Deep learning-driven prediction of microstructure evolution via latent space interpolation

Sachin Gaikwad [1,*] Thejas Kasilingam [2,*] Owais Ahmad [2] Rajdip Mukherjee [2] and Somnath Bhowmick [2,†]
[1]School of Materials Science and Technology, Indian Institute of Technology (BHU), Varanasi 221005, India
[2]Department of Materials Science & Engineering, Indian Institute of Technology, Kanpur, Kanpur 208016, India

(Received 23 June 2025; revised 3 September 2025; accepted 8 October 2025; published 29 October 2025)

Phase-field models accurately simulate microstructure evolution, but their dependence on solving complex differential equations makes them computationally expensive. This work achieves a significant acceleration via a deep learning-based framework, utilizing a conditional variational autoencoder (CVAE) coupled with cubic spline interpolation and spherical linear interpolation (SLERP). We demonstrate the method for binary spinodal decomposition by predicting microstructure evolution for intermediate alloy compositions from a limited set of training compositions. First, using microstructures from phase-field simulations of binary spinodal decomposition, we train the CVAE, which learns compact latent representations that encode essential morphological features. Next, we use cubic spline interpolation in the latent space to predict microstructures for any unknown composition. Finally, SLERP ensures smooth morphological evolution with time that closely resembles coarsening. The predicted microstructures exhibit high visual and statistical similarity to phase-field simulations. This framework offers a scalable and efficient surrogate model for microstructure evolution, enabling accelerated materials design and composition optimization.

DOI: 10.1103/5ngk-4v9j

I. INTRODUCTION

Artificial intelligence (AI) and machine learning (ML) have rapidly emerged as a transformative force across scientific disciplines, owing to the availability of high-performance computing and vast datasets generated over the past century. In the domain of materials science and engineering, AI and ML have demonstrated their potential over traditional methods by enabling automated analysis and prediction across scales [1,2]. Some prominent examples are interpreting and denoising microstructural images [3–5], linking atomic-scale imaging with material properties [6], materials discovery and manufacturing [7], and predicting early signs of material failure [8].

The focus of the current research is microstructure modeling. Among the physics-based computational approaches, the phase-field method stands out as a robust framework to simulate complex microstructural phenomena such as solidification [9,10], precipitate growth [11], grain growth [12,13], and spinodal decomposition [14–16]. These simulations are governed by partial differential equations that evolve spatially and temporally, requiring fine discretization and substantial computational resources. Efforts to reduce the computational overhead have led to the integration of numerical optimizations and parallel processing techniques [17,18].

Recent advances in deep learning have inspired a new class of data-driven surrogate models capable of approximating phase-field simulations with significantly reduced runtimes. These works mainly rely on generating the training data from phase-field simulations, compressing them to a low-dimensional latent space, followed by time series prediction models like recurrent neural networks (RNN), including the long short-term memory (LSTM) unit and the gated recurrent unit (GRU) to learn the microstructure evolution in latent space [19,20]. While these machine learning methods have shown great success in terms of accelerating the study of microstructure evolution, one needs to provide the models with some initial microstructures from the phase-field simulations to generate the subsequent microstructures. Moreover, due to error compounding, very long-term predictions are not possible using time series prediction models.

Generative models are also very popular in the field of microstructure modeling, particularly for learning microstructure features, generating synthetic microstructures, and predicting time evolution. While generative models are known for speed, efficiency, diversity, and statistical accuracy, they are more data and resource intensive. Models like generative adversarial network (GAN) [21,22] and denoising diffusion probabilistic models (DDPM) [23,24] have shown excellent potential in the field of microstructure modeling.

This work uses conditional variational autoencoder (CVAE) [25,26], a generative model which learns a latent representation of input microstructures, conditioned on composition (c_{avg}) values. To demonstrate our method, we deal with binary spinodal decomposition, a phase separation phenomenon that spontaneously separates a homogeneous mixture into two coexisting phases with different compositions. Spinodal decomposition is characterized by a time-dependent amplification of composition fluctuations, resulting in phase separation. It is observed in binary as well as in multicomponent materials, including metallic alloys,

*These authors contributed equally to this work.
†Contact author: bsomnath@iitk.ac.in

