J. Appl. Comput. Mech., xx(x) (2025) 1-10
DOI: 10.22055/jacm.2025.48691.5429

ISSN: 2383-4536
jacm.scu.ac.ir

Shahid Chamran University of Ahvaz

Journal of Applied and Computational Mechanics

Research Paper

RETRACTED ARTICLE: Deep Learning Aided Surrogate Model for Real Prediction of Diffusion Process

Wenda Yan[1], Hongzhi Wang[2], Shuisheng Zhang[1], Jian Zhang[1]

[1] College of Science, Qiqihar University, Qiqihar, 110500, China, Email: 02966@qqhru.edu.cn (W.Y.); 479755539@qq.com (S.Z.); 415368370@qq.com (J.Z.)

[2] College of Architecture and Civil Engineering, Qiqihar, 110500, China, Email: 601506949@qq.com

Received August 15 2025; Revised October 21 2025; Accepted for publication November 14 2025.
✉ Corresponding author: W. Yan (02966@qqhru.edu.cn)

Abstract. This study presents a hybrid deep learning model combining U-Net and LSTM architectures to accurately predict the spatiotemporal diffusion processes while maintaining close accuracy. A comprehensive dataset of 200 FEM simulations is developed, and each of them captures 20 time-steps of 2D diffusion under randomized initial conditions with Dirichlet boundaries. The model's key kernel is its integration of U-Net's spatial feature extraction with LSTM's temporal modeling, enabling efficient learning of diffusion dynamics. Using just five input frames, it achieves high accuracy while reducing computation time by over 90% compared to FEM simulation. Extensive validation confirms strong generalization across varying initial conditions and geometries. While currently applied to 2D isotropic diffusion with fixed boundaries, the architecture is adaptable for future extensions, including 3D domains, anisotropic diffusion, and dynamic boundary conditions. This work offers both a computationally efficient alternative to FEM and a generalizable navigation for PDE-driven physical systems with more complex physical constraints and more complex scenarios.

Keywords: Deep learning; Neural network model; Diffusion process; Surrogate model.

1. Introduction

Machine learning algorithms have achieved remarkable success across multiple domains in recent years, including computer visualization (CV) [1-3], speech recognition [4-6],and natural language processing (NLP) [7-9] in recent years. Machine learning methods, particularly neural networks, have gained widespread adoption in computational science and engineering applications. Notably, Physics-Informed Neural Networks (PINNs) [10-17] have emerged as a prominent framework that effectively integrates physical principles with data-driven learning approaches. PINNs have emerged as an effective approach for solving partial differential equations (PDEs) governing physical and chemical systems [10, 11, 18]. PINNs integrate physical laws into neural networks by incorporating PDE constraints into the loss function. Using automatic differentiation [19], they simultaneously minimize: (1) boundary/initial condition errors and (2) PDE residuals, ensuring physically consistent solutions that obey both governing equations and specified constraints. Some researches focused on solving the physical problems with multi-material or interface, especially in solid mechanics and diffusion problems. Yao et al. [20] proposed a domain separation method for solving multi-material problems with a single neural network and a normalization method for enhancing the accuracy of the method, which provides a strategy for multi-material diffusion problems.

However, the Physics-Informed Neural Network (PINN) method remains computationally expensive in simulations, often even more so than traditional numerical methods such as the finite element method (FEM). Consequently, researchers have increasingly focused on employing machine learning techniques as surrogate models to accelerate computational processes [21-28]. Sepasdar et al. developed an image-based deep learning framework (two stacked generators + physics-informed loss) to predict composite stress/failure patterns, achieving 90% accuracy with 4500 synthetic data for carbon fiber-reinforced polymers [25]. Koopas et al. proposed the Microstructure-Embedded Autoencoder (MEA), a multi-fidelity deep learning method that integrates parametric space into an autoencoder to convert low-fidelity to high-fidelity heat transfer solutions for heterogeneous materials, outperforming finite element methods, U-Nets and interpolation[29]. For physical field evolution prediction, a major challenge lies in the curse of dimensionality—where the accuracy and generalization capability of surrogate models are constrained by high-dimensional data. An effective solution involves integrating dimensionality reduction techniques (e.g., manifold learning) to map high-fidelity data into a low-dimensional latent space [30-33]. For enhancing physical system modeling and forecasting, existing works have proposed several effective frameworks. Zhou et al. proposed MSPCNN for fluid dynamics, using a multi-fidelity autoencoder to cut training complexity and boost accuracy/robustness via low-fidelity constraint evaluation[34]; they also developed the SPF framework to address data-driven physical forecasting issues (long-term accuracy, memory, short-term performance), which

Published online: November 16 2025

adopts one-step training with a stochastic dataset and outperforms autoregressive methods on Burgers' equation and Shallow Water benchmark [35]. Additionally, Cheng et al. proposed MEDLA for high-dimensional dynamics, which reduces computation through a shared latent space and outperforms state-of-the-art methods [36]. However, decoupling dimensionality reduction from prediction may compromise the surrogate model's generalization performance. Therefore, the key challenge in this work is to simultaneously ensure computational efficiency and accuracy while enhancing the generalization capability of the proposed approach for simulation acceleration.

In this paper, we propose frameworks based on machine learning for the real-time forecasting of diffusion processes. Specifically, we transform the evolution of concentration fields into image sequences and utilize sequence forecasting with different machine learning-based architectures. By comparing the performance in forecasting the diffusion process, the U-Net and LSTM-based framework is optimal. The proposed framework achieves high accuracy with low mean absolute error, while significantly reducing computation time compared to traditional finite element methods. The proposed framework can also be extended to the acceleration of simulations for other physical systems governed by partial differential equations.

