

Reduced-order prediction model for the Cahn–Hilliard equation based on deep learning

Zhixian Lv ^a, Xin Song ^a, Jiachen Feng ^a, Qing Xia ^a, Binhu Xia ^b, Yibao Li ^a, ^{*}

^a School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, Shaanxi, China

^b School of Computer Science, Xijing University, Xi'an 710049, Shaanxi, China



ARTICLE INFO

Keywords:

Reduced-order prediction model

Cahn–Hilliard equation

Deep learning

Data driven

ABSTRACT

This study presents an end-to-end deep learning framework for nonlinear reduced-order modeling and prediction, combining Variational Autoencoders (VAE) for feature extraction and Long Short-Term Memory (LSTM) networks for temporal prediction. The framework simplifies the modeling process by integrating multiple steps into a unified architecture, improving both design and training efficiency. The VAE compresses input data into a low-dimensional latent space while using a progressive channel reduction strategy to retain key features and minimize redundancy. The LSTM network captures temporal dependencies, ensuring accurate predictions based on historical data. The framework is validated through applications to the Cahn–Hilliard (CH) equation, demonstrating superior performance over traditional dimensionality reduction and prediction models. A comprehensive hyperparameter analysis identifies optimal configurations, and the model's extrapolation capabilities and computational efficiency are thoroughly assessed. Results highlight the framework's potential as an effective tool for modeling and predicting complex dynamic systems governed by partial differential equations.

1. Introduction

High-fidelity simulations of nonlinear partial differential equations often require substantial computational costs, posing challenges for real-time decision-making applications that demand rapid responses. In response to this challenge, researchers have concentrated on investigating reduced-order modeling for such systems, aiming to address the pressing need for swift predictive simulations while mitigating costs [1,2]. This research has garnered widespread interest, striving to devise more efficient and practical methods to adeptly handle the simulation demands of complex systems, thereby offering viable and effective solutions for practical decision-making [3]. Recently, owing to the rapid improvement in computational power, reduced-order models have made significant strides across various domains. A reduced-order model is a modeling approach utilizing an approximate simplified system to substitute complex dynamic models in numerical simulations, which has proven successful in computational fluid dynamics [4], aeroelasticity [5,6], and environmental science [7,8]. Wang et al. introduced a non-intrusive deep learning-based reduced order modeling approach for stochastic underground flow in highly heterogeneous media, enabling rapid simulations decoupled from input parameters [9].

This reduced-order modeling introduces a fresh perspective for comprehending and applying the phase-field equation and its dynamic evolution.

The phase-field equation holds a pivotal role across diverse scientific domains, with successful applications spanning physics [10], materials science [11,12], biology [13,14], topology optimization [15,16], data assimilation [17,18], and chemistry [19–21], providing a significant framework for comprehending the spatiotemporal evolution of matter. Its applications in additive manufacturing [22,23], thermo-mechanical fracture [24], crystal growth [25–27], particle interactions [28,29], and porous media [30] within complex systems have profound implications for advancing basic science and engineering applications. Despite these achievements, modeling phase-field equations encounters considerable challenges, particularly in high-dimensional and rapidly evolving scenarios. Traditional modeling methods seem inadequate in addressing such complexities. Phase-field models, employed to depict spatiotemporal evolution in physical systems, often involve high-dimensional and intricate data. Microscopic structures in these models are described by continuous field variables evolving in space and time, necessitating precise spatiotemporal discretization and an accurate representation of free energy. These factors contribute to the computational expense

* Corresponding author.

E-mail address: yibaoli@xjtu.edu.cn (Y. Li).

URL: <http://gr.xjtu.edu.cn/web/yibaoli> (Y. Li).

associated with these models.

To enhance the efficiency of phase-field model simulations and reduce computational costs, researchers have primarily concentrated on leveraging high-performance computing architectures [31–33] and implementing advanced numerical schemes [34–36]. Liao et al. proposed an adaptive time-stepping strategy for efficiently capturing multi-scale behaviors and accelerating numerical simulations [37,38]. Despite these endeavors, the computational cost for simulating phase-field equations remains high, particularly when real-time predictions are necessary, resulting in slow computation speeds. Due to the prohibitive cost of predicting the entire phase-field model, researchers have dedicated substantial efforts to developing data-driven [39,40] and machine learning (ML) surrogate models [41–43]. Kiyani et al. introduced data-driven architectures utilizing a multilayer perceptron, a convolutional neural network, and a combination of a CNN and LSTM structures to discover the nonlinear equations of motion for phase-field models with nonconserved and conserved order parameters [44]. Hernandez et al. developed a technique to learn the latent dimensionality of a physical system from data and obtain a thermodynamics-aware time integrator, ensuring compliance with the laws of thermodynamics [45]. Wang et al. explored 3D feature extraction using 3D MF-DFA [46], and also developed a hybrid MF-DFA-phase-field MRI classification system [47].

In recent years, researchers have actively explored efficient and reliable ROM techniques for simulating nonlinear systems with multiscale complexity in both spatial and temporal domains. ROM consists primarily of two key components: the process of transforming a high-dimensional space into a low-dimensional latent space and modeling the temporal evolution within that latent space. Common dimensionality reduction methods include Proper Orthogonal Decomposition(POD) [48], Principal Component Analysis(PCA), and Singular Value Decomposition(SVD)-based techniques [49], all of which are used for feature extraction in ROMs. Fu et al. proposed a physics-informed machine learning method suitable for non-intrusive parametric ROM in low-data environments [50]. In addition to linear dimensionality reduction methods, various machine-learning-based nonlinear approaches have also been explored, as machine learning techniques show strong capabilities in handling nonlinear flow phenomena [51]. Cheng et al. developed a system integrating reduced-order surrogate models with an innovative data assimilation technique, significantly enhancing the accuracy and speed of deep learning surrogate models compared to computational fluid dynamics simulations by incorporating real-time observations from different physical spaces [52]. Maulik et al. proposed a ROM for advection-dominated systems using convolutional autoencoders and recurrent neural networks, extending it into a parametric ROM by directly embedding parametric information into the latent space, thus enhancing the model's ability to capture system evolution trends [53]. However, these models' performance may be substantially influenced by input data characteristics and distribution, which can affect their generalization capability. Variational autoencoders as probabilistic generative models, learn the latent distribution of data, thereby modeling the data generation process. This approach allows VAEs to flexibly generate new samples and perform sample reconstructions, with enhanced expressive capabilities for handling complex, high-dimensional data. The latent space generated by VAEs captures the diversity and complexity of the data [54].

This paper introduces an end-to-end deep learning framework for reduced-order modeling that seamlessly integrates the feature extraction capabilities of VAE with the prediction strengths of LSTM. By consolidating complex multi-step processes, this unified framework minimizes the intricacies associated with data transfer and interface management between independent modules, thereby streamlining both design and training. The proposed model employs VAE feature extraction to effectively map input data into a low-dimensional latent space. Initially, it utilizes fewer channels to represent low-level features, progressively increasing the number of channels to enhance feature extraction across multiple levels. In the deeper layers, a reduction in

channel numbers directs focus towards more representative high-level features. During the prediction phase, LSTM captures the temporal dependencies inherent in the input data, elucidating how current outputs are influenced by preceding time steps. Moreover, the reduced-order model framework showcases remarkable scalability and adaptability, allowing for flexible adjustments to inputs and outputs to accommodate a variety of problem scenarios. This study first validates the application of the proposed reduced-order prediction model using the CH equation and conducts a comprehensive comparative analysis of hyperparameters to ascertain the optimal model configuration. Subsequently, we compare both the dimensionality reduction and prediction modules with alternative models, highlighting the superiority of our approach. Finally, we evaluate the model's extrapolation capabilities and analyze its computational efficiency, thereby providing a thorough assessment of its performance.

The remainder of this paper is organized as follows: Section 2 details the generation of data for the CH equation and introduces the reduced-order model and its configurations; Section 3 validates the feasibility of the reduced-order predictive model, compares the advantages of VAE and LSTM, analyzes parameter settings, and assesses model efficiency. Finally, Section 4 summarizes the main contributions of this research.

2. Data preparation and model construction

2.1. Solving CH equation to prepare data

Phase field models employ a set of field variables to characterize microscopic structures. It is assumed that these field variables exhibit continuity in the interface region, in stark contrast to their discontinuity in the sharp interface model. The temporal evolution of conserved field variables is dictated by the CH equation:

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = \Delta\mu(\mathbf{x}, t), \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^2, \quad 0 < t, \quad (1a)$$

$$\mu(\mathbf{x}, t) = -\epsilon^2 \Delta u(\mathbf{x}, t) + F'(u(\mathbf{x}, t)). \quad (1b)$$

The boundary conditions are periodic for $u(\mathbf{x}, t)$ and $\mu(\mathbf{x}, t)$. The above free energy density in the CH equation can be represented as:

$$F(u(\mathbf{x}, t)) = \int_{\Omega} [F(u(\mathbf{x}, t)) + \frac{\epsilon^2}{2} |\nabla u(\mathbf{x}, t)|^2] d\mathbf{x}, \quad (2)$$

where $F(u) = \frac{1}{4}(u^2 - 1)^2$, and ϵ is the gradient energy coefficient. This expression captures the energy contribution associated with the phase field u , including the chemical free energy density $F(u)$ and a gradient term $|\nabla u|^2$ that penalizes sharp variations. Our computational domain consists of $N_x \times N_y$ pixels on the 2D space $\Omega = (0, L_x) \times (0, L_y)$. Unless otherwise specified, we set $L_x = L_y$ and $N_x = N_y$. Unless otherwise specified, ϵ is defined as $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$ [55], where $h = L_x/N_x$ is the spatial step size. Furthermore, let $u_{p,q}^n$ be an approximation of $u(p, q, n\Delta t)$, where Δt is the time step. For solving Eq. (1), we employ a Fast Fourier Transform (FFT) method with a computational complexity of $O(N \log N)$ [56]. The fast fourier transform $\hat{u}_{a,b}^n$ for $a = 1 - N_x/2, \dots, N_x/2$, and $b = 1 - N_y/2, \dots, N_y/2$ is defined as:

$$\hat{u}_{a,b}^n = \sum_{p=1}^{N_x} \sum_{q=1}^{N_y} u_{p,q}^n e^{-2\pi i (\xi_a x_p + \eta_b y_q)}, \quad (3)$$

The variables x_p , y_q , ξ_a and η_b are defined as $x_p = pL_x/N_x$, $y_q = qL_y/N_y$, $\xi_a = a/L_x$ and $\eta_b = b/L_y$ respectively. The inverse fast fourier transform is:

$$u_{p,q}^n = \frac{1}{N_x N_y} \sum_{p=1-N_x/2}^{N_x/2} \sum_{q=1-N_y/2}^{N_y/2} \hat{u}_{a,b}^n e^{2\pi i (\xi_a x_p + \eta_b y_q)}. \quad (4)$$

We have employed the fast fourier transform for the Laplacian operator, which is defined as

$$\Delta_d \hat{u}_{a,b}^n = -((2\pi\xi_a)^2 + (2\pi\eta_b)^2) \hat{u}_{a,b}^n. \quad (5)$$