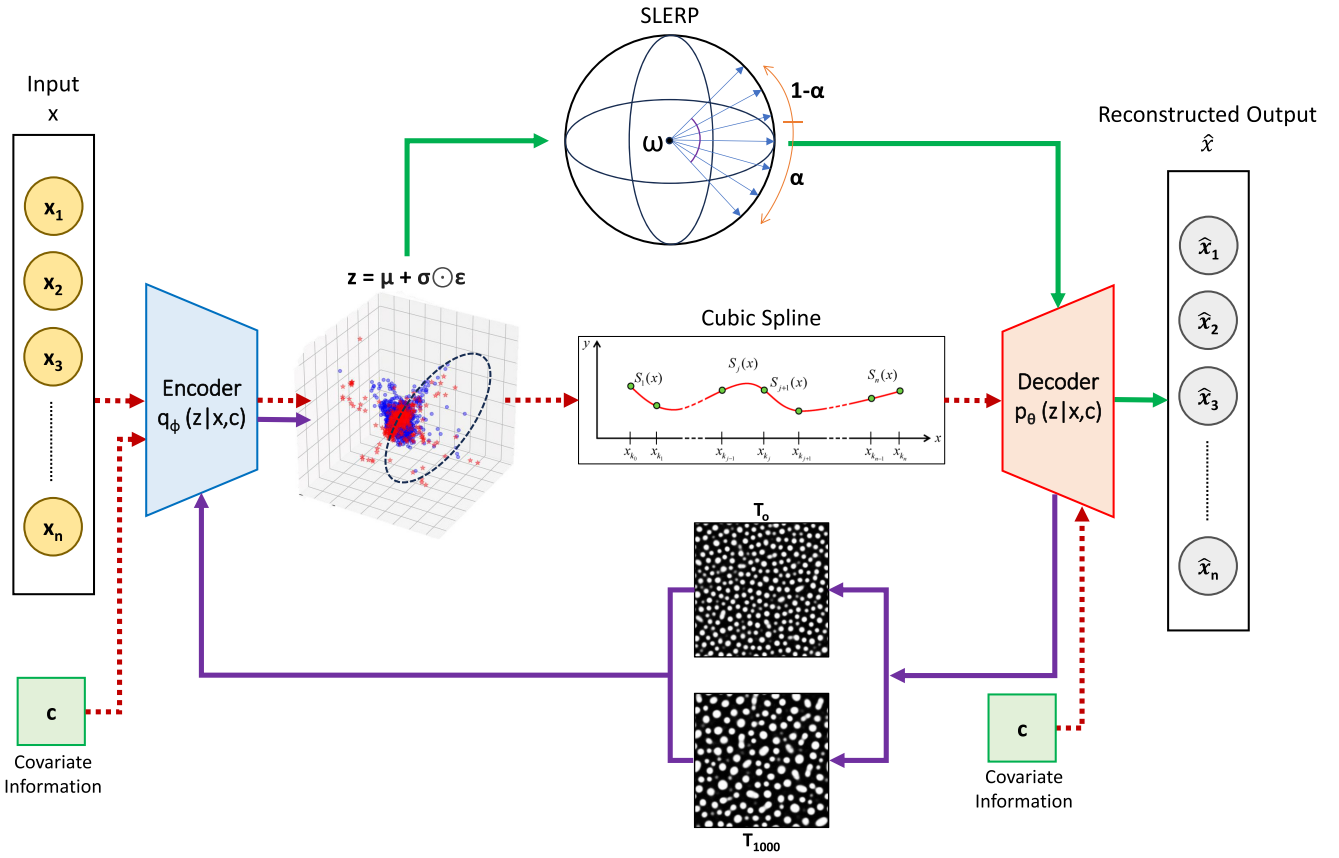


FIG. 1. Workflow of the conditional variational autoencoder (CVAE) framework for modeling microstructure evolution. The CVAE takes as input a set of microstructure images $x_1, x_2, ..., x_n$ along with their corresponding covariate information (c_{avg}). The encoder maps the inputs to a latent space $z = \mu + \sigma \odot \varepsilon$. The dotted red arrows represent the cubic spline interpolation workflow in the latent space, where representations for the targeted composition value are generated, which are then decoded to produce synthetic microstructures. The purple arrows show the selection process of latent vectors of images with the smallest and largest feature size (white shapes) and their injection back into the encoder. The green arrows depict the SLERP (spherical linearinterpolation) workflow, which operates between latent vectors corresponding to the microstructures with the largest and smallest feature size (white shapes) to generate a sequence of microstructures showing gradual and realistic morphological transitions of the targeted composition value.

polymers, and glasses. Unlike conventional phase transformations, it does not involve any nucleation barrier. Spinodal decomposition has practical significance in diverse fields, ranging from structural applications [27] and functional materials (e.g., permanent magnets) [28], to catalysis applications [29]. Extensive experimental and computational investigations have been reported in the literature on spinodal decomposition [30–37], with the computational studies primarily based on the pioneering theories of John Cahn and his co-workers [38–40]. While these simulations, particularly when implemented with parallel or GPU-based approaches, can model very large systems and provide statistically robust insights [41], the integration of machine learning offers the potential to further enhance their efficiency by reducing both computational cost and time [42–45].

In this study, the initial average compositions lie within the interval of $c_{avg} = 0.27$ to 0.48. The training set consists of 700 microstructures, each from nine selected compositions, 6300 images in total. Each grayscale microstructure has a dimension of $256 \times 256 \times 1$. Given the high dimensionality of the dataset, the computational burden becomes significant in terms of both memory and processing time. The encoder compresses the data while preserving its essential features [46,47] for further processing, which includes concatenating the input image with a learned label embedding, extracting features via convolutional layers, and mapping them to a lower-dimensional latent space with mean and log variance outputs. The reparametrization trick [25] is applied to generate latent vectors. The decoder reconstructs images by combining these vectors with class embeddings and upsampling via transposed convolution layers [48]. The loss function includes mean squared error (MSE) for reconstruction and a KL divergence loss [25], enforcing a structured latent space.

The workflow is illustrated in Fig. 1. After compressing the data with the encoder, we use two types of latent space interpolations. The dotted red arrows represent the cubic spline interpolation, which generates representations for the targeted composition value. The decoder produces synthetic microstructures; the one with the smallest and largest feature size is chosen and is injected back into the encoder (purple arrows). Finally, spherical linear interpolation (green arrows) operates between latent vectors corresponding to the microstructures with the largest and smallest feature size, generating a smooth and realistic morphological evolution for

a given targeted composition value. The rest of the paper provides technical details of various components of the workflow, followed by a discussion and a conclusion section.

II. PHASE-FIELD METHOD

To generate a dataset for training and validation, we simulate spinodal decomposition in an A-B binary alloy using a phase-field framework, as implemented in the μ2mech package [49]. The evolution of microstructure is driven by minimizing the total free energy [38]:

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

where $c(\mathbf{r}, t)$ represents the conserved composition field, κ is the gradient energy coefficient, and $f(c)$ is the bulk free energy density modeled as a double-well potential,

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

where W determines the height of the potential barrier. The free energy landscape favors phase separation when $\frac{\partial^2 f}{\partial c^2} < 0$, indicating thermodynamic instability. The time evolution of the composition field follows the Cahn-Hilliard equation:

$$\frac{\partial c}{\partial t} = M\left[\nabla^2\left(\frac{\partial f}{\partial c}\right) - 2\kappa\nabla^4 c\right]. \tag{3}$$

In the above equation, M is the mobility, taken as constant.

Nine different compositions are sampled, with initial average compositions of c_{avg} = 0.27, 0.31, 0.35, 0.39, 0.40, 0.42, 0.44, 0.46, 0.48. All simulations are run on a 2D square domain with a grid size of 256 × 256. Each simulation outputs a time series of microstructure images, evolved for 1000 time steps. However, due to the absence of significant features in the early evolution stages (t_0 to t_{300}), we discard these frames and use data from t_{301} onward, capturing the critical period of phase separation and domain coarsening. Thus, 700 phase-field microstructures are obtained per composition, totalling 6300 images. This dataset is used to train our machine learning model, as detailed in the following sections.

III. CONDITIONAL VARIATIONAL AUTOENCODER (CVAE) ARCHITECTURE

First, we need to map each image of size 256 × 256 × 1 to a lower-dimensional latent space that retains essential structural features while significantly reducing the data complexity. Dimensionality reduction techniques are widely used for this purpose, aiming to extract a compact set of features that retain the essential characteristics of the original data [46,47]. In this study, we adopt a conditional variational autoencoder (CVAE) [25,26], a generative model that learns a latent representation of the input microstructure images while being conditioned on additional variables, specifically the average concentration (c_{avg}) values. The CVAE consists of an encoder-decoder framework, wherein the encoder maps the input and the conditioning information to a structured latent space, and the decoder reconstructs the image from the sampled latent vectors and class embeddings. An overview of the CVAE architecture is illustrated in Fig. 2.

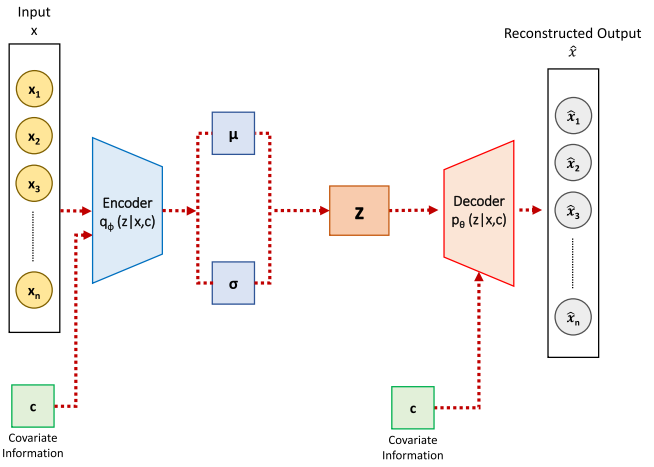


FIG. 2. Schematic representation of the conditional variational autoencoder (CVAE) architecture. The encoder maps the input x and covariate information c into a latent distribution parametrized by mean μ and standard deviation σ. A latent variable z is sampled and decoded back into the reconstructed output $\hat{x}$ while conditioning on c.

Encoder network. The encoder takes an input image $x \in \mathbb{R}^{H\times W\times C}$ along with its corresponding condition label $y \in \mathbb{R}^{N_c}$, where H, W, and C represent the height, width, and number of channels of the image, respectively, and N_c is the number of conditioning labels. The label is first embedded and reshaped to match the image dimensions,

$$y' = \text{Reshape}(\text{Dense}(H \times W)(y)). \tag{4}$$

Then, the input image and the label embedding are concatenated:

$$z_{\text{input}} = \text{Concatenate}(x, y'). \tag{5}$$

Feature extraction is performed using a series of convolutional layers followed by max-pooling operations, resulting in a feature representation z_{conv}. The encoder then learns a Gaussian distribution in the latent space by computing the mean μ and log variance $\ln\sigma^2$:

$$\mu = \text{Dense}\left(\prod_{i=1}^{d} L_i\right)(z_{\text{conv}}), \tag{6}$$

$$\ln\sigma^2 = \text{Dense}\left(\prod_{i=1}^{d} L_i\right)(z_{\text{conv}}), \tag{7}$$

where L_i represents the dimensions of the latent space.