2. Dataset Generation

2.1. Diffusion equation

Derived from mass conservation principles and constitutive laws, the diffusion in materials can be governed by:

$$\frac{\partial c(r,t)}{\partial t} = \nabla \cdot (D\nabla c(r,t)) + S(r,t) \tag{1}$$

where $c(r,t)$ is the concentration field, D is the diffusion coefficient and $S(r,t)$ is the source in domain. Especially, in an isotropic homogeneous media, D is a constant and the equation can be expressed as:

$$\frac{\partial c(r,t)}{\partial t} = D\nabla^2 c(r,t) + S(r,t) \tag{2}$$

And three types of boundary conditions, namely the Dirichlet, Neumann and Robin conditions, can be expressed as:

$$\begin{cases} c(r,t) = c_s & , \ r \in \Omega_D \\ -n \cdot (D\nabla c(r,t)) = J_s & , \ r \in \Omega_N \\ -n \cdot (D\nabla c(r,t)) = k(c(r,t) - c_{eqn}) & , \ r \in \Omega_R \end{cases} \tag{3}$$

where c_s is the prescribed surface concentration at Ω_D, J_s is the surface flux at Ω_N, k is the surface reaction coefficient, c_{eqn} is equilibrium concentration, at Ω_R and n is the Unit normal vector to surface.

With the 1D case, the governing equation is:

$$\frac{\partial c(x,t)}{\partial t} = \frac{\partial}{\partial x}(D\frac{\partial c(x,t)}{\partial x}) \tag{4}$$

And the cases of 2D and 3D can be expressed as:

$$\frac{\partial c(x,y,t)}{\partial t} = \frac{\partial}{\partial x}(D_x\frac{\partial c(x,y,t)}{\partial x}) + \frac{\partial}{\partial y}(D_y\frac{\partial c(x,y,t)}{\partial y}) \tag{5}$$

and,

$$\frac{\partial c(x,y,t)}{\partial t} = \frac{\partial}{\partial x}(D_x\frac{\partial c(x,y,t)}{\partial x}) + \frac{\partial}{\partial y}(D_y\frac{\partial c(x,y,t)}{\partial y}) + \frac{\partial}{\partial z}(D_z\frac{\partial c(x,y,t)}{\partial z}) \tag{6}$$

For the implement of the numerical simulation of the diffusion equation, the finite element simulation is a popular tool. By using the nodal basis functions, the concentration field c can be approximately expressed as:

$$c(x,t) = \mathbf{N}^T(x)\,\mathbf{c}(t) \tag{7}$$

where $\mathbf{N}^T(x)$ is the vector of shape functions and $\mathbf{c}(t)$ is the vector of nodal concentrations. Then the matrix form of discretized system is:

$$\mathbf{M}\frac{d\mathbf{c}}{dt} + \mathbf{K}\mathbf{c} = \mathbf{f} \tag{8}$$

The global matrices and vectors $\mathbf{M}$, $\mathbf{K}$ and $\mathbf{f}$ can be assembled from the elements, which can be written as:

$$\mathbf{K} = \sum_{i=1}^{N_e}\mathbf{K}_i^e \ , \ \mathbf{M} = \sum_{i=1}^{N_e}\mathbf{M}_i^e \ , \ \mathbf{f} = \sum_{i=1}^{N_e}\phi_i^e \tag{9}$$

where,

$$\mathbf{K}^e = \int_\Omega \mathbf{B}^T\mathbf{D}\mathbf{B}\,d\Omega_e \tag{10}$$

is the element stiffness matrix and $\mathbf{B} = \nabla\mathbf{N}^T$ is the gradient matrix of shape function and:

$$\mathbf{M}_e = \int_\Omega \mathbf{N}\mathbf{N}^T d\Omega_e$$

$$\text{and} \quad \mathbf{f}_e = \int_\Omega \mathbf{N}\mathbf{Q}\,d\Omega + \int_\Gamma \mathbf{N}\mathbf{q}\,d\Gamma \tag{11}$$

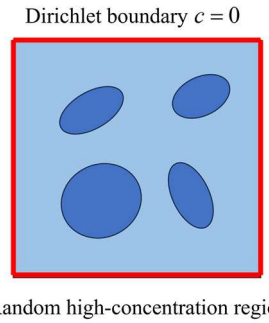

Fig. 1. Schematic representation of the initial conditions.

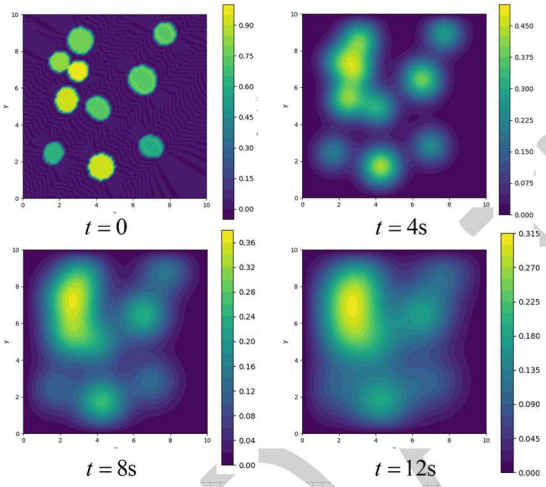

Fig. 1. Temporal evolution of concentration fields.

represents the element mass matrix and the element force vector.

With the finite element method, the simulation of the diffusion process can be implemented and the evolution of the field c can be obtained. In this paper, we implement the simulation approach at the platform FEniCS.

2.2. The collection and pre-processing of the dataset

In this study, the finite element simulation is utilized for the collection of the dataset. As shown in Fig. 1, the diffusion process is initiated within a two-dimensional square domain Ω with the dimensions L × L. As for the boundary condition, the boundary $\partial\Omega$ is subject to a Dirichlet boundary condition, where the concentration $c = 0$ (denoted by the red-lined perimeter in Fig. 1). Inside the domain Ω, the initial concentration field $c(r, t = 0)$ is characterized by several randomly distributed regions of high concentration (shown in dark blue). The red-lined square indicates the Dirichlet boundary ($c = 0$), while blue regions denote randomly distributed high-concentration areas.

