)
13. H. Yan, K. Wang, J.E. Jones, *Model. Simul. Mater. Sci. Eng.* **24**, 055016 (2016)
14. A. Vondrous, M. Selzer, J. Hötzer, B. Nestler, *Int. J. High Perform. Comput. Appl.* **28**, 61 (2014)
15. A. Yamanaka, T. Aoki, S. Ogawa, T. Takaki, *J. Cryst. Growth* **318**, 40 (2011)
16. C. Lan, Y. Chang, C.-J. Shih, *Acta Mater.* **51**, 1857 (2003)
17. M. Greenwood, K.N. Shampur, N. Ofori-Opoku, T. Pinomaa, L. Wang, S. Gurevich, N. Provatas, *Comput. Mater. Sci.* **142**(3), 153 (2018)
18. J. Rosam, P. Jimack, A. Mullis, *Acta Mater.* **56**, 4559 (2008)
19. P. Bollada, C.E. Goodyer, P.K. Jimack, A.M. Mullis, F. Yang, *J. Comput. Phys.* **287**, 130 (2015)
20. W. Feng, P. Yu, S.Y. Hu, Z.K. Liu, Q. Du, L.Q. Chen, *J. Comput. Phys.* **220**(1), 498 (2006)
21. S. Poulsen, P. Voorhees, *Acta Mater.* **113**, 98 (2016)
22. T. Takaki, S. Sakane, M. Ohno, Y. Shibuta, T. Shimokawabe, T. Aoki, *Acta Mater.* **118**, 230 (2016)
23. S. DeWitt, S. Rudraraju, D. Montiel, W.B. Andrews, K. Thornton, *NPJ Comput. Mater.* **6**(1), 29 (2020)
24. K. Yang, Y. Cao, Y. Zhang, S. Fan, M. Tang, D. Aberg, B. Sadigh, F. Zhou, *Patterns* **2**(5), 100243 (2021)
25. D. Montes de Oca Zapiain, J.A. Stewart, R. Dingreville, *NPJ Comput. Mater.* **7**(1), 3 (2021)
26. E. Herman, J.A. Stewart, R. Dingreville, *Appl. Math. Model.* **88**, 589 (2020)

27. H. Frisch, F. Stillinger, *J. Chem. Phys.* **38**, 2200 (1963)

28. S.R. Niezgoda, A.K. Kanjarla, S.R. Kalidindi, *Integr. Mater. Manuf. Innov.* **2**, 54 (2013)

29. A. Çeçen, T. Fast, E. Kumbur, S. Kalidindi, *J. Power Sources* **245**, 144 (2014)

30. A. Gupta, A. Cecen, S. Goyal, A.K. Singh, S.R. Kalidindi, *Acta Mater.* **91**, 239 (2015)

31. V. Sundararaghavan, N. Zabaras, "Representation and Classification of Microstructures Using Statistical Learning Techniques," in *AIP Conference Proceedings* (American Institute of Physics, 2004), vol. 712, pp. 98–102

32. M.I. Latypov, M. Kühbach, I.J. Beyerlein, J.-C. Stinville, L.S. Toth, T.M. Pollock, S.R. Kalidindi, *Mater. Charact.* **145**, 671 (2018)

33. R.A. Harshman, et al. "Foundations of the PARAFAC Procedure: Models and Conditions for an 'Explanatory' Multimodal Factor Analysis," in *UCLA Working Papers in Phonetics* (University Microfilms, Ann Arbor, 1970), vol. 16, pp. 1–84

34. L.R. Tucker, *Psychometrika* **31**, 279 (1966)

35. I.V. Oseledets, *SIAM J. Sci. Comput.* **33**, 2295 (2011)

36. J.R. Fienup, *Appl. Opt.* **21**(15), 2758 (1982)

37. R. Gerchberg, W. Saxton, *Optik* **35**, 646 (1972)

38. L. Ma, E. Solomonik, "Fast and Accurate Randomized Algorithms for Low-Rank Tensor Decompositions," in *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (2021). https://openreview.net/forum?id=B4szfz7W7LU

39. V. Snyder, J. Alkemper, P. Voorhees, *Acta Mater.* **49**, 699 (2001)

40. H. Le Ferrand, M. Duchamp, B. Gabryelczyk, H. Cai, A. Miserez, *J. Am. Chem. Soc.* **141**, 7202 (2019)

41. A. Bray, C. Emmott, *Phys. Rev. B* **52**, R685 (1995)

42. S. Torquato, *Appl. Mech. Rev.* **55**(4), B62 (2002)

43. S.R. Kalidindi, "Statistical Quantification of Material Structure," in *Hierarchical Materials Informatics,* ed. by S.R. Kalidindi (Butterworth-Heinemann, Boston, 2015), p. 75

44. J.D. Carroll, J.-J. Chang, *Psychometrika* **35**, 283 (1970)

45. T.G. Kolda, B.W. Bader, *SIAM Rev.* **51**, 455 (2009)

46. J. Kossaifi, Y. Panagakis, A. Anandkumar, M. Pantic, *J. Mach. Learn. Res.* **20**, 925 (2019)

47. J.-P. Liu, *J. Opt. Soc. Am. A* **29**(9), 1956 (2012)

48. A. Kumar, D. Benjamin, R. Rahul, M. Amit, J.D. Michael, *J. Appl. Phys.* **126**, 075306 (2019)

49. C.G. Sharon, F.G. Mark, S. Francesco, C. Antonio, *Phys. Rev. E* **49**, 247 (1994) □

Publisher's note

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