Latent space sampling. To allow backpropagation, the reparametrization trick is applied, where a random variable $\epsilon \sim \mathcal{N}(0, I)$ is introduced:

$$z = \mu + \exp\left(\frac{\ln\sigma^2}{2}\right) \cdot \epsilon. \tag{8}$$

Decoder network. The decoder reconstructs the original image given the latent variable z and the conditional label y. The label is embedded similarly to the encoder:

$$y'' = \text{Reshape}\left(\text{Dense}\left(\prod_{i=1}^{d} L_i\right)(y)\right). \tag{9}$$

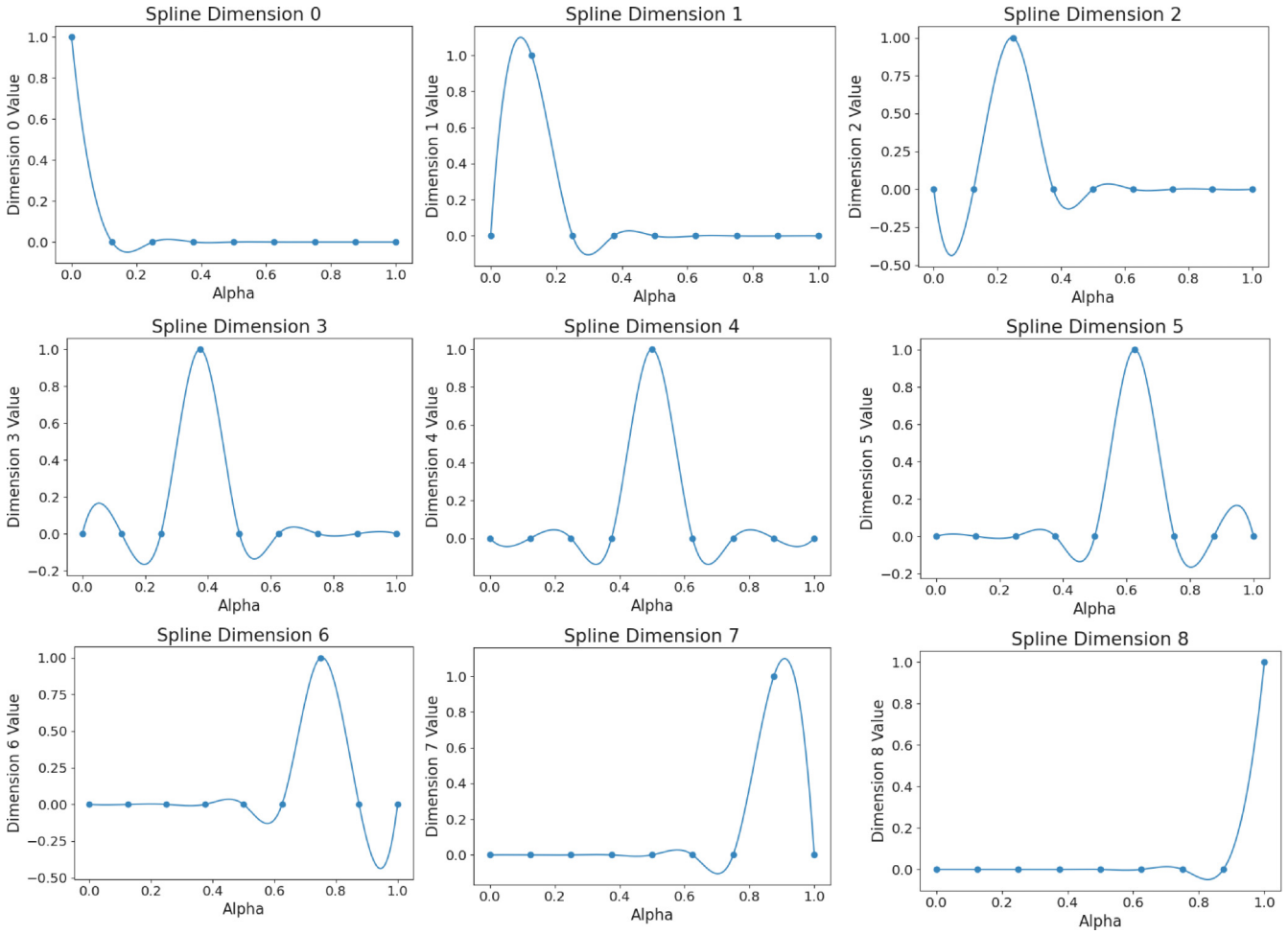


FIG. 3. Visualization of cubic spline interpolation applied to the conditional label vectors in a nine-dimensional label space. Each subplot illustrates the smooth variation of one label dimension as a function of the interpolation parameter "alpha," constructed from known one-hot encoded composition vectors corresponding to alloy compositions between 0.27 and 0.48. The peaks in each dimension reflect dominant contributions. These splines enable continuous and physically meaningful transitions between known compositions, thereby facilitating smooth label-guided image synthesis within the CVAE framework.

The decoder input is then formed by concatenating z with y'':

$$z_{\text{dec}} = \text{Concatenate}(z, y''). \tag{10}$$

This representation is passed through fully connected and transposed convolutional layers to reconstruct the image $\hat{x}$:

$$\hat{x} = f_\theta(z_{\text{dec}}). \tag{11}$$

Loss function. The CVAE model is trained using a combination of a reconstruction loss and a Kullback-Leibler (KL) divergence loss. The reconstruction loss is computed as the mean squared error (MSE) between the original and reconstructed images:

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2. \tag{12}$$

The KL divergence loss ensures that the learned latent distribution approximates a standard normal distribution:

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2}\sum_{j=1}^{d}(1 + \ln\sigma^2 - \mu^2 - \sigma^2). \tag{13}$$

The total loss function is given by

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda\mathcal{L}_{\text{KL}}, \tag{14}$$

where λ is a weighting factor.

Training process. The CVAE model is trained using the Adam optimizer with an adaptive learning rate. Stratified sampling is applied for dataset splitting to ensure balanced learning, with an $80-20$ train-validation ratio. Integer-encoded labels are used to preserve the class distribution of the images, as required by the conditional variational autoencoder (CVAE). The training process incorporates early stopping to prevent overfitting and learning rate scheduling to enhance convergence.

IV. CUBIC SPLINE INTERPOLATION

To generate intermediate microstructural states between known phase compositions, we employ cubic spline interpolation [50] (see Fig. 3), which ensures smooth transitions between discrete phase labels. This method is particularly

useful in maintaining consistency in phase fraction variations while interpolating between microstructural states.

Mathematical formulation. Given a set of discrete phase fraction labels,

$$\{y_0, y_1, \ldots, y_n\}, \tag{15}$$

we construct a cubic spline function $S(x)$ that smoothly interpolates between these values. Each spline segment is defined as

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad x_i \leqslant x \leqslant x_{i+1}, \tag{16}$$

where a_i, b_i, c_i, d_i are coefficients computed to ensure:

(i) Continuity of the function: $S_i(x_i) = y_i$;

(ii) Continuity of the first derivative: $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$;

(iii) Continuity of the second derivative: $S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$;

(iv) Natural boundary conditions, where the second derivative is set to zero at the endpoints.