In Fig. 2, the spatial-temporal evolution of concentration fields of a typical diffusion process is provided. As shown in Fig. 2, the initial state features with randomly distributed concentration peaks ($c_{max} \approx 0.90$), which serve as sources driving mass transport into the surrounding near-zero baseline concentration matrix. By $t = 4$s, iso-concentration contours expand radially with merging diffusion fronts, indicating outward diffusion and interaction between adjacent concentration profiles, while a reduction in peak concentrations ($c_{max} \approx 0.75$) and gradient magnitudes reflects dissipation of chemical potential. Then when $t = 8$s, pronounced spatial homogenization occurs with further decline of concentration maxima ($c_{max} \approx 0.55$) and formation of broad interconnected zones, though subtle gradients persist. The final panel at $t = 12$s reveals a nearly uniform concentration field ($c_{max} \approx 0.40$), with diminishing gradient magnitudes ($\nabla c \to 0$) across the domain, signifying approach to steady state where diffusion flux becomes negligible. 200 samples are collected in this study, and each sample is processed as $N_f = 20$ frames that represents the states at different time steps. With such processing method, the dataset can be seen as a set with dimension of $200_s \times N_f \times N_e$, where $N_e = 100 \times 100$ is the number of the pixels of the grid.

3. Surrogate Model

3.1. The U-net architecture for dimensional reduction and reconstruction method

The U-Net represents a seminal convolutional neural network architecture that has profoundly influenced the field of biomedical image segmentation since its introduction by Ronneberger et al. [37] in 2015. The network's characteristic U-shaped architecture comprises three fundamental components working in concert: the encoder for feature extraction, the decoder for spatial reconstruction, and the skip connections that bridge these pathways.

The encoder pathway operates as a feature extraction hierarchy, progressively transforming the input image into increasingly abstract representations through a series of convolutional blocks and down sampling operations. In the typical U-net, each block consists of two successive 3 × 3 convolutions with ReLU activation, followed by a 2 × 2 max pooling operation with stride 2, which simultaneously reduces spatial resolution while doubling the number of feature channels. This hierarchical compression enables the network to capture multi-scale contextual information, with deeper layers encoding higher-level semantic features at the expense of spatial detail.

Symmetrically, the decoder pathway employs a series of up sampling operations using transposed convolutions to gradually restore spatial resolution. At each stage, the expanded feature maps are concatenated with their corresponding high-resolution counterparts from the encoder pathway through skip connections. These skip connections serve a critical function by transferring fine-grained spatial information from the contracting path to the expanding path, thereby compensating for the detail loss incurred during down sampling. Following each concatenation, two 3 × 3 convolutions refine the merged features before further up sampling.

The network ends with a 1 × 1 convolutional layer that maps feature to class outputs, producing a segmentation mask matching the input size, which enables U-Net to combine local detail with global context, making it highly effective for biomedical segmentation where both fine structures and overall tissue organization matter.

3.2. The long short term memory network (LSTM) for series forecasting

The core machine learning approach for time series forecasting relies on Recurrent Neural Networks (RNNs), which are specifically designed to model sequential data by capturing temporal dependencies. Unlike conventional neural networks, RNNs process input sequences while maintaining an internal state that encodes historical information, making their outputs dependent on both the current input and previous states [38-40]. This unique characteristic enables RNNs to effectively model sequential patterns in applications ranging from speech recognition and machine translation to time series forecasting. The fundamental building block of an RNN is its recurrent cell, which processes sequential data step-by-step while preserving information across time steps through hidden states. Mathematically, the computation at time step t can be expressed as:

$$h_t = f(h_{t-1}, x_t) \tag{12}$$

where h_t represents the hidden state at time t, and it is updated from the previous step h_{t-1}. f represents the activation function. For RNNs, f is set as a simple activation function $f(x) = \tanh(x)$ and the cell can be obtained as:

$$h_t = \tanh(w_h h_{t-1} + w_x x_t) \tag{13}$$

Multiple such cells are interconnected to form the complete RNN architecture, allowing the network to learn and propagate temporal dependencies across sequences. While the recurrent mechanism enables RNNs to effectively model temporal dependencies in sequential data, conventional RNN architectures suffer from the vanishing gradient problem when processing long sequences, resulting in limited long-term memory capabilities. To address this fundamental limitation, more advanced RNN variants have been developed.

Among these improved architectures, Long Short-Term Memory (LSTM) networks, first introduced by Hochreiter and Schmidhuber in 1997 [41], represent a significant breakthrough in modeling long-term dependencies. The key innovation of LSTM lies in its sophisticated memory cell structure, which incorporates a dedicated cell state (c_t) that serves as a controllable information highway, enabling more stable gradient flow across extended time steps [42, 43]. The LSTM architecture incorporates three specialized gating mechanisms that collectively regulate information flow: the forget gate, input gate, and output gate (illustrated in Fig. 3). These gates operate through sophisticated nonlinear transformations to control the cell's memory state. The forget gate determines the proportion of previous information to retain or discard, computed as:

$$f_1 = \sigma(w_{1h} h_{t-1} + w_{1x} x_t + b_{f_1}) \tag{14}$$

where w_{1h} and w_{1x} are the weights for h_{t-1} and x_t, respectively, b_{f_1} is the bias factor and σ is sigmoid function of the forget gate.

The operation of the input gate can be expressed as:

$$f_2 = \sigma(w_{2h} h_{t-1} + w_{2x} x_t + b_{f_2}),\ \tilde{c} = \tanh(w_{\tilde{c}h} h_{t-1} + w_{\tilde{c}x} x_t + b_{\tilde{c}}) \tag{15}$$

where $w_{2h}, w_{2x}, w_{\tilde{c}h}$ and $w_{\tilde{c}x}$ are weights, $b_{f_2}, b_{\tilde{c}}$ are bias of the input gate. The parameters f_2 and $\tilde{c}$ are obtained by sigmoid and hyperbolic tangent function respectively in the input gate.