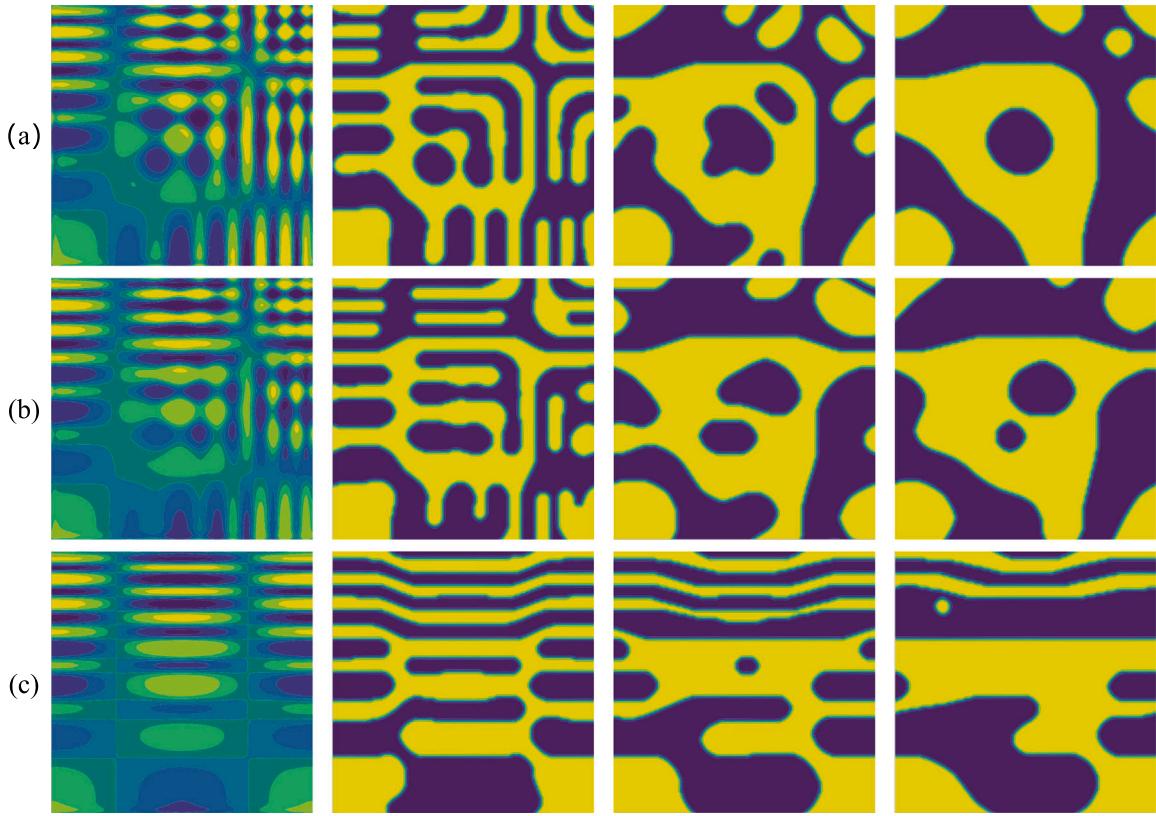


Fig. 1. Panels (a), (b), and (c) depict the phase-field evolution results of the high-fidelity model for three separate cases at four discrete time frames ($n = 1, 10, 50$, and 100), respectively.

$$\hat{u}_{a,b}^{n+1} = \frac{(2 - \epsilon^2((2\pi\xi_a)^2 + (2\pi\eta_b)^2)^2 \Delta t - \lambda((2\pi\xi_a)^2 + (2\pi\eta_b)^2)\Delta t)\hat{u}_{a,b}^n - 2((2\pi\xi_a)^2 + (2\pi\eta_b)^2)\hat{h}_{a,b}^{n+\frac{1}{2}}\Delta t}{2 + \epsilon^2((2\pi\xi_a)^2 + (2\pi\eta_b)^2)^2 \Delta t + \lambda((2\pi\xi_a)^2 + (2\pi\eta_b)^2)\Delta t}. \quad (8)$$

Box I.

In order to achieve second-order accuracy in both temporal and spatial domains, we employ Crank–Nicolson method to formulate the discretization of Eq. (1) as follows:

$$\frac{u^{n+1} - u^n}{\Delta t} = \Delta_d h^{n+\frac{1}{2}}, \quad (6a)$$

$$\mu^{n+\frac{1}{2}} = \frac{-\epsilon^2(\Delta_d u^{n+1} + \Delta_d u^n) + 3(F'(u^n) - \lambda u^n) - (F'(u^{n-1}) - \lambda u^{n-1}) + \lambda(u^{n+1} + u^n)}{2}. \quad (6b)$$

Thus, Eq. (6) can be transformed into the Fourier space as follows:

$$\frac{\hat{u}_{a,b}^{n+1} - \hat{u}_{a,b}^n}{\Delta t} = -((2\pi\xi_a)^2 + (2\pi\eta_b)^2)\hat{\mu}_{a,b}^{n+\frac{1}{2}}, \quad (7a)$$

$$\begin{aligned} \hat{\mu}_{a,b}^{n+\frac{1}{2}} &= \frac{\epsilon^2}{2}(((2\pi\xi_a)^2 + (2\pi\eta_b)^2)\hat{u}_{a,b}^{n+1} + ((2\pi\xi_a)^2 + (2\pi\eta_b)^2)\hat{u}_{a,b}^n) \\ &\quad + \hat{h}_{a,b}^{n+\frac{1}{2}} + \frac{\lambda}{2}(\hat{u}_{a,b}^{n+1} + \hat{u}_{a,b}^n). \end{aligned} \quad (7b)$$

Here, $\hat{h}^{n+\frac{1}{2}} = 3/2(\hat{F}'(u^n) - \lambda\hat{u}^n) - 1/2(\hat{F}'(u^{n-1}) - \lambda\hat{u}^{n-1})$. Therefore, we obtain the following fast Fourier transform (see Box I):

The corresponding function $u_{p,q}^{n+1}$ can be computed using Eq. (4). The initial condition is given as:

$$u_{p,q}^0 = \frac{1}{4}(\rho \cos(2\pi p) \cos(2\pi q) + \tau \cos((2\pi p)^2) \cos(2\pi q) + \cos(2\pi p) \cos((2\pi q)^2)). \quad (9)$$

In this study, we generate 100 sets of initial conditions for phase-field simulations, with ρ and τ being random variables each ranging from 0

to 1. Each simulation evolves for 10,000 time steps with a time step size of $\Delta t = 1 \times 10^{-4}$. The structural state is recorded every 100 time steps, yielding 100 time frames per simulation. These simulations are performed on a domain $\Omega = [0, 1] \times [0, 1]$, discretized with a mesh size of 128×128 . As a result, we obtain phase-field data for 100 cases across 100 time frames. Fig. 1 illustrates the evolution of the phase-field for three representative cases at four time frames ($n = 1, 10, 50, 100$). For model training, the n th and $(n+1)$ th frames are used as input and label pairs. The 100 phase-field cases are randomly split into 80% training data and 20% testing data. Consequently, we have training data for 80 cases, each with a dimension of 128×128 across 100 frames, and testing data for 20 cases with the same dimensions.

Remark 1. Extensive research has been conducted on the energy stability [57,58] and convergence [59] of the Cahn–Hilliard equation. In this study, we employ the FFT method to solve the equation. Numerous studies demonstrate that the FFT method not only provides significant computational efficiency but also effectively satisfies the energy stability and convergence requirements of the equation [60,61]. Following the approach outlined in [62,63], unconditional energy stability and convergence analysis of this numerical scheme can be directly established. Therefore, a detailed discussion of the energy stability and convergence of the FFT method is omitted in this paper.

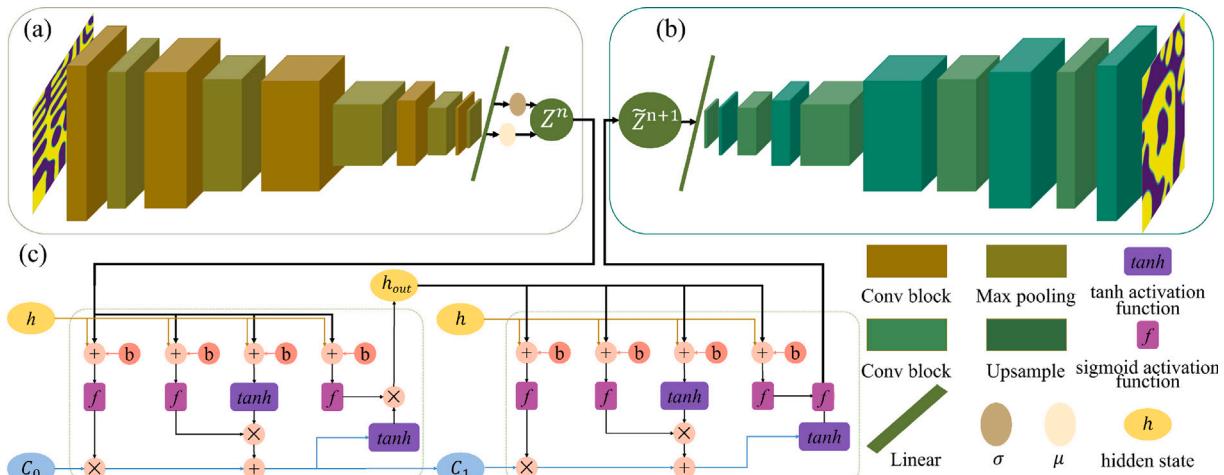


Fig. 2. The structure of the VAE-LSTM Reduce order model.

2.2. Combining VAE and LSTM to construct ROM

This section outlines the details of the proposed method. We devise a end-to-end reduced-order prediction model by amalgamating the variational autoencoder with Long LSTM. The structural schematic of the model is illustrated in Fig. 2, delineating the encoder, decoder, and LSTM prediction module. For the encoder module, as illustrated in Fig. 2(a), feature extraction is performed employing convolutional modules. Each convolutional block comprises two convolutional layers, followed by a batch normalization layer and Rectified Linear Unit (ReLU) activation function. The convolutional kernel size is 3×3 , with a stride of 1 and padding of 1. The specific computational process is detailed as follows:

$$\begin{aligned} \text{Conv block}(\text{ch_in}, \text{ch_out})(X) = \\ \text{ReLU}(\text{BN}(\text{Conv2d}(\text{ReLU}(\text{BN}(\text{Conv2d}(X, \text{ch_in}, \text{ch_out}))), \text{ch_out}, \text{ch_out}))). \end{aligned} \quad (10)$$

Here, X represents the input data, Conv2d denotes the two-dimensional convolution operation, and ch_in and ch_out denote the input and output channel counts, respectively. Then, max-pooling layers are employed to reduce the spatial dimensions of the feature maps while retaining the most salient features, thereby reducing the parameter count and computational load in subsequent layers. This aids in mitigating overfitting and enhancing the model's generalization capability. We employ a max-pooling layer with a kernel size of 2×2 and a stride of 2. The input u^n represents phase field model data at the n th time frame, with dimensions $(1, 128, 128)$. We utilize Conv block1, Conv block2, and Conv block3 to progressively increase channel numbers, iteratively extracting features. Subsequently, Conv block4 and Conv block5 are employed to decrease the feature channel dimensions, achieving dimensionality reduction. Incrementally increasing channel numbers, allows the model to gradually adapt to the complexity of the input data. Initiating with fewer channels assists the model in focusing on learning simpler features. As the channel numbers increase, the model can better capture more complex patterns and details within the data. When reducing the channel numbers in higher-level feature representations, irrelevant features are filtered out, rendering the model more robust and less susceptible to noise interference. Moreover, fewer channels imply fewer multiplication operations and parameter counts, thereby making the model more lightweight and computationally efficient. Detailed channel variations and data dimension changes are illustrated in Table 1. Subsequently, to enhance the model's ability to learn the data distribution and facilitate gradient propagation, we apply reparameterization to the mean and log variance of the encoder output. The

features obtained from the Flatten layer are input into the Reparameterize layer, enabling us to sample from a Gaussian distribution and generate samples in the latent space rather than directly sampling from a fixed distribution. The reparameterized latent variable Z^n is defined as follows:

$$Z^n = \mu^n + \text{eps} \cdot \exp\left(\frac{1}{2} \cdot \sigma^n\right), \quad (11)$$

where μ^n represents the mean, σ^n is the standard deviation, and $\text{eps} \sim \mathcal{N}(0, 1)$. After passing through the encoder module, the input data u^n undergoes dimensionality reduction, resulting in a latent space representation Z^n with a dimensionality of $(1, L)$. In our study, L is set to 2. Detailed experimental results are discussed in Section 3.2.