The dataset consists of nine discrete phase fraction values, and we define the interpolation function over normalized positions $x \in [0, 1]$:

$$\{x_0, x_1, ..., x_8\}, \quad \{y_0, y_1, ..., y_8\}. \tag{17}$$

The spline function $S(x)$ smoothly interpolates between these labels, generating intermediate phase fractions.

Selection of labels. We select the labels based on target phase composition. To generate synthetic images with a specific target average phase composition c_{avg}, we define a threshold tolerance:

$$|\text{rounded\_avg} - \text{target\_rounded\_avg}| \leqslant \text{tolerance}. \tag{18}$$

Based on the target composition range, the corresponding interpolation interval $[x_i, x_{i+1}]$ is selected from a predefined mapping:

$$(x_{\text{start}}, x_{\text{end}}) = \begin{cases} (0.27, 0.29), & 0.27 \leqslant c_{\text{avg}} < 0.29, \\ (0.29, 0.31), & 0.29 \leqslant c_{\text{avg}} < 0.31, \\ (0.31, 0.33), & 0.31 \leqslant c_{\text{avg}} \leqslant 0.33, \\ \vdots & \vdots \\ (0.47, 0.48), & 0.47 \leqslant c_{\text{avg}} < 0.48. \end{cases} \tag{19}$$

A set of interpolated labels is then sampled from $S(x)$ within this range.

Image generation. Once the interpolated labels are obtained, we generate images by sampling a Gaussian latent vector:

$$z \sim \mathcal{N}(0, I). \tag{20}$$

The interpolated label is converted into a tensor and passed into the CVAE decoder, which reconstructs the corresponding microstructure:

$$\hat{x} = h_\phi(z, \tilde{y}), \tag{21}$$

where $\tilde{y}$ is the interpolated phase label.

To ensure the generated image has the desired c_{avg}, we compute the normalized average pixel intensity:

$$\bar{c} = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{ij}, \tag{22}$$

where H and W are the image dimensions. If $|\bar{c} - c_{\text{avg}}| \leqslant$ tolerance, the image is accepted; otherwise, the process is repeated.

Morphological characterization. We perform further image analysis to assess the microstructural variations in the dataset generated via cubic spline interpolation. This analysis aims to arrange images in terms of the size of the microstructural features. First, each generated (from cubic spline interpolation) microstructure image is converted to a binary format using thresholding:

$$B(I) = \begin{cases} 255, & \text{if } I(x, y) \geqslant 127, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

Contours corresponding to distinct microstructural regions are extracted using OpenCV's contour detection algorithm. Given N_c detected contours in an image, the total shape area is computed as

$$A_{\text{total}} = \sum_{i=1}^{N_c} A_i, \tag{24}$$

where A_i represents the area of the ith detected microstructural shape. Finally, the average feature (denoted by white shapes) size per image is calculated as

$$A_{\text{avg}} = \begin{cases} \frac{A_{\text{total}}}{N_c}, & N_c > 0, \\ 0, & \text{if } N_c = 0. \end{cases} \tag{25}$$

The average feature size for a given composition increases with time because of coarsening. We rank the images (generated from cubic spline interpolation) based on their computed feature size A_{avg}. Three key images are identified:

(i) Smallest average feature size: $I_{\text{min}} = \arg\min A_{\text{avg}}$;

(ii) Largest average feature size: $I_{\text{max}} = \arg\max A_{\text{avg}}$;

(iii) Medium average feature size: I_{mid}, corresponding to the median rank in the sorted dataset.

These three images are further used to get the time evolution of microstructures.

Note that while cubic spline interpolation generates microstructure images for a targeted composition value, it does not ensure smooth morphological transitions with time. For a given target composition, between the latent vectors of images with the smallest and largest feature size, as obtained from the cubic interpolation, SLERP (spherical linear interpolation) is applied to generate intermediate steps. This method ensures smooth morphological transitions with time, very similar to the phase-field time evolution. Thus, both the interpolation methods are necessary and complementary. Further details about SLERP are given in the following section.

V. SPHERICAL LINEAR INTERPOLATION (SLERP) FOR MICROSTRUCTURE EVOLUTION

To capture the smooth evolution of microstructures between different morphological states, we employ spherical

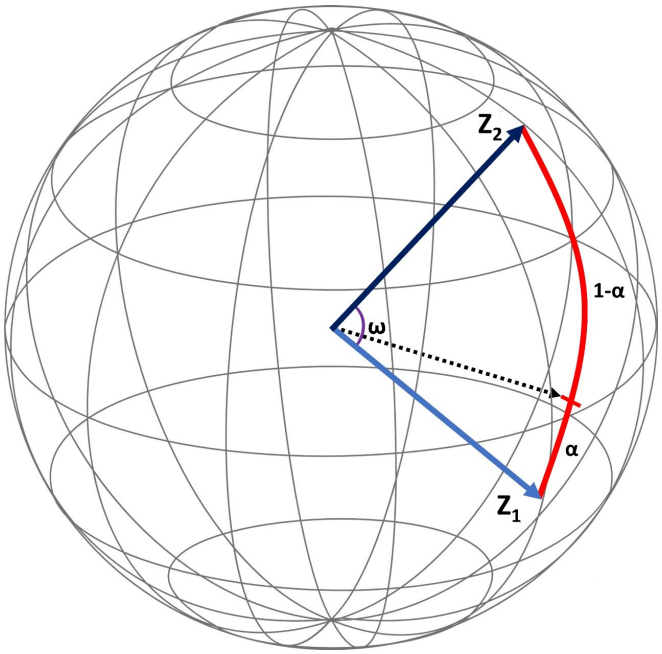


FIG. 4. Illustration of spherical linear interpolation (SLERP). Given two latent vectors z_1 and z_2, SLERP computes an intermediate point along the geodesic path on the unit hypersphere. The interpolation factor α determines the position of the generated latent vector, ensuring smooth and consistent transitions in the latent space.

linear interpolation (SLERP) [51] in the latent space of the conditional variational autoencoder (CVAE). This method (see Fig. 4) ensures a geodesic transition between latent representations, preventing distortions caused by linear interpolation in high-dimensional spaces.

Mathematical formulation. Given two latent vectors z_1 and z_2 in a high-dimensional space, the spherical linear interpolation (SLERP) function computes an intermediate latent representation $z_{\text{interpolated}}$ at a specific interpolation factor α:

$$z_{\text{interpolated}}(\alpha) = \frac{\sin((1-\alpha)\omega)}{\sin\omega} z_1 + \frac{\sin(\alpha\omega)}{\sin\omega} z_2, \tag{26}$$

where the angular distance between the two latent vectors is

$$\omega = \arccos\left(\frac{z_1 \cdot z_2}{\|z_1\|\|z_2\|}\right). \tag{27}$$

The interpolation parameter α varies in the range $\alpha \in [0, 1]$, where

(i) $\alpha = 0$: returns the initial latent vector z_1;

(ii) $\alpha = 1$: returns the final latent vector z_2;

(iii) $0 < \alpha < 1$: yields smooth intermediate latent vectors.

Image generation. As mentioned in Sec. IV, we first identify three key images: $I_{\text{min}}, I_{\text{mid}}, I_{\text{max}}$. Next, we perform SLERP-based latent space interpolation in two stages:

(i) Between I_{min} and I_{mid} to generate intermediate images covering the transition from small to medium feature size;

(ii) Between I_{mid} and I_{max} to generate intermediate images covering the transition from medium to large feature size.