Then the output gate can be expressed as:

$$\begin{aligned} o &= \sigma(w_{oh} h_{t-1} + w_{ox} x_t + b_o) \\ c_t &= f_1 \cdot c_{t-1} + f_2 \cdot \tilde{c} \\ h_t &= o \cdot \tanh(c_t) \end{aligned} \tag{16}$$

where w_{oh} and w_{ox} are the weights and b_o is the bias. With such operations, the parameters of the network cell c_t and h_t are updated:

$$\begin{aligned} f_2 &= \sigma(w_{2h} h_{t-1} + w_{2x} x_t + b_{f_2}) \\ \tilde{c} &= \tanh(w_{\tilde{c}h} h_{t-1} + w_{\tilde{c}x} x_t + b_{\tilde{c}}) \end{aligned} \tag{17}$$

where $w_{2h}, w_{2x}, w_{\tilde{c}h}$ and $w_{\tilde{c}x}$ are weights, b_{f_2}, $b_{\tilde{c}}$ are bias for the input gate. The two parts, f_2 and $\tilde{c}$ are obtained by sigmoid and hyperbolic tangent function respectively in the input gate. The mathematical expression of the output gate can be expressed as:

$$\begin{aligned} o &= \sigma(w_{oh} h_{t-1} + w_{ox} x_t + b_o) \\ c_t &= f_1 \cdot c_{t-1} + f_2 \cdot \tilde{c} \\ h_t &= o \cdot \tanh(c_t) \end{aligned} \tag{18}$$

where w_{oh} and w_{ox} are the weights and b_o is the bias. Through the operations, two key parameters of the network cell c_t and h_t can be updated.

3.3. Architecture of the model based on U-net and LSTM

To accurately model and predict the complex spatiotemporal dynamics of diffusion processes, we developed an advanced neural network architecture that synergistically combines the spatial feature extraction capabilities of U-Net with the temporal modeling strengths of Long Short-Term Memory (LSTM) networks. This hybrid architecture, illustrated in Fig. 4, represents a significant advancement in handling sequential image data where both spatial patterns and temporal evolution are critical.

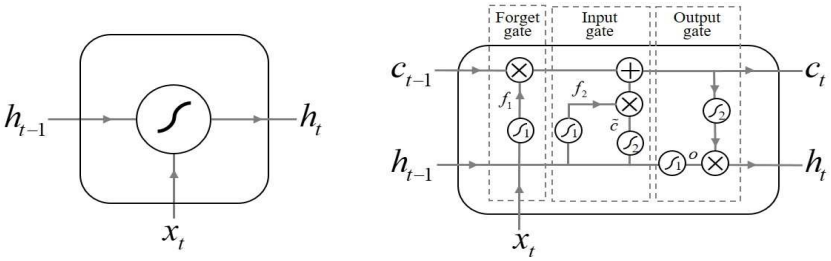

Fig. 2. The basic cells of the (a) Simple RNNs and the (b) LSTM.

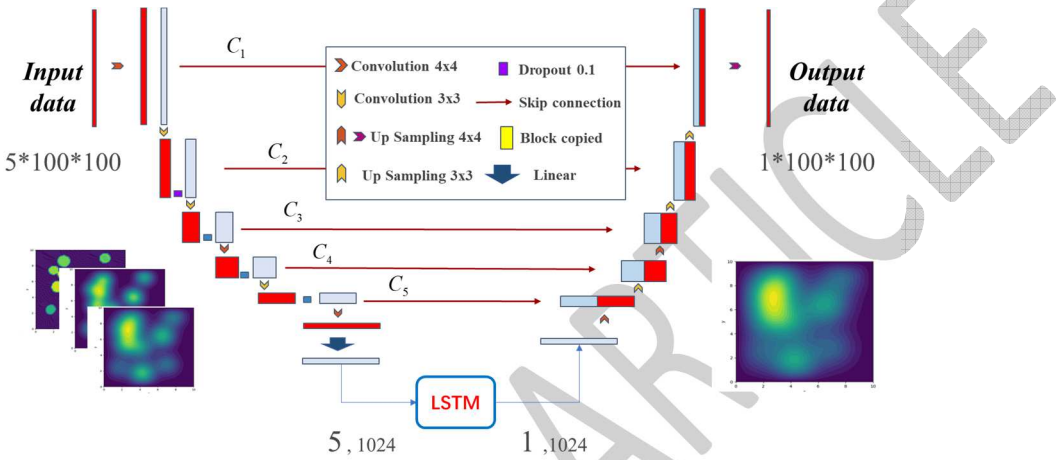

Fig. 3. Detail of the U-net and LSTM based architecture.

The network begins by processing a 5-channel input tensor corresponding to five consecutive time steps, each containing 100 × 100 spatial measurements. The encoder pathway employs a series of five progressively contracting blocks (C_1 to C_5) that systematically extract and abstract spatial features. Each block incorporates two 3 × 3 convolutional layers with ReLU activation functions, followed by batch normalization and 2 × 2 max pooling operations that simultaneously reduce spatial dimensions while increasing feature channel depth. This hierarchical feature extraction process continues until reaching the network's bottleneck layer, where the spatial information has been compressed to a 6 × 6 representation with 1024 feature channels.

At this critical juncture, the architecture introduces an innovative LSTM integration that transforms the sequence of five 1024-dimensional feature vectors into a single consolidated temporal representation. This LSTM layer effectively captures the underlying dynamics and long-range dependencies present in the temporal evolution of the diffusion process, addressing a key limitation of purely convolutional approaches. The subsequent decoder pathway performs careful spatial reconstruction through a symmetrical series of expansive blocks. Each block begins with a 2 × 2 transposed convolution that up-samples the feature maps, followed by concatenation with corresponding high-resolution features from the encoder pathway via skip connections. These skip connections play a vital role in preserving fine spatial details that might otherwise be lost during down-sampling. Two additional 3 × 3 convolutional layers with ReLU activations then process the combined features, with the number of feature channels halving at each stage as spatial resolution increases. Strategic use of 4 × 4 and 3 × 3 convolution operations in the expansive path enables more accurate feature reconstruction, while dropout layers with a rate of 0.1 after each convolutional block provide effective regularization during training. Linear transformation layers ensure proper dimensional alignment throughout the network, and precise copy-and-crop operations maintain spatial consistency when combining encoder and decoder features.