The LSTM module, depicted in Fig. 2(c), comprises two layers with hidden states configured to 32 units. The output from the hidden states of the first layer serves as the input for the second layer, enabling the generation of predictions for the latent space. With Z^n as input to the LSTM module, the model extrapolates the latent space value for the $(n+1)$ th time frame based on the n th time frame, resulting in \tilde{Z}^{n+1} . In contrast to feedforward networks, LSTM architectures provide a more efficient parameterization. Their design promotes parameter reuse across different temporal instances, which can help reduce overfitting and enhance generalization capabilities. When necessary, this model should be applied flexibly, requiring minimal adjustments for multi-timepoint data prediction tasks. While our proposed reduced-order model processes the phase field data for individual time frames u^n , the LSTM module inherently captures the temporal dependencies within each frame. Despite the seemingly isolated treatment of phase field data, LSTM enables the implicit integration and utilization of underlying temporal correlations present in the dataset.

The decoder section reconstructs the latent space into the original data size using \tilde{Z}^{n+1} as input, as illustrated in Fig. 2(b). Starting with an input data size of $(1, L)$, linear layers and size adjustment operations produce an intermediate output of size $(8, 4, 4)$. The model further refines the features through UpConv and Convolutional blocks, ultimately generating an output of size $(1, 128, 128)$. The operations executed by the convolutional blocks are detailed in Eq. (10). The UpConv module is designed to upsample the input feature maps. Within this module, the convolution operation preserves the number of channels, followed by batch normalization and ReLU activation, which together enhance the resolution and enrich feature representation. The formulation for this operation is as follows:

$$\text{UpConv}(X) = \text{ReLU}(\text{BN}(\text{Conv2d}(\text{Upsample}(X)))), \quad (12)$$

where X represents the input to this module, and Upsample denotes the upsampling operation. The specific parameter configurations of the

Table 1
Structure of the VAE model.

Encoder		Decoder	
Layer	Data size	Layer	Data size
Input	(1,128,128)	Input	(1,L)
Conv block1 (1, 32)	(32,128,128)	Linear (L,128)	(1,128)
Max pooling1 (2, 2)	(32,64,64)	Resize	(8,4,4)
Conv block2 (32, 64)	(64,64,64)	UpConv1 (2, 2)	(8,8,8)
Max pooling2 (2, 2)	(64,32,32)	Conv block1 (8, 32)	(32,8,8)
Conv block3 (64, 128)	(128,32,32)	UpConv2 (2, 2)	(32,16,16)
Max pooling3 (2, 2)	(128,16,16)	Conv block2 (32, 128)	(128,16,16)
Conv block4 (128, 32)	(32,16,16)	UpConv3 (2, 2)	(128,32,32)
Max pooling4 (2, 2)	(32,8,8)	Conv block3 (128, 64)	(64,32,32)
Conv block5 (32, 8)	(8,8,8)	UpConv4 (2, 2)	(64,64,64)
Max pooling5 (2, 2)	(8,4,4)	Conv block4 (64, 32)	(32,64,64)
Flatten	(1,128)	UpConv5 (2, 2)	(32,128,128)
Reparameterize (128,L)	(1,L)	Conv block4(32, 1)	(1,128,128)

decoder are detailed in Table 1.

2.3. Evaluation metrics and model configuration

In this research, the Mean Squared Error (MSE) is employed to quantify the average square difference between the model's output and the actual labels. Notably, MSE exhibits robustness to outliers during the training process, as the squared errors of outliers contribute disproportionately to the overall loss, facilitating a faster convergence of the model to optimal parameters. The computational formula is expressed as follows:

$$MSE(u^m, \tilde{u}^m) = \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} (u_{i,j}^m - \tilde{u}_{i,j}^m)^2, \quad (13)$$

where $u_{i,j}^m$ is the true label for the m th sample, and $\tilde{u}_{i,j}^m$ is the predicted value for the m th sample. In contrast to the MSE scalar error metric, which fails to provide insights into the accuracy of our surrogate model across various prediction intervals, the autocorrelation function serves as a more informative indicator [64]. To achieve this, a component indicator function $f(u_{i,j}^n)$ is utilized to ascertain the dominant phase at position (i,j) for a given microstructure. This function is spatially mapped at each time frame to partition the domain based on the distribution of the dominant phase:

$$f(u_{i,j}^n) = \begin{cases} 1, & u_{i,j}^n > 0, \\ 0, & u_{i,j}^n \leq 0. \end{cases} \quad (14)$$

The autocorrelation function of the phase field variable $f(u_{i,j}^n)$ is employed to statistically characterize the microstructure's evolution :

$$A(u_{i,j}^n) = f(u_{i,j}^n) \otimes f(u_{i,j}^n) = \frac{\mathfrak{F}^{-1}(|\mathfrak{F}(f(u_{i,j}^n))|^2)}{N^2}, \quad (15)$$

where \mathfrak{F} is the Fourier transform and \mathfrak{F}^{-1} is the inverse Fourier transform. It denotes the conditional probability that two points at positions (i,j) have a phase field variable $u = 1$. Both the autocorrelation and its radial average Eq. (16) encapsulate information akin to high-fidelity simulations, providing a quantitative measure for the representative evolution of the microstructure.

$$\bar{A}(r, n) = \frac{\sum_k^{N_r} A(u_k^n)}{N_r}, \quad (16)$$

where N_r denotes the number of points at a distance r from the center.

We employ the Adam optimizer, a gradient-based optimization algorithm. Adam facilitates swift convergence to local optima while avoiding entrapment in saddle points. The learning rate parameter, initialized at 0.001, defining the step size for updating parameters in each iteration. Additionally, the study employs the cosine annealing warm restarts learning rate scheduler. This scheduler dynamically adjusts the learning rate following a cosine function, aiming for improved training outcomes. Specifically, it decreases the learning rate to a smaller value

at the end of each cycle, restarts training in the next cycle, and gradually increases the learning rate. This approach facilitates achieving comparable or even superior results with fewer training epochs [65].

3. Result and comparison

This section first validates the application of the proposed reduced-order prediction model in the CH equation and conducts a detailed comparative analysis of the hyperparameters to identify the optimal model configuration. Then, we compare both the dimensionality reduction and prediction modules with other models, demonstrating the superiority of our model. Finally, we evaluate the extrapolation capability of the model and analyze its computational efficiency. In this study, we focused on the 2D case as it effectively demonstrates the core principles of our deep learning framework.

3.1. Prediction evolution of CH equation

This subsection presents the predictive evolution of the CH equation and introduces a novel metric to validate the efficacy of our model. The results of a randomly selected case from the test set at three distinct time frames ($n=10$, $n=50$, and $n=100$) are illustrated in Figs. 3. Specifically, Fig. 3(a) shows the ground truth evolution of the CH equation, Fig. 3(b) displays the predictions generated by the reduced-order prediction model, and Fig. 3(c) presents the absolute values of the point-wise errors between the ground truth and the predictions. By comparing the point-wise errors between the predicted and ground truth values, we can clearly identify the distribution of model errors. The majority of point-wise errors are less than 0.1, with regions exceeding 0.4 primarily located at the phase interfaces, attributed to significant numerical discrepancies at the phase boundaries. Moreover, the reduced-order prediction model achieves a Mean Squared Error of just 0.022. Overall, the predictive evolution of our model for the CH equation demonstrates its effectiveness through the comparison of point-wise errors between the true data and the predictions.

The autocorrelation representation of the ROM predictions and the corresponding error analysis, as shown in Fig. 4, spans three different time points ($n=10$, $n=50$, and $n=100$), corresponding to the analyses in Fig. 3. The autocorrelation profiles derived from the ROM predictions are showcased in Fig. 4(a). The point-wise absolute errors between the autocorrelation profiles of the observed data and the model predictions are delineated in Fig. 4(b). Notably, the errors initially exhibit considerable dispersion during the early stages of case evolution, gradually converging as the process advances, leading to a more concentrated distribution of errors. In Fig. 4(c), the solid and dashed curves denote the radial averages of the autocorrelation for the observed data and ROM predictions, respectively, exhibiting almost complete alignment. While minor discrepancies are present in the autocorrelation representation, they are notably small (maximum error is only 8×10^{-3}),

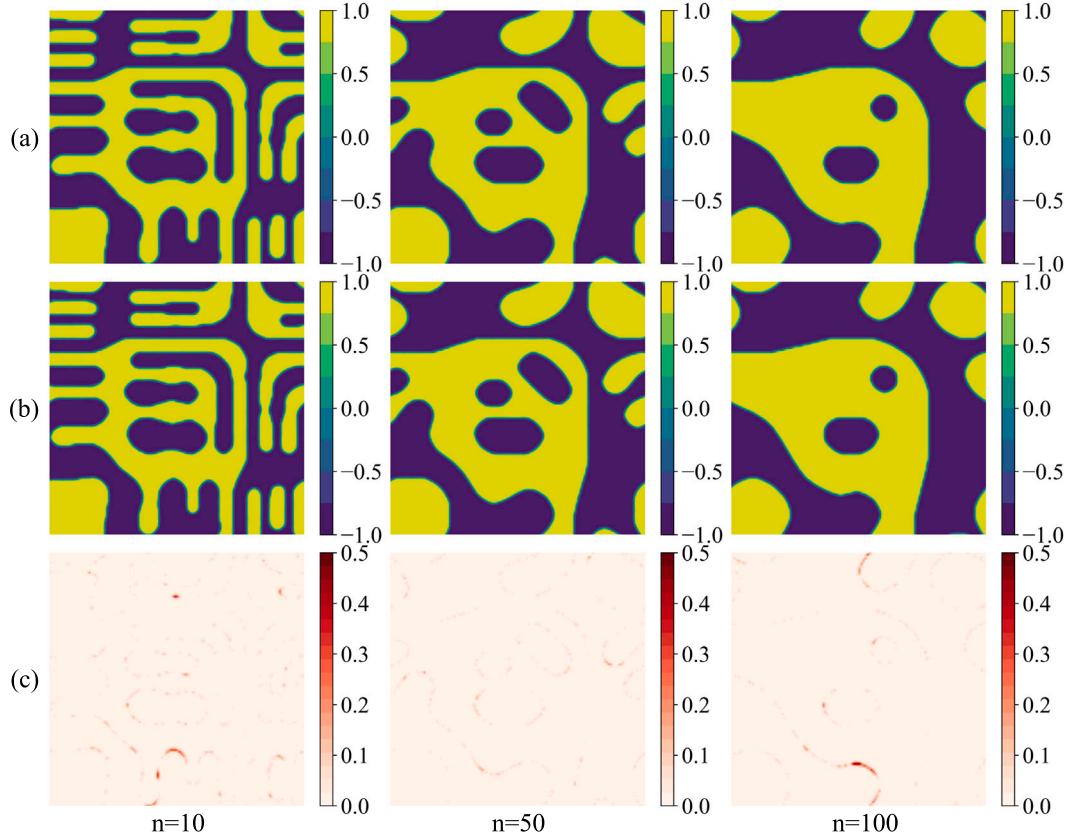


Fig. 3. The test set outcomes at three time frames ($n=10$, $n=50$, and $n=100$) are shown: (a) depicts the actual evolution of the CH equation, (b) shows predictions by reduced-order prediction model, and (c) displays the absolute value of point-wise errors between the ground truth and the predictions.

affirming the close correspondence between the predictions of our ROM model and the ground truth. This indicates that the ROM model effectively captures the dynamics of the CH equation, maintaining a reliable performance throughout the evolution process.

In this study, we adopt a polynomial approximation of the double-well potential. By substituting the free energy functional $F(u) = (1+u)\ln(1+u) + (1-u)\ln(1-u) - 0.5\theta_0 u^2$ into Eq. (2), we derive the CH equation with a Flory–Huggins logarithmic energy potential [66], where θ_0 is a physical parameter. This equation has been widely studied and is well-known to satisfy the positivity-preserving property [67,68]. Specifically, when solved using FFT, the numerical solution avoids reaching the singular values ± 1 , ensuring that $0 < (1-u)$ and $0 < (1+u)$ at all times [69,70]. Although the Flory–Huggins CH equation introduces more complex logarithmic terms, it shares significant similarities with the polynomial approximation used in this study, particularly in describing phase separation and interface evolution. Both models rely on the gradient flow framework, where the system evolves by minimizing the energy functional. Through fine-tuning and transfer learning, our reduced-order model can efficiently adapt to the CH equation with a Flory–Huggins logarithmic potential, maintaining high accuracy on new tasks. Fig. 5 illustrates a comparison between the high-fidelity simulation and the reduced-order model predictions for the CH equation with a Flory–Huggins logarithmic potential at $n = 100$. The point-wise absolute error distribution between the two indicates that the reduced-order model effectively captures the main features of phase evolution.