The above method ensures that the entire transition from small to large feature size is captured smoothly for a given target composition.

To obtain latent representations, the CVAE encoder extracts the mean and variance of the Gaussian distribution for the terminal images. For example, in the case of I_{min} and I_{mid},

$$\mu_1, \ln\sigma_1^2 = f_\theta(I_{\text{min}}, y_1), \tag{28}$$
$$\mu_2, \ln\sigma_2^2 = f_\theta(I_{\text{mid}}, y_2).$$

Latent vectors z_1 and z_2 are sampled using the reparametrization trick

$$z_1 = \mu_1 + \exp\left(\frac{\ln\sigma_1^2}{2}\right)\cdot\epsilon, \tag{29}$$
$$z_2 = \mu_2 + \exp\left(\frac{\ln\sigma_2^2}{2}\right)\cdot\epsilon.$$

Interpolated latent representations are computed using SLERP at multiple values of α:

$$z_{\text{interpolated}} = \text{SLERP}(\alpha, z_1, z_2), \tag{30}$$

where α is sampled from a uniform distribution between 0 and 1 to generate a sequence of smooth transitions.

Finally, each interpolated latent representation $z_{\text{interpolated}}$ is passed through the CVAE decoder to reconstruct microstructures:

$$\hat{x} = h_\phi(z_{\text{interpolated}}, y_{\text{interpolated}}), \tag{31}$$

where $y_{\text{interpolated}}$ is the interpolated phase label, computed as

$$y_{\text{interpolated}} = (1-\alpha)y_1 + \alpha y_2. \tag{32}$$

Morphological characterization. To ensure physical consistency, we evaluate the generated images based on their average feature area A_{avg}, as described in Eq. (24) and Eq. (25). The interpolation process is accepted provided the following conditions are satisfied:

(i) Absolute difference between the average area of the first generated image and I_{min} is less than 2, i.e., ;

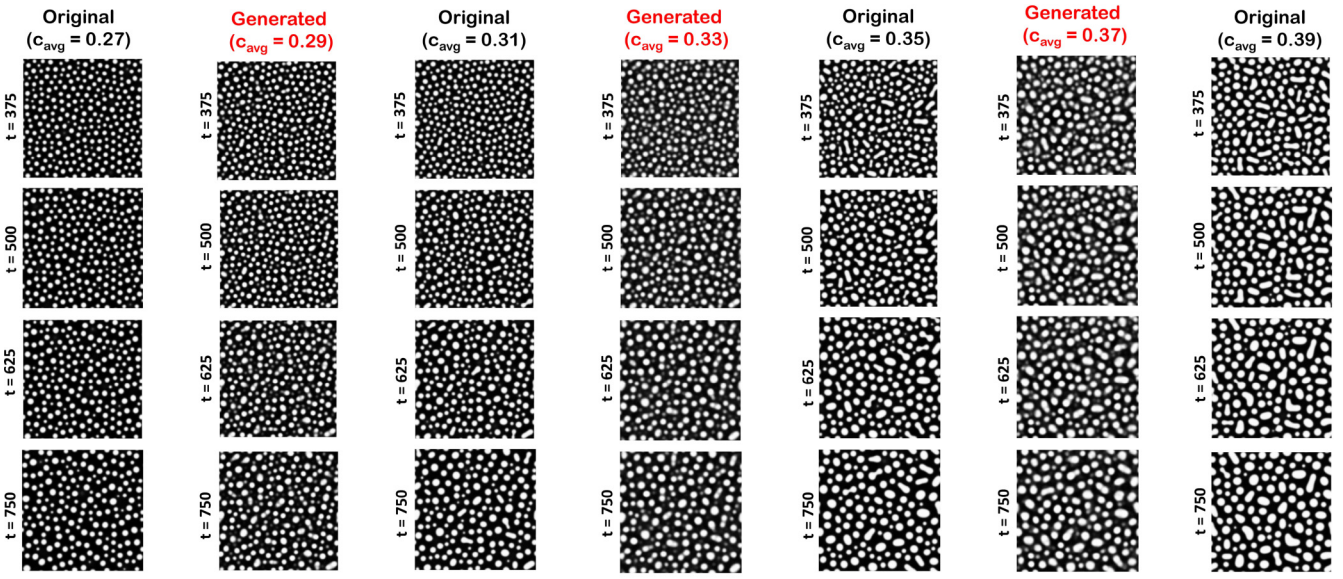
(ii) Absolute difference between the average area of the last generated image and I_{max} is less than 2, i.e., $|A_{\text{gen,last}} - A_{\text{max}}| < 2$.

If any one of the conditions are not met, the present set of generated microstructures is discarded and SLERP is rerun to generate another set of images. This process ensures morphological consistency in the microstructure evolution process. Model performance is analyzed in detail in the following section.

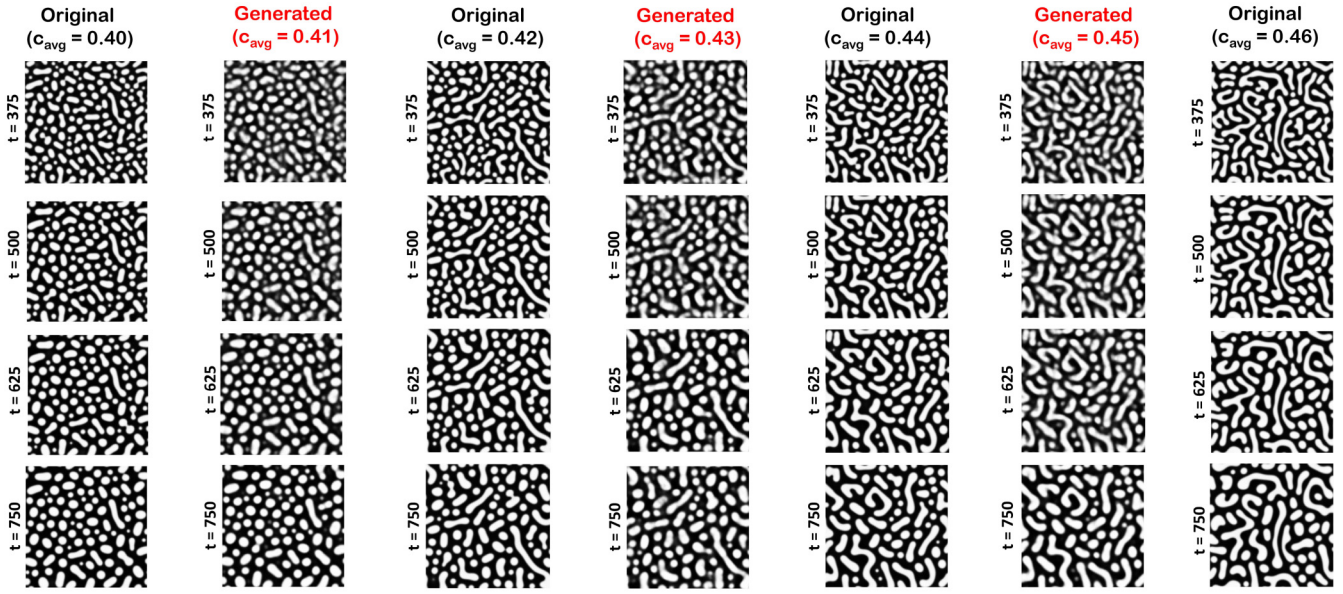
VI. RESULTS AND DISCUSSIONS

So far, we have shown a step-by-step development of a conditional variational autoencoder (CVAE) model for generating microstructure images conditioned on varying compositions. Cubic spline and spherical linear interpolation in the latent space generate the composition and time variation. The method predicts microstructure evolution for intermediate targeted composition values, based on a training dataset comprising a limited number of compositions. This results in a significant computational speedup compared to the case of spanning the entire composition range via the phase-field approach.