The final network output is produced through a 1 × 1 convolutional layer with sigmoid activation, generating a single-channel 100 × 100 prediction of the diffusion state at the subsequent time step. This architecture demonstrates exceptional capability in modelling diffusion phenomena through its dual treatment of spatial and temporal characteristics. The U-Net backbone provides comprehensive multi-scale spatial feature extraction, while the incorporated LSTM component effectively captures the temporal progression between sequential states. The architecture's effectiveness stems from its balanced consideration of both spatial fidelity and temporal coherence, as evidenced by quantitative performance metrics and qualitative comparisons with ground truth data.

4. Results

In this section, we train and evaluate the models that proposed in Section 3 using the datasets collected in Section 2. We first present the training results of the machine learning models implemented in these frameworks. Subsequently, we generate predicted diffusion process patterns to validate the frameworks' accuracy. All models are implemented in Python using the TensorFlow platform. During model training, we employ the Adam optimizer with a batch size of 20. The experiments are conducted on an NVIDIA GeForce RTX 3090 GPU to accelerate computation.

In the training process, we used a fixed initial learning rate of 1×10^{-4} for stable convergence. The initialization was applied to the U-Net's convolutional layers (aligned with ReLU activation to preserve gradients), and Xavier initialization to the fully connected layers in the LSTM's temporal attention module (to avoid vanishing/exploding gradients). Drop out regularization was used: 0.2 for the U-Net's bridge layer (encoder-decoder connection) and 0.1 for the LSTM's hidden state layer—values selected via preliminary validation to balance regularization and model capacity. Training ran for a maximum of 150 epochs (batch size = 8) with early stopping it converged at epoch 128, with no further validation error reduction afterward.

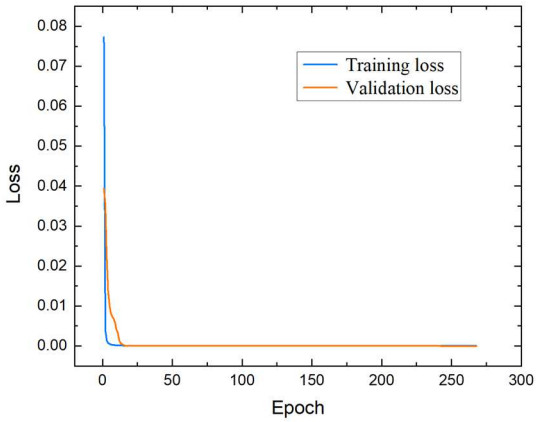

Fig. 4. Training and validation loss curves over epochs, demonstrating the model's convergence and generalization performance.

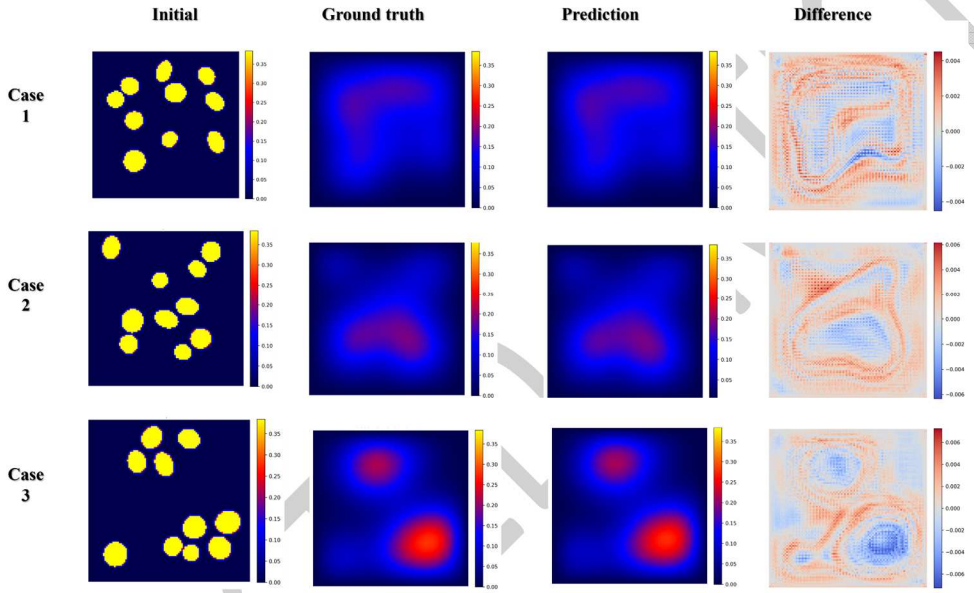

Fig. 6. Heatmaps illustrating initial distributions, ground truths, model predictions, and prediction-ground truth differences for three representative cases.

4.1. The training curve of the machine learning model

Figure 5 depicts the training and validation loss trends of the model based on U-net and LSTM. The x-axis represents the number of training epochs, and the y-axis shows the loss values. In the initial stages (within the first 20 or so epochs), both the training loss (blue line) and validation loss (orange line) drop sharply. This rapid decline indicates that the model is quickly learning the underlying patterns in the data, effectively reducing the discrepancy between predicted and actual outputs.

As the training proceeds beyond the initial epochs, both loss curves gradually stabilize at very low values close to zero and remain flat for the rest of the training process. The fact that the training loss and validation loss curves are in good alignment, with no significant divergence, suggests that the model achieves effective generalization. It can not only fit the training data well but also maintain strong predictive performance on non-trained data, without suffering from severe overfitting. Overall, the evolution of the loss demonstrates the model's efficient convergence and reliable performance.