3.2. Optimal model parameters

In this subsection, we conduct a comprehensive analysis of the proposed reduced-order predictive model, focusing primarily on the

performance of the VAE framework across different latent space dimensions. We also explore the impact of hyperparameter configurations in the LSTM prediction module on model accuracy.

Firstly, regarding the VAE framework analysis, we examine the effect of latent space dimensions on the test set loss function during the training of the phase field model. Fig. 6 illustrates the variations in test set loss values throughout the training period, with the horizontal axis representing training epochs and the vertical axis indicating the loss function values. Each curve corresponds to a different latent space dimension L . The results indicate that the VAE demonstrates significant dimensionality reduction, particularly during the initial 50 epochs of training, where the loss values decrease markedly, stabilizing after approximately 150 epochs. Further subfigures reveal that the model with a dimension of 2 achieves the lowest loss function value throughout the training process. The mean squared error (MSE) results for latent space dimensions $L = 2, 4, 8, 16$, and 32 are $0.022, 0.052, 0.042, 0.043$, and 0.045 , respectively, suggesting that $L = 2$ is a reasonable choice for the CH equation data.

To further elucidate the influence of latent space dimensions, we visualize the evolution results of the first two latent dimensions. Fig. 7 presents the dimensionality reduction results for different latent space dimensions ($L = 2, 4, 8, 16, 32$). The trajectory for $L = 2$ exhibits a highly ordered and regular structure, with smooth, well-separated lines, indicating a more linear and stable evolution of the latent space that effectively captures the main features of the model. As the dimension increases, the trajectories become more complex and intertwined, with significant overlaps and distortions among the lines, suggesting that the evolution of the latent space loses its order at higher dimensions. This may result in the model losing some linear characteristics and potentially overfitting to the complexity of the data, introducing unnecessary noise and complexity.

Next, we provide a detailed comparison of the LSTM prediction module's hyperparameter configurations, with particular attention to

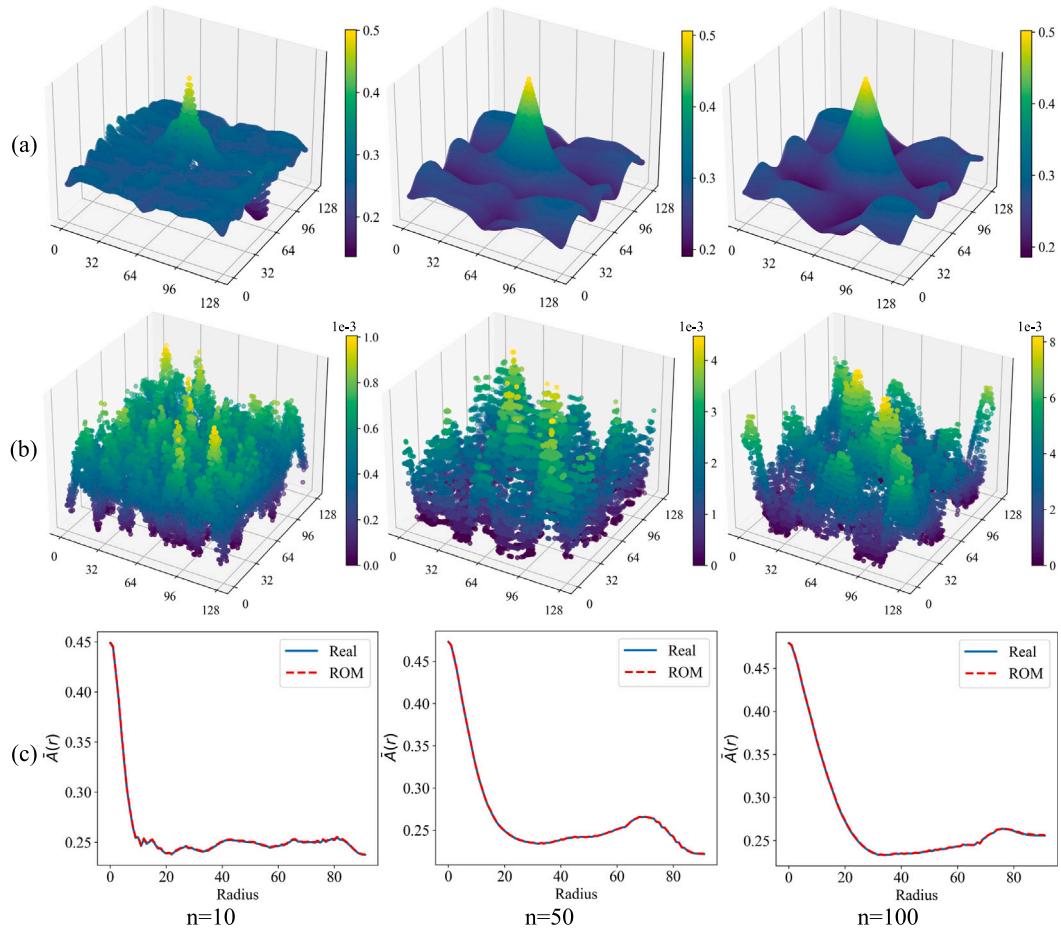


Fig. 4. The autocorrelation representation of the ROM predictions and the corresponding error analysis. (a) Illustrates the autocorrelation profiles derived from the ROM prediction results. (b) Depicts the point-wise absolute errors between the autocorrelation profiles of the ground truth and the predictions. (c) Shows the radial averages of the autocorrelation for both the ground truth (solid line) and the ROM predictions (dashed line).

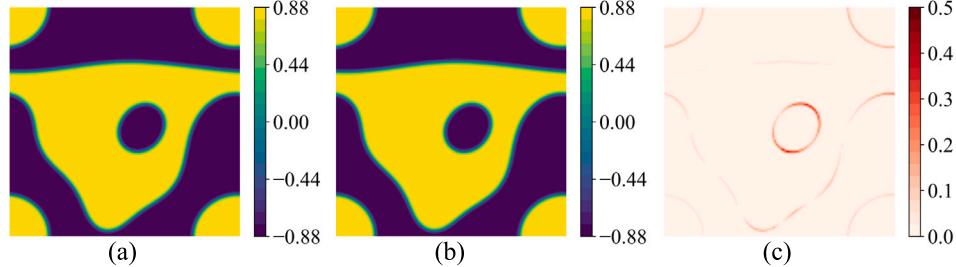


Fig. 5. Comparison of the predicted results from the reduced-order model and high-fidelity simulations for the CH equation with the Flory–Huggins logarithmic potential at $n = 100$: (a) High-fidelity phase field simulation results; (b) Reduced-order model predictions; (c) Absolute point-wise errors between the predicted and high-fidelity results.

Table 2

The hyperparameters and corresponding mean squared error for various LSTM models used in the comparison. The parameters include hidden size, number of layers, and MSE values for each configuration.

	a	b	c	d	e	f(ours)
Hidden size	16	128	256	32	32	32
Num layers	2	2	2	4	8	2
MSE	0.028	0.031	0.047	0.039	0.050	0.022

the effects of hidden layer size and depth on model performance. We compare six different hyperparameter configurations (Table 2), during which the parameters of the VAE framework remain consistent, using

the same learning rate and learning rate scheduler for training. Our model, with a hidden layer size of 32 and a depth of 2 layers, demonstrates the lowest MSE (0.022), outperforming other configurations. This indicates superior feature extraction and generalization capabilities. Other models with larger hidden layers and greater depth suffered from overfitting, leading to higher MSE and greater error. Through the comprehensive comparison of the VAE latent space dimensions and LSTM hyperparameters, we conclude that for reduced-order prediction of the CH equation, setting the latent space dimension to 2 and selecting LSTM parameters of a hidden layer size of 32 and a depth of 2 layers represents the most suitable configuration. This combination effectively enhances predictive accuracy while maintaining model simplicity.

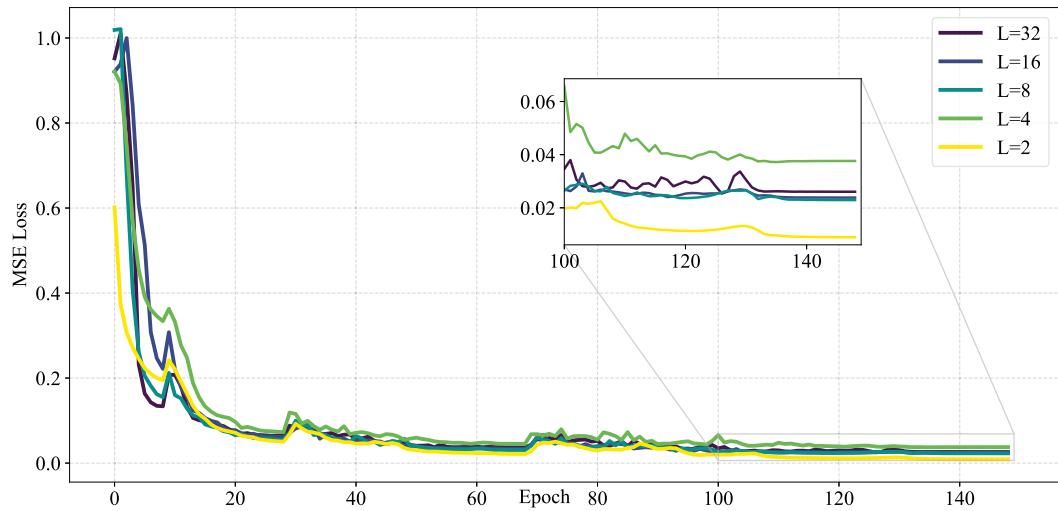


Fig. 6. Variation of the test set loss function during training across different latent space dimensions. Each curve represents a distinct latent space dimension ($L = 2, 4, 8, 16$, and 32). The subgraph emphasizes the trends in loss function changes between epochs 100 and 150 .

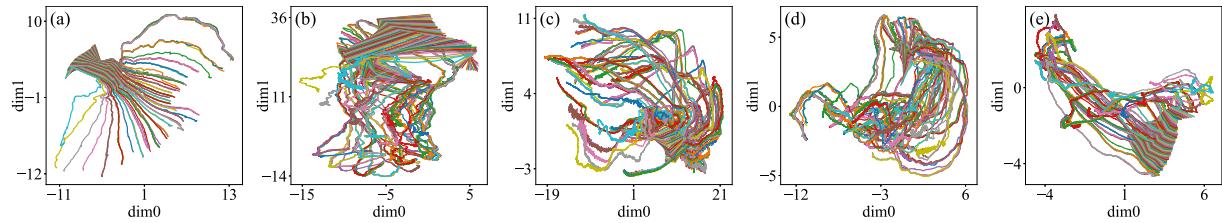


Fig. 7. Visualization of latent space evolution for the first two dimensions (dim0 and dim1) across different latent space dimensions: (a) $L=2$, (b) $L=4$, (c) $L=8$, (d) $L=16$, and (e) $L=32$.

3.3. Comparative analysis of dimensionality reduction and prediction models

This subsection primarily compares the performance of different dimensionality reduction and prediction models in the evolution of phase field equations. By analyzing the latent space characteristics of VAE, PCA, and CAE, we highlight the superiority of VAE in feature extraction and data representation. Simultaneously, we evaluate the relative errors of LSTM, GRU, and RNN in their prediction results, further demonstrating the advantages of LSTM in prediction accuracy. When comparing different dimensionality reduction methods (such as PCA, CAE, and VAE), we used a prediction module with the same configuration. Fig. 8 illustrates the comparison of the first two dimensions of the latent space for different dimensionality reduction techniques applied to the CH equation. The key advantage of VAE in the latent space is its smooth trajectories, enabling it to learn continuous latent space representations and better model complex data distributions. Compared to traditional PCA methods, VAE is more effective in handling nonlinear relationships, avoiding the limitations of linear transformations. In Fig. 8(a), the data variation in PCA is constrained, while VAE's nonlinear transformations preserve the complex nonlinear structure, resulting in more diverse and information-rich trajectories. Unlike CAE, which learns deterministic mappings, VAE models the latent space probabilistically. This allows VAE to not only reduce dimensionality but also generate new samples by sampling from the latent space, which is beneficial for generative tasks or exploring data manifolds. Although CAE can also perform nonlinear dimensionality reduction, its lack of a probabilistic framework results in less clarity in the organization of the latent space compared to VAE, making it more susceptible to overfitting.