It takes ≈60 minutes to train the model on an Intel® Xeon® Gold 6338N CPU (2.20 GHz), supported by 128 GB of RAM.



(a) $c_{\text{avg}} = 0.27 \rightarrow 0.39$



(b) $c_{\text{avg}} = 0.40 \rightarrow 0.46$

FIG. 5. Comparison of original and CVAE-generated microstructure evolution for two composition intervals: (a) $0.27 \rightarrow 0.39$ and (b) $0.40 \rightarrow 0.46$. The black areas represent the A-rich phase, and the white areas represent the B-rich phase.

The model is faster than conventional numerical techniques for solving the Cahn-Hilliard equation. While the ML-assisted approach requires only ≈3 minutes to generate 1000 frames of 256×256 grid-size microstructure, showing time evolution for a given composition, the traditional numerical technique using phase-field calculation takes almost ≈12 minutes for the same. For a larger grid size, the time taken by the phase-field method to generate 1000 frames is even higher, for example, 1.5x for 512×512 microstructures, and 5x for 1024×1024 microstructures.

Figure 5 and Figs. S1–S3 (Supplemental Material (SM) [52]) illustrate the original training compositions and the generated intermediate target compositions obtained by deep learning. In particular, Fig. 5 shows microstructure evolution for two composition intervals: (i) 0.27–0.39 and (ii) 0.40–0.46. These compositions are initial scaled compositions of the alloys prior to spinodal decomposition (or equivalently, average alloy compositions at all times). Based on the bulk free energy expression [Eq. (2)], the equilibrium compositions of the A-rich (black) and B-rich (white) phases are 0 and 1, respectively. For the bulk free energy, we employed a double-well potential with two degenerate minima corresponding to the equilibrium A-rich and B-rich phases [Eq. (2)]. Consequently, both phases (A-rich black and B-rich white) possess equal energies (zero) and, importantly, the same chemical potentials. This ensures equilibrium across a flat interface. Following spinodal decomposition, the two phases undergo coarsening over time to minimize the total interfacial energy,

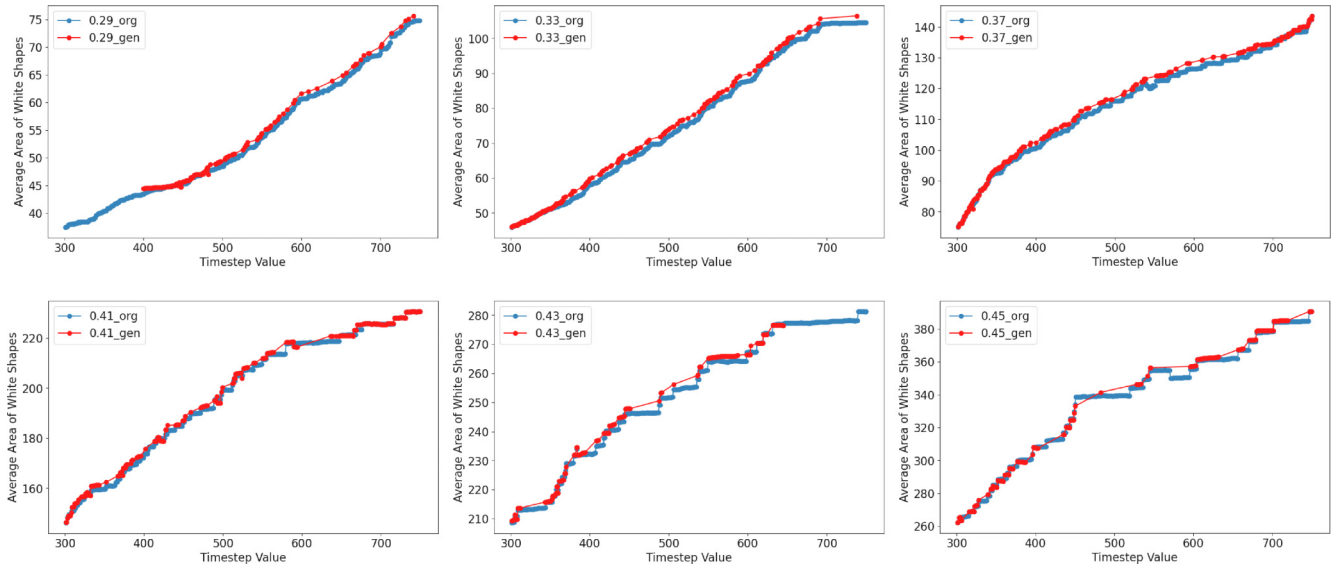


FIG. 6. Timestep vs average area of the B-rich white phase (equivalent to average particle size) during the microstructure evolution. Each subplot corresponds to a distinct composition: the blue curve represents the original microstructure, and the red curve denotes the generated counterpart.

which acts as the global driving force. Locally, the driving force for this coarsening is governed by the curvature of the interfaces. The temporal evolution of the size distribution of the B-rich white phase is illustrated in Figs. S4 and S5 in the SM [52]. The peak of the distribution diminishes while the distribution shifts toward larger sizes, which are the characteristic features of the coarsening process.

From Fig. 5, it is clear that CVAE can produce high-resolution images showing smooth time evolution of microstructures, very similar to actual phase-field simulations. After visually confirming our model's effectiveness in generating phase-field-like microstructure evolution, let us perform a more quantitative analysis across two key metrics: the temporal evolution of feature size and spatial correlation similarity.

Figure 6 compares the CVAE-generated microstructure evolution and the phase-field model evolution across different compositions. The horizontal axis represents timestep values, while the vertical axis shows the average area (equivalent to average particle size) of the features (B-rich white phase) in the microstructure at each timestep. The area increases with time because of coarsening. The blue curves represent the microstructure evolution via the phase-field model, and the red curves represent the same via CVAE-based deep learning. We observe that for all compositions tested, both models exhibit similar trends of coarsening over time. The close alignment between the two curves indicates that the CVAE model can learn and replicate the temporal dynamics of microstructure evolution like traditional phase-field simulations. This is a significant finding, as it demonstrates that the model can generate physically plausible microstructure evolution, closely resembling the actual phase-field simulations. Additionally, the model's ability to create microstructures with varying particle sizes across different compositions highlights its potential for use in material design and alloy optimization, where controlling particle size and distribution is critical.

Figure 7 compares the two-point autocorrelation function [53] between the phase-field and CVAE-generated microstructures. The spatial arrangement of features in the CVAE-generated and phase-field microstructures is compared using the two-point autocorrelation function $S_2(r)$, as shown in Fig. 7. For this analysis, microstructures with equivalent composition and average particle size are utilized. The function $S_2(r)$ represents the probability that two points separated by a distance r lie within the same phase and is mathematically expressed as follows:

$$S_2(r) = \frac{1}{N(r)} \sum_{i=1}^{N(r)} \delta(x_i)\delta(x_i + r). \tag{33}$$

It is calculated as the normalized sum of products of an indicator function, $\delta(x)$, where $\delta(x)$=1 if point x is in the phase of interest and 0 otherwise. The sum is taken over all point pairs $(x_i, x_i + r)$ and divided by the total number of pairs, $N(r)$, at that separation.