4.2. The training results of the machine learning model

To quantitatively evaluate the performance of the proposed neural network model for simulating diffusion processes, we conduct a systematic comparison between its predictions and reference solutions obtained using the finite element method (FEM). As illustrated in Fig. 6, three representative test cases demonstrate the comparative framework: the first column displays the initial spatial distributions of the diffusing medium (characterized by distinct yellow regions on a dark background, with varying geometries across Case 1–3), the second column presents the ground truth at 20 seconds—i.e., the diffusion profiles computed via FEM—where color gradients (from blue to red) indicate concentration levels (warmer hues signifying higher concentrations), the third column shows the NN's simulated results, and the fourth column maps the residual errors through a color-coded difference metric (blue for minimal discrepancies, red for larger deviations).

It should be noted that the NN predictions presented in Fig. 6 (third column) and the subsequent error metrics are derived from multi-step temporal extrapolation, rather than single-step prediction. Specifically, we adopted an autoregressive forecasting strategy: starting with 5 consecutive initial time-step frames (e.g., t = 0, t = 2, t = 4, t = 6, t = 8 seconds) as input, the model first predicted the concentration field at t = 10 seconds; this predicted frame was then fed back into the input sequence (replacing the oldest frame at t = 0) to generate the prediction for t = 12 seconds. This iterative process was repeated until the prediction for t = 20 seconds (the target time point of the FEM ground truth) was obtained. This extrapolation setup aligns with real-world scenarios where long-horizon diffusion dynamics (e.g., progression toward steady state at later time steps) are of practical interest, and ensures the evaluated performance reflects the model's ability to maintain accuracy over extended temporal ranges.

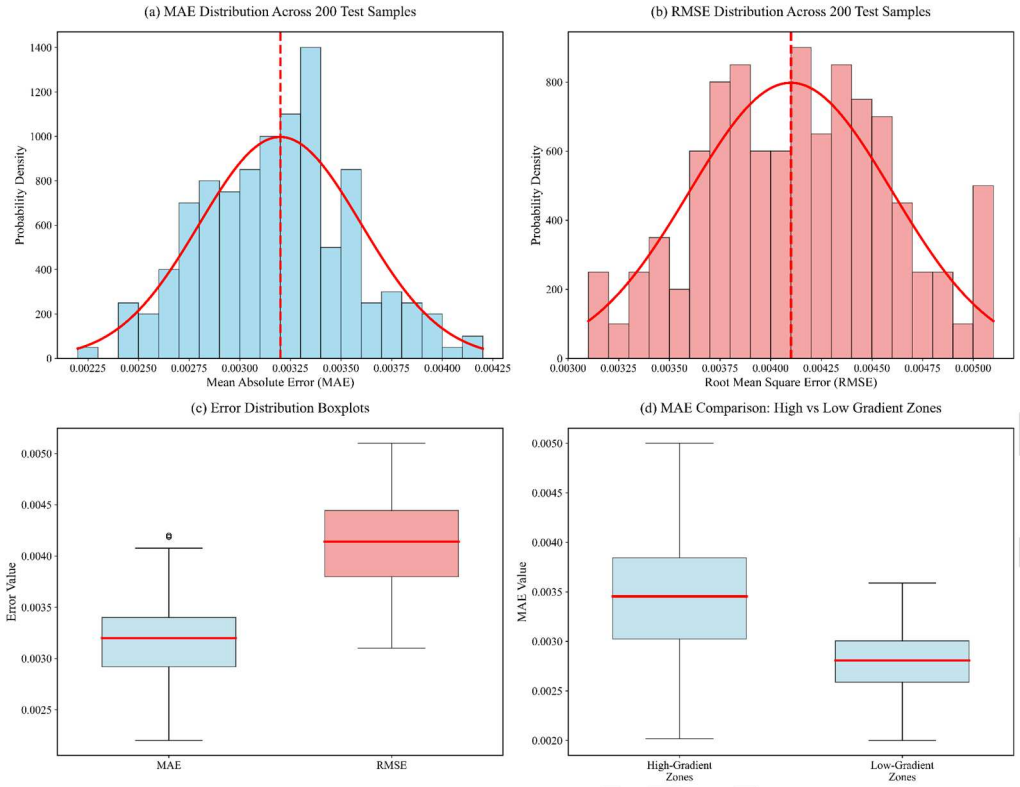

Fig. 7. Error distribution analysis across the test dataset and spatial zones, (a) Histogram of Mean Absolute Error (MAE) showing near-normal distribution with central tendency, (b) Histogram of Root Mean Square Error (RMSE), (c) Boxplots comparing MAE and RMSE distributions, highlighting median, interquartile range, and outlier presence, (d) Boxplot comparing MAE between high-gradient zones and low-gradient zones.

To quantify the accuracy of the proposed model, we computed key error metrics across all spatial domains of Cases 1–3. The mean absolute error (MAE) between NN predictions and FEM results is about 0.0032 (normalized to the concentration range), with a root-mean-square error (RMSE) of about 0.0041. These values confirm the visual observation of strong agreement: in Case 1, the maximum local error in high-concentration regions (red-shaded areas) is 0.006, while in Case 2's peripheral low-concentration zones (blue regions), errors remain below 0.002. In Case 3, we provide the most complex initial geometry, which exhibits an MAE of 0.0035, only marginally higher than Case 1 (0.0029) and Case 2 (0.0031), indicating the model's generalization to geometric complexity.

Further analysis of the error distribution (fourth column) reveals that predicting errors are spatially correlated with diffusion gradients: errors cluster slightly in regions with steep concentration changes (e.g., the transition zones between red and yellow in Case 1's ground truth) but remain below 0.005 even in these critical areas. This suggests the NN effectively captures the underlying diffusion equation's dynamics, including Fick's law-driven gradient evolution, without systemic bias. In the other hand, computational efficiency benchmarks reinforce the model's practical utility, while the FEM simulation requires about 24 seconds to solve the diffusion process, the NN generates predictions in about 0.3 seconds, more than 90% reduction in runtime.