Additionally, we used the radial average absolute error of the autocorrelation function $Error_{\bar{A}}(r)$ to compare these models. The error

is defined as $Error_{\bar{A}}(r) = |\bar{A}_{true}(r) - \bar{A}_{predict}(r)|$, where $\bar{A}_{true}(r)$ and $\bar{A}_{predict}(r)$ are the radial averages of the true and predicted phase field model autocorrelation functions, respectively. By randomly selecting a test set, we evaluated the prediction results of these three dimensionality reduction models. Figs. 9(a), (b), and (c) show the radial average errors of autocorrelation representations at times $n = 10$, $n = 50$, and $n = 100$. There are significant differences in performance among the dimensionality reduction methods at different time steps, with VAE consistently outperforming the others at all time steps, particularly in the larger radius regions. This indicates that VAE has exceptional capabilities in capturing the autocorrelation of the phase field model. In contrast, while PCA shows reasonable performance in short time steps, it exhibits substantial fluctuations in error during long-term predictions and lacks adaptability. CAE performs stably at moderate time steps but also displays significant error fluctuations in long-term predictions. Overall, VAE's latent space effectively captures the complex patterns of microstructural evolution, and its consistently low error demonstrates stability and robustness in long-term predictions, making it an ideal choice for addressing microstructural evolution problems. In evaluating the predictive performance of different models, we further validated the effectiveness of LSTM, GRU, and RNN in phase field evolution tasks. In this experiment, we utilized the same VAE dimensionality reduction module, only replacing the prediction modules to assess the predictive accuracy of the phase field evolution. Fig. 10 displays the prediction error distributions of different models at time steps $n = 10$ and $n = 50$ for two randomly selected test cases. The red areas in the figure represent the magnitude of the errors, with darker colors indicating larger errors. The error distribution for the LSTM model is clearly shallower at both time steps, indicating that its predictions have minimal errors compared to the true phase field evolution. In contrast, the GRU and RNN models exhibit relatively larger errors, particularly in regions of complex phase field evolution, where their errors are

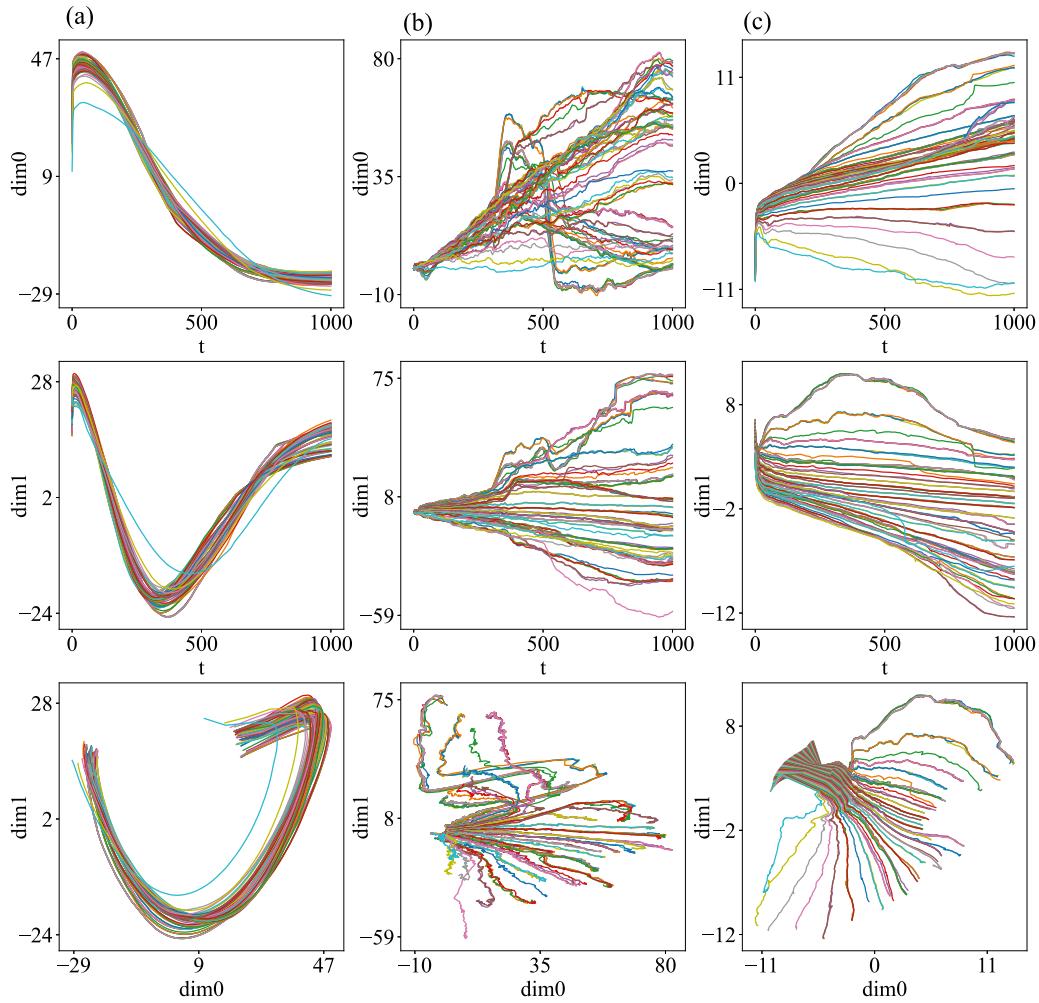


Fig. 8. Different dimensionality reduction methods were applied to study the evolution of microstructures in the low-dimensional representations of phase field simulations across all cases. This analysis includes using (a) the first two principal components obtained via PCA, (b) the two-dimensional reductions from CAE, and (c) the two-dimensional reductions from VAE. The top two rows show the latent spaces derived from each method, while the bottom row illustrates the two-dimensional evolution trajectories of microstructures within the latent spaces.

significantly higher than those of the other two models. These results indicate that LSTM has a stronger capability in capturing the complex dynamics of phase field evolution. Its lower error suggests that LSTM can predict phase field behavior more accurately. Although GRU also possesses some memory capacity, its performance is slightly inferior to that of LSTM, while the traditional RNN model shows noticeable error accumulation due to the vanishing gradient problem. Furthermore, the variation of loss function values for different prediction models (LSTM, GRU, RNN) in the test set is illustrated in Fig. 11. Throughout the training process, the loss function consistently decreases, indicating that the models are continually learning and optimizing. From Fig. 11, it is evident that the LSTM module trains the fastest and achieves the lowest mean squared error (MSE) within the same number of training epochs. On the test set, the MSE loss function values for LSTM, GRU, and RNN are 0.022, 0.031, and 0.049, respectively. Thus, in the predictive tasks of the phase field model, LSTM's performance significantly surpasses that of GRU and RNN. By comparing the dimensionality reduction effects of VAE, PCA, and CAE, as well as the prediction performance of LSTM, GRU, and RNN, we find that VAE demonstrates a clear advantage in capturing complex data distributions and feature extraction. This is particularly evident when dealing with the evolution of phase field equations, where VAE exhibits lower errors and better

stability. Meanwhile, LSTM outperforms GRU and RNN in prediction accuracy, effectively modeling the dynamics of phase field evolution.

3.4. Extrapolation capabilities

We explore the extrapolation capability of the reduced-order model to assess its predictive power beyond the training data range. The initial conditions, as defined in Eq. (9), vary with ρ and τ for the phase field model. The model's training data is generated within the ranges $\rho \in [0, 1]$ and $\tau \in [0, 1]$. To demonstrate the extrapolation capability, we compare the MSE for interpolation and extrapolation segments, as depicted in Fig. 12(a). Interpolation corresponds to $\rho > 0.0$ and $\tau < 1.0$, while extrapolation corresponds to $\rho < 0.0$ and $\tau > 1.0$. The MSE for interpolation is over 10 times smaller than that for extrapolation. Visually, Fig. 13 shows high-fidelity phase field simulations and model predictions for interpolation ($\rho = 0.5$, $\tau = 0.5$) and extrapolation ($\rho = -0.1$, $\tau = 1.1$ and $\rho = -0.2$, $\tau = 1.2$) at the $n = 100$ time frame. The images for $\rho = 0.5$, $\tau = 0.5$ are nearly identical, indicating precise prediction. However, discrepancies are observed for extrapolation cases, though they still yield approximate predictive outcomes. To further evaluate the model's performance on extrapolation tasks, we designed a new set of experiments specifically to test the model's predictive

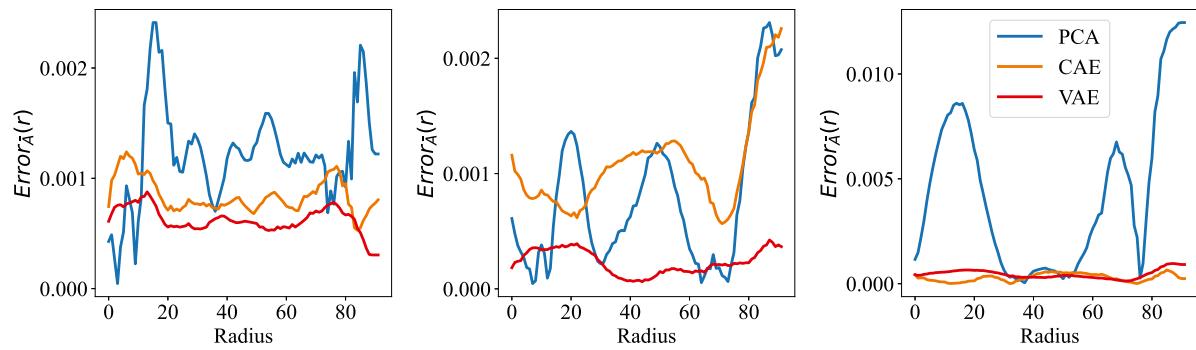


Fig. 9. Comparison of the radial average absolute errors of autocorrelation representations $Error_{\bar{A}}(r)$. Figures (a), (b), and (c) correspond to the radial average errors of autocorrelation representations at time steps $n = 10$, $n = 50$, and $n = 100$, respectively, for the three dimensionality reduction models (PCA, CAE, and VAE), indicated in blue, orange, and red. The error is defined as $Error_{\bar{A}}(r) = |\bar{A}_{true}(r) - \bar{A}_{predicted}(r)|$, where $A_{true}(r)$ and $A_{predicted}(r)$ represent the radial averages of the autocorrelation function of the true and predicted phase field models, respectively.