The autocorrelation function measures the spatial similarity between microstructure features at varying distances, which indicates grain distribution and phase homogeneity in the material. As shown in Fig. 7, the blue lines represent the autocorrelation functions for the phase-field model, and the red lines represent the autocorrelation functions for the CVAE-generated microstructures. The spatial decay patterns are similar for each composition between the two models. This suggests that the CVAE-generated microstructures not only follow the same evolutionary trends over time but also exhibit comparable spatial structures to the phase-field microstructures.

Note that the minimum point of the two-point autocorrelation function corresponds to the average feature size of the B-rich white phase. Figure 8 illustrates the minimum point of the two-point autocorrelation function (Fig. 7) as a function of composition. With increasing average composition, the average size of the B-rich white phase increases. As a result,

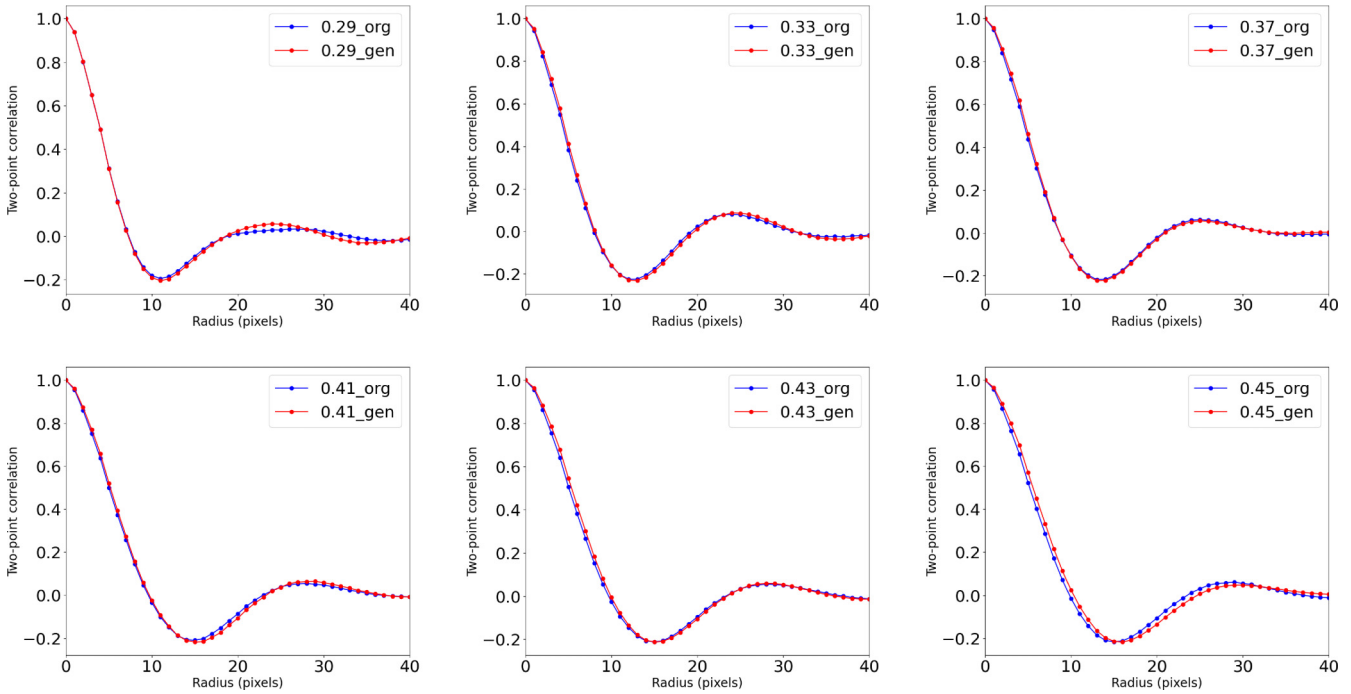


FIG. 7. Two-point autocorrelation compared at the 750th timestep between original and generated microstructures. Each subplot corresponds to a distinct composition, where the blue (red) curve represents the two-point autocorrelation for the original phase-field (CVAE-generated) microstructure. The high degree of overlap across all subplots indicates that the generative model successfully captures the spatial correlation characteristics of the original data.

the minimum point of the two-point autocorrelation function shifts to a higher value. The overall similarity of the two-point autocorrelation analysis between the original and generated microstructures further proves the success of the CVAE model for generating realistic microstructures for material science applications, where understanding spatial correlations is key to predicting material properties.

VII. CONCLUSIONS

This study demonstrates the application of a conditional variational autoencoder (CVAE) for generating microstructure images conditioned on varying alloy compositions. The use of latent space interpolation techniques (cubic spline and SLERP) enables the model to generate intermediate microstructures across varying compositions, making the model very useful in exploring a large number of alloy compositions in a short time. A detailed comparison between phase-field and deep-learned microstructures proves that the CVAE model can replicate the overall characteristics of microstructure evolution, including both temporal and spatial features. Key findings from this work include the following:

(i) The timestep vs average area plot (Fig. 6) confirms that the CVAE model effectively generates microstructures that evolve similarly to phase-field simulations, with corresponding grain coarsening patterns across timesteps;

(ii) The two-point autocorrelation comparison (Fig. 7) demonstrates that the CVAE model generates spatially realistic microstructures similar to phase-field simulations, highlighting the deep-learning model's potential for capturing spatial distribution patterns of particles;

(iii) The model's ability to generate physically plausible microstructures opens up new possibilities for material design and alloy optimization, where understanding and controlling microstructure at different stages of evolution is critical;

(iv) Overall, the results affirm the effectiveness of the CVAE model in generating realistic microstructures with temporal and spatial features comparable to phase-field simulations, making it a promising tool for materials informatics and design optimization.

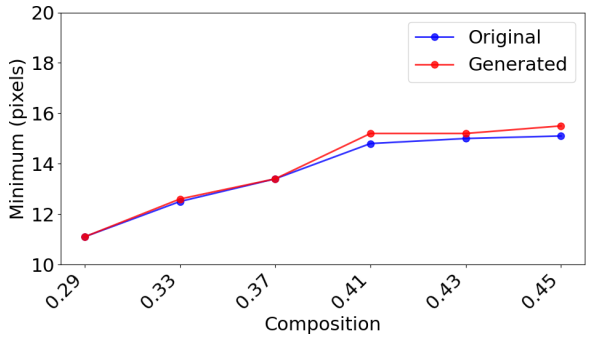


FIG. 8. The radius pixel value at which the two-point correlation value (taken from Fig. 7) is minimum for $c_{avg} = 0.29$, 0.33, 0.37, 0.41, 0.43, and 0.45, comparing the original (blue) and generated (red) microstructures.

ACKNOWLEDGMENTS

We acknowledge the National Super Computing Mission (NSM) for providing computing resources of "PARAM

Sanganak" at IIT Kanpur, which is implemented by CDAC and supported by the Ministry of Electronics and Information Technology (MeitY) and Department of Science and Technology (DST), Government of India. We also thank the ICME National Hub, IIT Kanpur, and CC, IIT Kanpur, for providing the HPC facility.

DATA AVAILABILITY

The data that support the findings of this article are openly available [54] because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

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

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

[3] B. L. DeCost, T. Francis, and E. A. Holm, Exploring the microstructure manifold: Image texture representations applied to ultrahigh carbon steel microstructures, Acta Mater. **133**, 30 (2017).

[4] E. A. Holm, R. Cohn, N. Gao, A. R. Kitahara, T. P. Matson, B. Lei, and S. R. Yarasi, Overview: Computer vision and machine learning for microstructural characterization and analysis, Metall. Mater. Trans. A **51**, 5985 (2020).