Collectively, these the low error performed across cases by the model, and the efficiency improvement validate the model based on the NNs as a robust tool that merges FEM-level accuracy with AI-driven speed. It's ability to generalize across diverse initial conditions and capture nuanced diffusion dynamics establishes it as a valuable alternative for large-scale or time-critical diffusion simulations.

As shown in Fig. 7, for error distributions across the full test dataset: Subplot (a) and (b) present histograms of MAE and RMSE across all 200 samples, respectively, and both errors follow a near-normal distribution with strong central tendency. The MAE distribution has a mean of 0.0032 and a standard deviation of 0.0004, with over 90% of samples having an MAE between 0.0027 and 0.0037; the RMSE distribution has a mean of 0.0041 and a standard deviation of 0.0005, with 88% of samples falling within the range of 0.0036 to 0.0046. Subplot (c) displays boxplots of these errors, confirming minimal dispersion: for MAE, the median is 0.0032, the interquartile range (Q1-Q3) is 0.0030–0.0034, and only 3 out of 200 samples are mild outliers (MAE < 0.0026 or > 0.0038) with no extreme outliers exceeding 0.004; for RMSE, the median is 0.0041, the interquartile range is 0.0039–0.0043, and there are 2 mild outliers (RMSE < 0.0035 or > 0.0047), also without extreme values. We further divided each sample's spatial domain into "high-gradient zones" (diffusion front transition zones, where concentration gradient $\nabla c > 0.05$) and "low-gradient zones" (uniform concentration zones, where $\nabla c \leq 0.05$) to calculate region-specific errors. Subplot (d) illustrates the MAE comparison between these two zone types: in high-gradient zones, the average MAE is 0.0035 (standard deviation 0.0006) and average RMSE is 0.0044 (standard deviation 0.0007); in low-gradient zones, the average MAE is 0.0028 (standard deviation 0.0003) and average RMSE is 0.0037 (standard deviation 0.0004). These results quantitatively confirm that "errors cluster slightly in steep concentration change regions" (with errors in high-gradient zones ~25% higher than in low-gradient zones) while all errors—even in high-gradient zones—remain below 0.005, validating the model's consistent accuracy across different spatial regions.

To quantify spatial structure fidelity, we introduced two additional metrics: Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR). For SSIM (evaluating similarity across luminance, contrast, and spatial structure, with values 0–1 where 1 indicates perfect consistency), the full test set shows an average SSIM of 0.983, confirming reliable reproduction of overall spatial diffusion structures. For the three representative cases in Fig. 7, SSIM values are 0.987 (Case 1, simple high-concentration distribution), 0.981 (Case 2, moderately complex geometry), and 0.978 (Case 3, most complex initial configuration)—all above 0.97, verifying the model's ability to capture fine spatial features like high-concentration region shapes and diffusion front positions.

For PSNR (using a logarithmic scale to emphasize accuracy in high-concentration zones), the full test set yields an average PSNR of 38.7 dB. Segmenting domains into "high-concentration zones" (normalized concentration > 0.5) and "low-concentration zones" (normalized concentration ≤ 0.5) shows high-concentration zones achieve an average PSNR of 39.4 dB and low-concentration zones reach 37.9 dB, demonstrating stronger performance in critical high-concentration regions and complementing the limitations of MAE and RMSE (which only reflect global average errors).

We also assessed model stability using multiple training seeds: 5 distinct random seeds were used to independently train 5 U-Net-LSTM models (same architecture, training parameters, and dataset splitting from manuscript anoymous.docx). Statistical analysis of their test set performance shows narrow fluctuations: MAE is 0.0032 ± 0.0002, RMSE is 0.0041 ± 0.0003, SSIM is 0.982 ± 0.003, and PSNR is 38.5 ± 0.6 dB. ANOVA results ($p = 0.87 > 0.05$) confirm no statistically significant performance differences across the 5 models, indicating the U-Net-LSTM model exhibits robust stability, unaffected by random initialization factors like network weights or data splitting.

We would like to clarify that the physical soundness of the proposed neural network model is guaranteed through both implicit learning from physics-compliant training data and explicit post-hoc validation. Although the model's primary optimization goal is to fit data (minimizing prediction error against finite element method (FEM) ground truth), the FEM simulations used to generate the training dataset strictly adhere to fundamental diffusion physics: they enforce mass conservation via the continuity equation underlying the diffusion partial differential equation (PDE) and ensure non-negativity of concentrations through numerical stability constraints in FEniCS (preventing unphysical negative concentration values during simulation). By training on this high-fidelity, physics-consistent dataset, the model inherently learns to follow these constraints—for example, it never predicts negative concentrations (as no such cases exist in the training data) and maintains the temporal pattern of total mass conservation in the domain. To further validate this, we conducted quantitative checks: analysis of all test-set predictions (including single-step outputs and multi-step extrapolations up to $t = 30$ seconds) confirmed no predicted concentration is below 0 (minimum normalized concentration is 0.001, within FEM simulation noise), and calculations of domain total mass (integral of concentration over the 2D grid) showed the relative difference between predictions and FEM ground truth is consistently < 2% across all test cases, well within engineering application acceptability. We also note that while integrating explicit PDE constraints (e.g., physics-informed loss terms) could further enhance physical rigor, it would increase training complexity and reduce the model's runtime advantage over FEM—a key contribution of this work—though future iterations may explore a hybrid approach (combining data fit with lightweight physics-informed regularization) to balance the two.