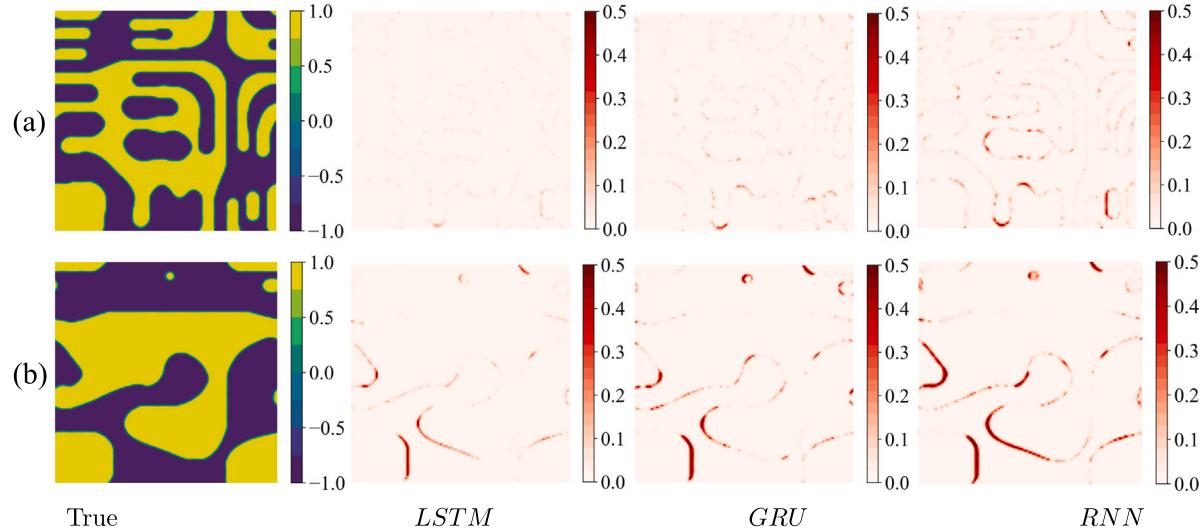


Fig. 10. A comparison of prediction errors between different models (LSTM, GRU, and RNN) for the phase field evolution task. The absolute point-wise errors at time steps $n = 10$ (a) and $n = 50$ (b) are shown, with darker red regions indicating larger errors. The images highlight each model's accuracy in capturing the dynamic evolution of the phase field by comparing their predictions with the ground truth values.

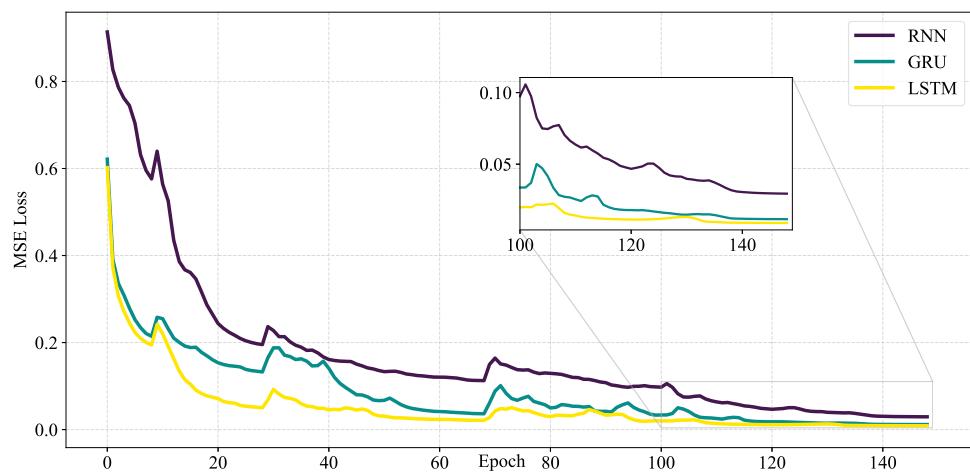


Fig. 11. The loss function values for each epoch of different prediction models (LSTM, RNN, GRU) in the test sets.

capability on unseen data ranges. In this experiment, we generated 80 data samples, all of which have parameter ranges of $\rho < 0.5$ and $\tau < 0.5$, and these samples were used for model training. Subsequently, we tested the model's performance outside the training range by making

predictions in the region where $\rho > 0.5$ and $\tau > 0.5$. The experimental results, shown in Fig. 12(b), illustrate the MSE on the test data. As ρ and τ values increase, the MSE exhibits a clear upward trend, indicating that the model's prediction error increases in the extrapolated range.

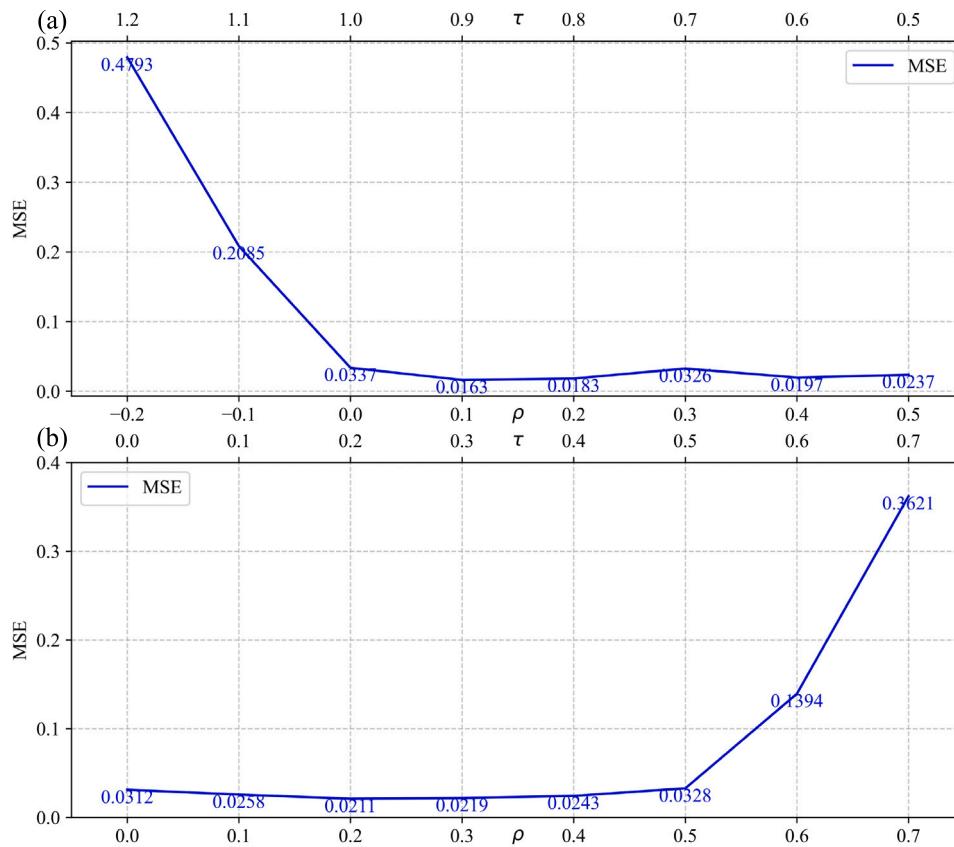


Fig. 12. Comparison of the Mean Squared Error (MSE) between interpolation and extrapolation segments. (a) Interpolation corresponds to $\rho > 0.0$ and $\tau < 1.0$, while extrapolation corresponds to $\rho < 0.0$ and $\tau > 1.0$. (b) Interpolation corresponds to $\rho < 0.5$ and $\tau < 0.5$, while extrapolation corresponds to $\rho > 0.5$ and $\tau > 0.5$.

In addition, we calculated the normalized mass during the evolution of the phase field under both interpolation and extrapolation scenarios. The formula is given as:

$$M_{\text{norm}} = \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |u_{ij}|}{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |u_{ij}^0|},$$

where $|u_{ij}|$ represents the mass at the current position, and $|u_{ij}^0|$ denotes the initial mass distribution. This normalized mass indicates the relative conservation of mass during the evolution. As shown in Fig. 14, the normalized mass over time is plotted for the high-fidelity phase field simulations method (FFT) and the proposed ROM method under three parameter combinations. The results show that the high-fidelity phase field simulation method effectively maintains a normalized mass close to 1 under different parameters, while the ROM method exhibits slight fluctuations in normalized mass, particularly under extrapolated parameters. This indicates that the FFT method demonstrates greater stability in preserving mass conservation. The reason for this difference lies in the nature of the ROM method, which is a data-driven modeling approach designed to extract dominant features or dynamic patterns from high-dimensional full-order models. However, this method may overlook certain critical physical constraints, such as mass conservation. If these constraints are not explicitly incorporated during the construction of the ROM, its predictions may not fully adhere to conservation laws. Nevertheless, the fluctuations in the ROM method's mass remain relatively small, and the overall mass variation is within an acceptable range. It should be remarked that the outcomes demonstrate that our model's predictions are confined to the parameter range employed in constructing the training set, a frequent observation in data-driven machine learning models.

3.5. The computational efficiency of reduced order model

After selecting and calibrating the optimal network structure and latent space dimensionality, we assessed the computational efficiency of our reduced-order models. Computational efficiency comprises two key factors: the one-time cost associated with training the network and the capacity to predict the evolution of microstructures in future field simulations once the model has been trained. In the case of dimensionality reduction using PCA, a fixed one-time cost is incurred for reconstructing the field model. The computational cost for the identical modules in the reduced-order models remains unchanged. For instance, the VAE+LSTM and CAE+LSTM architectures (shown in Fig. 2(b) and (c), respectively) share the same modules, resulting in identical time usage. Table 3 presents a comparison of the computational efficiency and associated training costs for predicting 100 Δt of high-fidelity simulations using various dimensionality reduction and prediction models. The computational gain of our ROM is evaluated by comparing its prediction time ($t_{\text{ROM}} = t_{\text{Prediction}} + t_{\text{Reduction}} + t_{\text{Reconstruction}}$) with the simulation time of the high-fidelity model:

$$\text{Computational gain} = \frac{t_{\text{High-fidelity simulation}}}{t_{\text{Prediction}} + t_{\text{Reduction}} + t_{\text{Reconstruction}}}. \quad (17)$$

Here, $t_{\text{High-fidelity simulation}} = 0.154\text{s}$ is the time required by the FFT method to simulate 100 Δt , while $t_{\text{Prediction}} = 3.74 \times 10^{-5}\text{s}$ is the time taken by the LSTM prediction module in the ROM. The times for $t_{\text{Reduction}}$ and $t_{\text{Reconstruction}}$ are listed in Table 3. From a training time perspective, the VAE+LSTM model converges faster and requires less training time, thereby reducing computational costs. High-fidelity models generally require detailed microstructure modeling in both spatial and temporal domains, with small time steps necessary to accurately capture the system's dynamic evolution. In contrast, the ROM learns to evolve the system over larger time steps. As highlighted

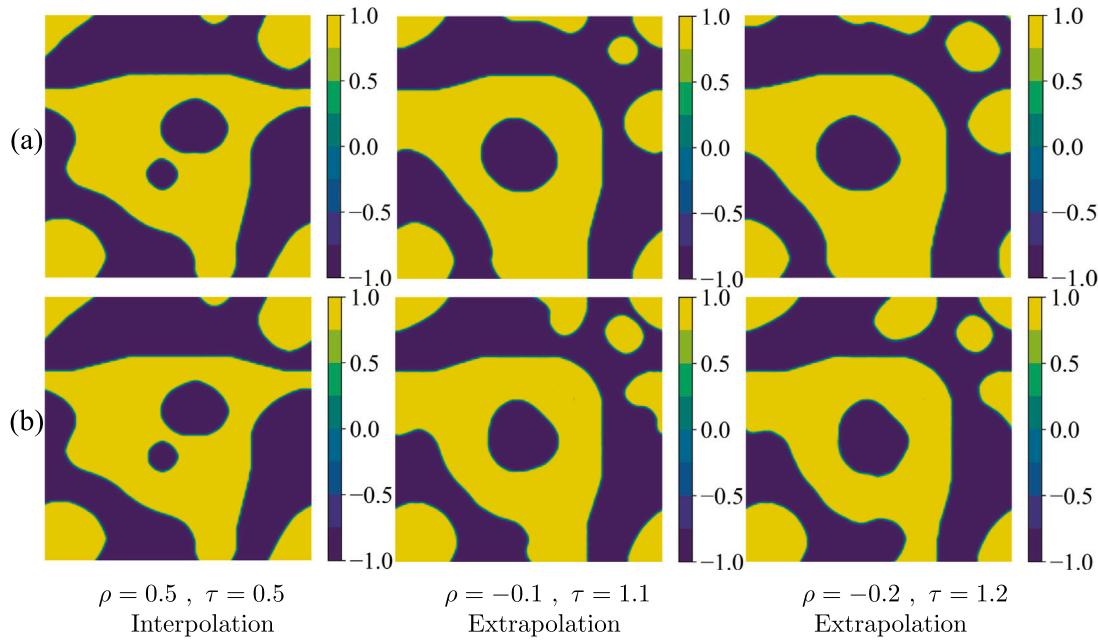


Fig. 13. The results of high-fidelity phase field simulations and model predictions at the $n = 100$ time frame for interpolation ($\rho = 0.5, \tau = 0.5$) and extrapolation ($\rho = -0.1, \tau = 1.1$ and $\rho = -0.2, \tau = 1.2$).