[5] O. Ahmad, A. Linda, S. R. Jha, and S. Bhowmick, A robust method of denoising experimental micrographs using deep learning, Mater. Charact. **223**, 114963 (2025).

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

[7] I. Papadimitriou, I. Gialampoukidis, S. Vrochidis, and I. Kompatsiaris, AI methods in materials design, discovery and manufacturing: A review, Comput. Mater. Sci. **235**, 112793 (2024).

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

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

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

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

[12] C. E. Krill, III, L. Helfen, D. Michels, H. Natter, A. Fitch, O. Masson, and R. Birringer, Size-dependent grain-growth kinetics observed in nanocrystalline Fe, Phys. Rev. Lett. **86**, 842 (2001).

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

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

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

[16] R. Chafle, S. Bhowmick, and R. Mukherjee, Domain boundary assisted spinodal decomposition in magnetic materials, Mater. Lett. **324**, 132630 (2022).

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

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

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

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

[21] G. N. Haribabu, J. T. Jegadeesan, C. Bhattacharya, and B. Basu, A deep adversarial approach for the generation of synthetic titanium alloy microstructures with limited training data, Comput. Mater. Sci. **230**, 112512 (2023).

[22] O. Ahmad, V. Panwar, K. Das, R. Mukherjee, and S. Bhowmick, Scope of generative artificial intelligence in microstructural studies: a case study, Phys. Scr. **100**, 076020 (2025).

[23] E. Azqadan, H. Jahed, and A. Arami, Predictive microstructure image generation using denoising diffusion probabilistic models, Acta Mater. **261**, 119406 (2023).

[24] A. Bentamou, S. Chretien, and Y. Gavet, 3D denoising diffusion probabilistic models for 3D microstructure image generation of fuel cell electrodes, Comput. Mater. Sci. **248**, 113596 (2025).

[25] D. P. Kingma and M. Welling, Auto-Encoding Variational Bayes, in *Proceedings of 2nd International Conference on Learning Representations (ICLR 2014) Banff, Canada* (University of Amsterdam, 2014).

[26] K. Sohn, H. Lee, and X. Yan, Learning structured output representation using deep conditional generative models, in *Advances in Neural Information Processing Systems*, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Curran Associates, Inc., New York, 2015), Vol. 28.

[27] J. K. Yang, C. L. Zhang, H. Zhang, J. Li, J. Y. Zhang, J. Kuang, G. Liu, and J. Sun, Spinodal decomposition-mediated multi-architectured α precipitates making a metastable β-Ti

alloy ultra-strong and ductile, J. Mater. Sci. Technol. **191**, 106 (2024).

[28] Y. Liu, M. Okugawa, K. Saito, D. Izumikawa, C. Zhu, Y. Seguchi, and Y. Koizumi, Phase-field modeling of spinodal decomposition in Fe–Cr–Co alloy under continuous temperature-changing conditions, ISIJ International **63**, 1245 (2023).

[29] L. Lu, M. Sun, A. Zhi, H. Ling, Y. Lan, H. Han, J. Wang, X. Zhang, Y. Zhao, M. Li, L. Cai, X. Li, X. Bai, and W. Wang, Spinodal decomposition enables coherent plasmonic metal/semiconductor heterostructure for full spectrum photocatalysis, Nat. Commun. **16**, 6479 (2025).

[30] M. P. Gururajan and T. A. Abinandanan, Phase inversion in two-phase solid systems driven by an elastic modulus mismatch, Philos. Mag. **87**, 5279 (2007).

[31] K. N. Sasidhar, M. P. Gururajan, and S. R. Meka, Concentration-gradient-driven anisotropic spinodal decomposition kinetics: Nitriding of metallic alloys, Philos. Mag. Lett. **101**, 432 (2021).

[32] S. Sugathan and S. Bhattacharyya, A phase-field study of elastic stress effects on phase separation in ternary alloys, Comput. Mater. Sci. **172**, 109284 (2020).

[33] P. Pankaj, S. Bhattacharyya, and S. Chatterjee, Surface-directed and bulk spinodal decomposition compete to decide the morphology of bimetallic nanoparticles, Modell. Simul. Mater. Sci. Eng. **31**, 015003 (2023).

[34] C.-L. Park, J. W. Gibbs, P. W. Voorhees, and K. Thornton, Coarsening of complex microstructures following spinodal decomposition, Acta Mater. **132**, 13 (2017).

[35] Y. Ishiguro, Y. Tsukada, and T. Koyama, Phase-field simulation of spinodal decomposition and its effect on stress-induced martensitic transformation in Ti–Nb–O alloys, Comput. Mater. Sci. **151**, 222 (2018).

[36] G. Radnóczi, E. Bokányi, Z. Erdélyi, and F. Misják, Size dependent spinodal decomposition in Cu-Ag nanoparticles, Acta Mater. **123**, 82 (2017).

[37] Y. Zhang, Z. Chen, D. Cao, J. Zhang, P. Zhang, Q. Tao, and X. Yang, Concurrence of spinodal decomposition and nano-phase precipitation in a multi-component AlCoCrCuFeNi high-entropy alloy, J. Mater. Res. Technol. **8**, 726 (2019).

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

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

[40] J. W. Cahn, On spinodal decomposition in cubic crystals, Acta Metall. **10**, 179 (1962).

[41] X. Shi, H. Huang, G. Cao, and X. Ma, Accelerating large-scale phase-field simulations with GPU, AIP Adv. **7**, 105216 (2017).

[42] P. Wu, A. S. Iquebal, and K. Ankit, Emulating microstructural evolution during spinodal decomposition using a tensor decomposed convolutional and recurrent neural network, Comput. Mater. Sci. **224**, 112187 (2023).

[43] O. Ahmad, N. Kumar, R. Mukherjee, and S. Bhowmick, Accelerating microstructure modeling via machine learning: A method combining autoencoder and ConvLSTM, Phys. Rev. Mater. **7**, 083802 (2023).

[44] O. Ahmad, R. Maurya, R. Mukherjee, and S. Bhowmick, Integrated phase field and machine learning study of microstructure evolution during interface-controlled spinodal decomposition, Solid State Phenomena **357**, 101 (2024).

[45] S. Tiwari, P. Satpute, and S. Ghosh, Time series forecasting of multiphase microstructure evolution using deep learning, Comput. Mater. Sci. **247**, 113518 (2025).

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

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

[48] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society, Los Alamitos, California, 2019).

[49] A. Linda, A. S. Negi, V. Panwar, R. Chafle, S. Bhowmick, K. Das, and R. Mukherjee, μ2mech: A software package combining microstructure modeling and mechanical property prediction, Phys. Scr. **99**, 055256 (2024).

[50] W. H. Press, S, A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 2007).

[51] K. Shoemake, Animating rotation with quaternion curves, in *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '85* (Association for Computing Machinery, New York, NY, 1985), pp. 245–254.

[52] See Supplemental Material http://link.aps.org/supplemental/10.1103/5ngk-4v9j to visualize the generated microstructures over the entire composition range of $c_{avg} = 0.27$–0.48, and temporal evolution of the size distribution of the B-rich white phase.

[53] S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*, 1st ed., Interdisciplinary Applied Mathematics (Springer, New York, NY, 2002), Vol. 16, pp. XXI, 703.

[54] https://github.com/Sachin-G13/Deep-Learning-Driven-Prediction-of-Microstructure-Evolution-via-Latent-Space-Interpolation/tree/main.