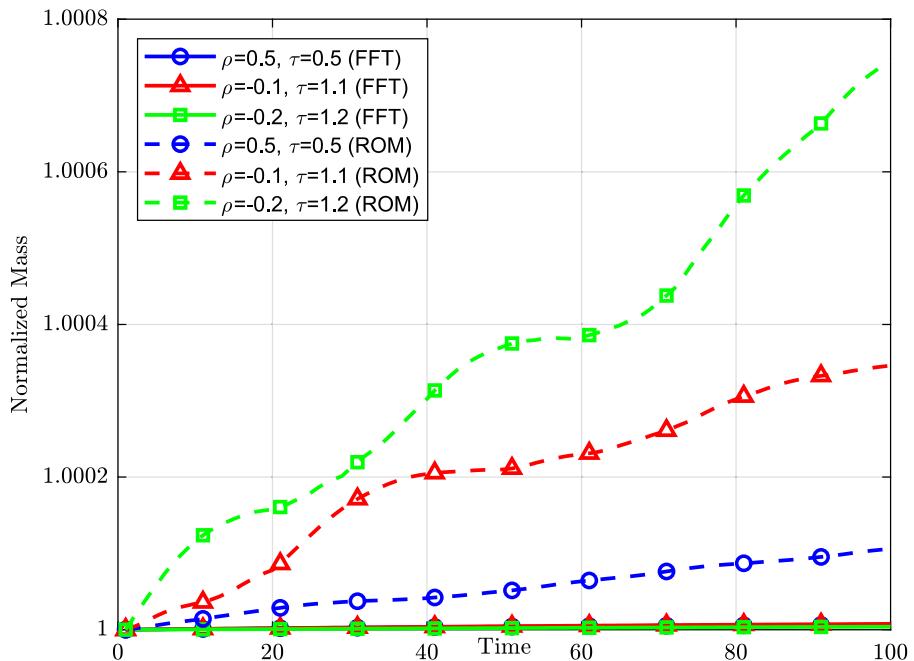


Fig. 14. Normalized mass evolution for FFT and ROM methods under three parameter combinations.

Table 3

The efficiency of predicting the CH equation using different reduced-order models is compared. $t_{Reduction}$ represents the average time for dimensionality reduction, $t_{Reconstruction}$ denotes the average time for model reconstruction, and $t_{Training}$ refers to the average training time. The computational gain is defined as the ratio of the time required to predict $100\Delta t$ using the ROM to the time needed for high-fidelity simulations.

Model	MSE loss	$t_{Reduction}$ (s)	$t_{Reconstruction}$ (s)	$t_{Training}$ (s)	Computational gain
VAE+LSTM	2.22×10^{-2}	1.87×10^{-3}	1.91×10^{-3}	2.90×10^3	201.71
CAE+LSTM	3.05×10^{-2}	1.56×10^{-3}	1.91×10^{-3}	1.44×10^4	219.54
PCA+LSTM	3.84×10^{-2}	1.99×10^{-1}	0.90×10^{-1}	2.09×10^5	2.66

by Hu et al. [71], the primary bottleneck in the ROM framework lies in the dimensionality reduction and subsequent reconstruction of microstructures, rather than the prediction process itself.

In our CH equation dataset, a single prediction by the reduced-order model corresponds to the high-fidelity model's computation over $100\Delta t$ intervals. The temporal evolution profiles for both models over

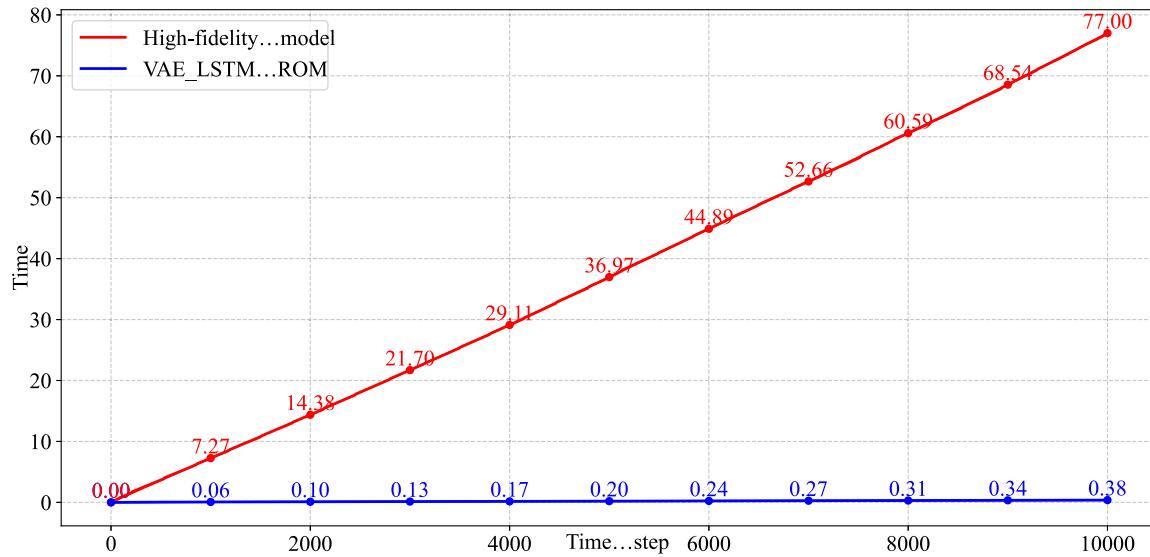


Fig. 15. The time curves depicting the duration required by the high-fidelity model and the ROM to evolve the phase field equations over $10000\Delta t$.

Table 4

The model's MSE loss values at different training data percentages.

Data volume	20%	40%	60%	80%	100%
MSE loss	0.4134	0.246	0.119	0.052	0.022

10,000 Δt intervals are illustrated in Fig. 15. In conclusion, by balancing both predictive efficiency and training time, the combined VAE+LSTM reduced-order model achieves an optimal trade-off in computational efficiency for predicting the evolution of CH equations.

3.6. The effect of training data size on model accuracy

This subsection investigates the impact of training data volume on the performance of our model. The effectiveness of deep learning models is closely tied to the size of the training dataset, as their ability to learn complex patterns and relationships in high-dimensional spaces depends critically on having sufficient data. Increasing the training data volume allows the model to capture more nuanced details in the data distribution, thereby improving generalization and reducing prediction errors. In this experiment, we varied the proportion of training data used, selecting 20%, 40%, 60%, 80%, and 100% of the training data for training the model. As shown in Table 4, the MSE consistently decreased as data volume increased, indicating significant performance improvements. This trend highlights that larger datasets enable the model to more effectively capture the complex mapping between inputs and outputs, leading to improved predictive accuracy. When the training data volume was limited to 20%, the MSE loss was 0.4134, indicating that the model struggled to capture underlying relationships in the data, resulting in higher prediction errors. In contrast, when using 100% of the training data, the MSE loss dropped to 0.022, demonstrating a significant improvement in the model's ability to extract features and accurately map inputs to outputs. These results underscore the pivotal role of larger datasets in improving model performance. The sensitivity of deep learning models to data volume can be attributed to their high degree of parameter flexibility and adaptability. Expanding the training dataset increases the sample space, reduces the sparsity of

sample relationships, and enhances the model's capacity to perform accurate fitting. In conclusion, the experiment reveals a strong correlation between data volume and model accuracy. As the amount of training data increases, the model's accuracy improves considerably.

4. Conclusions

In conclusion, this study has successfully developed an innovative end-to-end deep learning framework that integrates VAE and LSTM networks for reduced-order prediction modeling. The unified approach simplifies the design and training processes by effectively combining feature extraction and prediction within a single architecture. The results demonstrate that the VAE effectively captures essential features of the input data, mapping it to a low-dimensional latent space while maintaining high representation quality through a progressive channel reduction strategy. The LSTM component further enhances the model's capability by learning temporal dependencies, allowing for accurate predictions based on historical data. Validation using the CH equation shows the model's feasibility in real-world applications, and a detailed analysis of hyperparameters identifies optimal configurations that maximize performance. Comparisons with other dimensionality reduction and prediction methods highlight the advantages of the proposed framework, particularly in terms of predictive accuracy and computational efficiency. We focus on the two-dimensional case as it effectively demonstrates the core principles of our deep learning framework. We anticipate that the proposed method will offer even greater advantages in three-dimensional problems, particularly in terms of time efficiency. In the two-dimensional case, our model successfully captures the key features of the CH equation, providing significant computational savings and improved predictive accuracy compared to traditional methods. For three-dimensional problems, the model structure can be extended by simply adding an additional dimension, while the underlying learning mechanism remains unchanged. The CH equation, which describes phase separation, follows similar physical principles in both two and three dimensions, with the primary difference being the increased spatial complexity and computational demands in three dimensions. While training for three-dimensional problems may require more computational resources and time, the basic training process remains the same. Although this study focuses

on two-dimensional results, we expect the advantages demonstrated to carry over to three-dimensional applications, where time savings and accuracy improvements are likely to be even more pronounced. Future work will extend this framework to three-dimensional problems and assess its performance compared to traditional methods in higher-dimensional settings. Overall, this work contributes a valuable methodology for nonlinear reduced-order prediction modeling.

CRediT authorship contribution statement

Zhixian Lv: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Xin Song:** Writing – original draft, Visualization, Supervision, Software, Resources, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Jiachen Feng:** Writing – original draft, Visualization, Supervision, Resources, Methodology, Investigation, Data curation, Conceptualization. **Qing Xia:** Writing – review & editing, Writing – original draft, Validation, Software, Resources, Formal analysis, Data curation. **Binhu Xia:** Writing – original draft, Visualization, Software, Investigation, Data curation. **Yibao Li:** Writing – review & editing, Methodology, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 12271430), Natural Science Basic Research Program of Shaanxi, China (No. 2024JC225YBMS-016), Shaanxi Fundamental Science Research Project for Mathematics and Physics, China (No. 23JSQ054). The authors are grateful to the reviewers whose valuable suggestions and comments significantly improved the quality of this paper.

Data availability

No data was used for the research described in the article.

References

- [1] Wang Q, Hesthaven J, Ray D. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *J Comput Phys* 2019;384:289–307.
- [2] Chen J, Ma D, Zhang Z. A multiscale reduced basis method for Schrödinger equation with multiscale and random potentials. *Multiscale Model Simul* 2020;18:1409–34.
- [3] Fan D, Zhang B, Zhou Y, Noack BR. Optimization and sensitivity analysis of active drag reduction of a square-back ahmed body using machine learning control. *Phys Fluids* 2020;32.
- [4] Kou J, Clainche SL, Zhang W. A reduced-order model for compressible flows with buffeting condition using higher order dynamic mode decomposition with a mode selection criterion. *Phys Fluids* 2018;30.
- [5] Li K, Kou J, Zhang W. Aeroelastic reduced-order modeling for efficient static aeroelastic analysis considering geometric nonlinearity. *J Fluid Struct* 2024;124:104055.
- [6] dos Reis de Souza A, Vuillemin P, Poussot-Vassal C, Quero D. Gust load alleviation using reduced-order aeroelastic models and observer-based robust control. *J Guid Contr Dyn* 2023;46:949–57.
- [7] Xiao D, Heaney CE, Mottet L, Fang F, Lin W, Navon IM, et al. A reduced order model for turbulent flows in the urban environment using machine learning. *Build Env* 2019;148:323–37.
- [8] Masoumi-Verki S, Haghigat F, Eicker U. A review of advances towards efficient reduced-order models (rom) for predicting urban airflow and pollutant dispersion. *Build Env* 2022;216:108966.
- [9] Wang Y, Chung E, Fu S. A deep learning based reduced order modeling for stochastic underground flow problems. *J Comput Phys* 2022;467:111449.
- [10] Xia Q, Kim J, Xia B, Li Y. An unconditionally energy stable method for binary incompressible heat conductive fluids based on the phase-field model. *Comput Math Appl* 2022;123:26–39.
- [11] Yang J, Lee C, Jeong D, Kim J. A simple and explicit numerical method for the phase-field model for diblock copolymer melts. *Comput Mater Sci* 2022;205:111192.
- [12] Weng Z, Zhai S, Dai W, Yang Y, Mo Y. Stability and error estimates of strang splitting method for the nonlocal ternary conservative Allen–Cahn model. *J Comput Appl Math* 2024;441:115668.
- [13] Li Y, Lv Z, Xia Q. On the adaption of biological transport networks affected by complex domains. *Phys Fluids* 2024;36(10):101906.
- [14] Yoshizawa T, Nozawa RS, Jia TZ, Saio T, Mori E. Biological phase separation: cell biology meets biophysics. *Biophys Rev* 2020;12:519–39.
- [15] Mau J, Zhao J. Discovery of governing equations with recursive deep neural networks. *Commun Appl Math Comput* 2023;1–25.
- [16] Xie W, Feng J, Xia Q, Kim J, Li Y. Design of the shell-infill structures using a phase field-based topology optimization method. *Comput Methods Appl Mech Engrg* 2024;429:117138.
- [17] Song X, Xia Q, Kim J, Li Y. An unconditional energy stable data assimilation scheme for Navier–Stokes–Cahn–Hilliard equations with local discretized observed data. *Comput Math Appl* 2024;164:21–33.
- [18] Xia B, Yu R, Song X, Zhang X, Kim J. An efficient data assimilation algorithm using the Allen–Cahn equation. *Eng Anal Bound Elem* 2023;155:511–7.
- [19] Yang J, Tan Z, Wang J, Kim J. Modified diffuse interface fluid model and its consistent energy-stable computation in arbitrary domains. *J Comput Phys* 2023;488:112216.
- [20] Fang X, Pan Z, Ma R, Chen A. A multi-phase multi-species phase field model for non-uniform corrosion and chloride removal in concrete. *J Build Eng* 2024;82:108214.
- [21] Xia B, Xi X, Yu R, Zhang P. Unconditional energy-stable method for the Swift–Hohenberg equation over arbitrarily curved surfaces with second-order accuracy. *Appl Numer Math* 2024;198:192–201.
- [22] Ruan H, Rezaei S, Yang Y, Gross D, Xu BX. A thermo-mechanical phase-field fracture model: Application to hot cracking simulations in additive manufacturing. *J Mech Phys Solids* 2023;172:105169.
- [23] Xia Q, Sun G, Kim J, Li Y. Multi-scale modeling and simulation of additive manufacturing based on fused deposition technique. *Phys Fluids* 2023;35:3.
- [24] Prakash V, Behera AK, Rahaman MM. A phase-field model for thermo-mechanical fracture. *Math Mech Solids* 2023;28:533–61.
- [25] Li X, Shen J. Efficient linear and unconditionally energy stable schemes for the modified phase field crystal equation. *Sci China Math* 2021;1–18.
- [26] Zhao J, Wang Q, Yang X. Numerical approximations for a phase field dendritic crystal growth model based on the invariant energy quadratization approach. *Int J Numer Methods Eng* 2017;110:279–300.
- [27] Xia Q, Yang J, Kim J, Li Y. On the phase field based model for the crystalline transition and nucleation within the Lagrange multiplier framework. *J Comput Phys* 2024;113158.
- [28] Li Y, Xia Q, Kang S, Kwak S, Kim J. A practical algorithm for the design of multiple-sized porous scaffolds with triply periodic structures. *Math Comput Simulation* 2024;220:481–95.
- [29] Zhai S, Weng Z, Mo Y, Feng X. Energy dissipation and maximum bound principle preserving scheme for solving a nonlocal ternary Allen–Cahn model. *Comput Math Appl* 2024;155:150–64.
- [30] Chen H, Sun S. A new physics-preserving IMPES scheme for incompressible and immiscible two-phase flow in heterogeneous porous media. *J Comput Appl Math* 2021;381:113035.
- [31] Xu S. Recent progress in the phase-field dislocation dynamics method. *Comput Mater Sci* 2022;210:111419.
- [32] DeWitt S, Rudraraju S, Montiel D, Andrews WB, Thornton K. Prisms-pf: A general framework for phase-field modeling with a matrix-free finite element method. *Npj Comput Mater* 2020;6:29.
- [33] Li X, Shen S, Liu Z. New SAV-pressure correction methods for the Navier–Stokes equations: stability and error analysis. *Math Comp* 2022;323:141–67.
- [34] Kim J. Phase-field models for multi-component fluid flows. *Commun Comput Phys* 2012;12:613–61.
- [35] Bellis C, Ferrier R. Numerical homogenization by an adaptive fourier spectral method on non-uniform grids using optimal transport. *Comput Methods Appl Mech Engrg* 2024;419:116658.
- [36] Yang J. Surface phase-field surfactant fluid model and its practical closest point type finite difference computation. *Comput Math Appl* 2024;154:24–38.
- [37] Liao H, Ji B, Zhang L. An adaptive BDF2 implicit time-stepping method for the phase field crystal model. *IMA J Numer Anal* 2022;42:649–79.
- [38] Song X, Xia B, Li Y. An efficient data assimilation based unconditionally stable scheme for Cahn–Hilliard equation. *Comput Appl Math* 2024;43:121.
- [39] de Moraes EAB, Salehi H, Zayernouri M. Data-driven failure prediction in brittle materials: A phase field-based machine learning framework. *J Mach Learn Model Comput* 2021;2.

- [40] Luo X, Ye J, Ma P, Zhang L. Data-driven enhanced phase field models for highly accurate prediction of mode i and mode ii fracture. *Comput Methods Appl Mech Engrg* 2022;400:115535.
- [41] Chen X, Zhong W, Li T. Fast prediction of temperature and chemical species distributions in pulverized coal boiler using pod reduced-order modeling for cfd. *Energy* 2023;276:127663.
- [42] Chen J, Jin S, Lyu L. A deep learning based discontinuous Galerkin method for hyperbolic equations with discontinuous solutions and random uncertainties. 2021, arXiv preprint arXiv:2107.01127v1.
- [43] Zhao J. Discovering phase field models from image data with the pseudo-spectral physics informed neural networks. *Commun Appl Math Comput* 2021;106:357–69.
- [44] Kiyani E, Silber S, Kooshkbaghi M, Karttunen M. Machine-learning-based data-driven discovery of nonlinear phase-field dynamics. *Phys Rev E* 2022;106:065303.
- [45] Hernandez Q, Badia A, Gonzalez D, Chinesta F, Cueto E. Deep learning of thermodynamics-aware reduced-order models from data. *Comput Methods Appl Mech Engng* 2021;379:113763.
- [46] Wang J, Han Z, Jiang W, Kim J. 3D object feature extraction and classification using 3D MF-DFA. *Comput Vis Image Understand* 2023;232:103688.
- [47] Wang J, Xu H, Jiang W, Han Z, Kim J. A novel MF-DFA-phase-field hybrid MRIs classification system. *Expert Syst Appl* 2023;225:120071.
- [48] Zhang Z, Zhao J. General numerical framework to derive structure preserving reduced order models for thermodynamically consistent reversible-irreversible PDEs. 2023, arXiv preprint arXiv:2312.02363.
- [49] Herman E, Stewart JA, Dingreville R. A data-driven surrogate model to rapidly predict microstructure morphology during physical vapor deposition. *Appl Math Model* 2020;88:589–603.
- [50] Fu J, Xiao D, Fu R, Li C, Zhu C, Arcucci R, et al. Physics-data combined machine learning for parametric reduced-order modelling of nonlinear dynamical systems in small-data regimes. *Comput Methods Appl Mech Engng* 2023;404:115771.
- [51] Zhang X, Xie F, Ji T, Zhu Z, Zheng Y. Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. *Comput Methods Appl Mech Engng* 2021;373:113485.
- [52] Cheng S, Chen J, Anastasiou C, Angeli P, Matar OK, Guo Y, et al. Generalised latent assimilation in heterogeneous reduced spaces with machine learning surrogate models. *J Sci Comput* 2023;94:11.
- [53] Maulik R, Lusch B, Balaprakash P. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Phys Fluids* 2021;33.
- [54] Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. *Int Conf Mach Learn* 2017;1945–54.
- [55] Li Y, Liu R, Xia Q, He C, Li Z. First- and second-order unconditionally stable direct discretization methods for multi-component Cahn-Hilliard system on surfaces. *J Comput Appl Math* 2022;401:113778.
- [56] Yang J, Kim J. Energy dissipation-preserving time-dependent auxiliary variable method for the phase-field crystal and the Swift-Hohenberg models. *Numer Algorithms* 2022;1–30.
- [57] Cheng K, Wang C, Wise SM, Yue X. A second-order, weakly energy-stable pseudo-spectral scheme for the Cahn–Hilliard equation and its solution by the homogeneous linear iteration meth-od. *J Sci Comput* 2016;69:1083–114.
- [58] Yan Y, Chen W, Wang C, Wise SM. A second-order energy stable BDF numerical scheme for the Cahn–Hilliard equation. *Commun Comput Phys* 2018;23:572–602.
- [59] Diegel AE, Wang C, Wise SM. Stability and convergence of a second-order mixed finite element method for the Cahn–Hilliard equation. *IMA J Numer Anal* 2016;36:1867–97.
- [60] Cheng K, Feng W, Wang C, Wise SM. An energy stable fourth order finite difference scheme for the Cahn–Hilliard equation. *J Comput Appl Math* 2019;362:574–95.
- [61] Kim J, Tan Z, Yang J. Linear and conservative IMEX Runge–Kutta finite difference schemes with provable energy stability for the Cahn–Hilliard model in arbitrary domains. *Comput Math Appl* 2023;143:133–50.
- [62] Guo J, Wang C, Wise SM, Yue X. An H^2 convergence of a second-order convex-splitting, fi-nite difference scheme for the three-dimensional Cahn–Hilliard equation. *Commun Math Sci* 2016;14:489–515.
- [63] Li D, Quan C, Tang T. Stability and convergence analysis for the implicit-explicit method to the Cahn–Hilliard equation. *Math Comp* 2022;91:785–809.
- [64] Montes de Oca Zapiain D, Stewart JA, Dingreville R. Accelerating phase-field-based microstructure evolution predictions via surrogate models trained by machine learning methods. *Npj Comput Mater* 2021;7:3.
- [65] Loshchilov I, Hutter F. Sgdr: Stochastic gradient descent with warm restarts. 2016, arXiv preprint arXiv:1608.03983.
- [66] Chen W, Wang C, Wang X, Wise SM. Positivity-preserving, energy stable numerical schemes for the Cahn–Hilliard equation with logarithmic potential. *J Comput Phys X* 2019;3:100031.
- [67] Chen W, Jing J, Wang C, Wang X, Wise S. A modified Crank–Nicolson scheme for the Flo-ry-Huggins Cahn–Hilliard model. *Commun Comput Phys* 2022.
- [68] Duan J, Li X, Qiao Z. Convergence and stability analysis of energy stable and bound-preserving numerical schemes for binary fluid-surfactant phase-field equations. *Numer Methods Partial Differ Equ* 2024;40:e23125.
- [69] Chen W, Jing J, Wu H. A uniquely solvable, positivity-preserving and uncondition-ally energy stable numerical scheme for the functionalized Cahn–Hilliard equation with logarithmic potential. *J Sci Comput* 2023;96:75.
- [70] Kim J, Kwak S, Lee HG, Hwang Y, Ham S. A maximum principle of the Fourier spectral method for diffusion equations. *Electron Res Arch* 2023;31:5396–405.
- [71] Hu C, Martin S, Dingreville R. Accelerating phase-field predictions via recurrent neural networks learning the microstructure evolution in latent space. *Comput Methods Appl Mech Engrg* 2022;397:115128.