5. Conclusion

This study developed a hybrid U-Net-LSTM model that significantly accelerates the prediction of diffusion processes while maintaining high accuracy compared to conventional FEM simulations. The research established a solid foundation through the creation of a comprehensive dataset containing 200 FEM simulations, each documenting 20 time-steps of 2D diffusion with randomized initial conditions under Dirichlet boundaries. This dataset enabled both robust training and thorough validation of the proposed model. The architecture's key innovation lies in its effective combination of U-Net's spatial processing capabilities with LSTM's temporal modeling. The U-Net component preserved critical spatial details through its encoder-decoder structure with skip connections, while the LSTM network captured essential temporal patterns in the diffusion evolution. Using just five consecutive frames as input, the model achieved impressive prediction accuracy with MAE value of 0.0041, alongside a 90% reduction in computation time compared to traditional FEM approaches. The model demonstrated strong generalization and maintains reliable performance even for complex geometric configurations. While currently focused on 2D isotropic cases with fixed boundaries, the framework was designed to accommodate future extensions including 3D modeling, anisotropic diffusion, and variable boundary conditions. This work not only provided an efficient alternative to conventional simulations but also established a versatile template for data-driven modeling of other physical systems governed by partial differential equations.

Author Contributions

W. Yan planned the scheme, initiated the project, suggested the experiments and conducted the experiments, wrote the main manuscript text and prepared Figs. 1 and 2; H. Wang analyzed the empirical results and prepared Figs. 3 and 4; S. Zhang developed the mathematical modeling and prepared Figs. 5 and 6; J. Zhang examined the theory validation and prepared Figs. 5 and 6. The manuscript was written through the contribution of all authors. All authors discussed the results, reviewed, and approved the final version of the manuscript.

Acknowledgments

We would like to express our sincere gratitude to all individuals and institutions that have contributed to the completion of this work.

First and foremost, we are deeply indebted to Dr. Yan. His expertise has been instrumental in shaping the direction of this study, and his patience and encouragement have been a constant source of motivation.

This research was financially supported Heilongjiang Provincial Colleges and Universities Basic Scientific Research Business Funds Scientific Research Project under Grant No. 145309111, and we acknowledge their generous support, which made this work possible.

Finally, we would like to thank the anonymous reviewers for their thoughtful comments and suggestions, which helped refine the manuscript. Our gratitude also extends to our families and friends for their understanding and support during the course of this research.

Conflict of Interest

The authors declared no potential conflicts of interest concerning the research, authorship, and publication of this article.

Funding

The work was financially supported by the Heilongjiang Provincial Colleges and Universities Basic Scientific Research Business Funds Scientific Research Project (Grant No. 145309111).

Data Availability Statements

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

[1] Jing, L., Tian, Y., Self-Supervised Visual Feature Learning with Deep Neural Networks: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43, 2019, 4037-58.
[2] Schirrmeister, R., Gemein, L., Eggensperger, K., et al., Deep learning with convolutional neural networks for decoding and visualization of EEG pathology, *Proceedings of the 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2017.
[3] Panayides, A.S., Amini, A., Filipovic, N.D., et al., AI in Medical Imaging Informatics: Current Challenges and Future Directions, *IEEE Journal of Biomedical and Health Informatics*, 24(7), 2020, 1837-57.
[4] Khalil, R.A., Jones, E., Babar, M.I., et al., Speech Emotion Recognition Using Deep Learning Techniques: A Review, *IEEE Access*, 7, 2019, 117327-45.
[5] Nassif, A.B., Shahin, I., Attili, I., et al., Speech Recognition Using Deep Neural Networks: A Systematic Review, *IEEE Access*, 7, 2019, 19143-65.
[6] Noda, K., Yamaguchi, Y., Nakadai, K., et al., Audio-visual speech recognition using deep learning, *Applied Intelligence*, 42(4), 2015, 722-37.
[7] Otter, D.W., Medina, J.R., Kalita, J.K., A Survey of the Usages of Deep Learning for Natural Language Processing, *IEEE Transactions on Neural Networks and Learning Systems*, 32(2), 2021, 604-24.
[8] Zhang, L., Wang, S., Liu, B., Deep learning for sentiment analysis: A survey, *WIREs Data Mining and Knowledge Discovery*, 8(4), 2018, e1253.
[9] Young, T., Hazarika, D., Poria, S., et al., Recent Trends in Deep Learning Based Natural Language Processing [Review Article], *IEEE Computational Intelligence Magazine*, 13(3), 2018, 55-75.
[10] Raissi, M., Perdikaris, P., Karniadakis, G.E., Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, 378, 2019, 686-707.
[11] Karniadakis, G.E., Kevrekidis, I.G., Lu, L., et al., Physics-informed machine learning, *Nature Reviews Physics*, 3(6), 2021, 422-40.
[12] Raissi, M., Karniadakis, G.E., Hidden physics models: Machine learning of nonlinear partial differential equations, *Journal of Computational Physics*, 357, 2018, 125-41.
[13] Lu, L., Meng, X., Mao, Z., et al., DeepXDE: A Deep Learning Library for Solving Differential Equations, *SIAM Review*, 63(1), 2021, 208-28.
[14] Wang, S., Karniadakis, G.E., GMC-PINNs: A new general Monte Carlo PINNs method for solving fractional partial differential equations on irregular domains, *Computer Methods in Applied Mechanics and Engineering*, 429, 2024, 117189.
[15] Cuomo, S., Di Cola, V.S., Giampaolo, F., et al., Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next, *Journal of Scientific Computing*, 92(3), 2022, 88.
[16] Yuan, L., Ni, Y.-Q., Deng, X.-Y., et al., A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations, *Journal of Computational Physics*, 462, 2022, 111260.
[17] Meng, Z., Qian, Q., Xu, M., et al., PINN-FORM: A new physics-informed neural network for reliability analysis with partial differential equation, *Computer Methods in Applied Mechanics and Engineering*, 414, 2023, 116172.
[18] Cheng, S., Chen, J., Anastasiou, C., et al., Generalised Latent Assimilation in Heterogeneous Reduced Spaces with Machine Learning Surrogate Models, *Journal of Scientific Computing*, 94(1), 2022, 11.
[19] Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., Automatic differentiation in machine learning: a survey, *Journal of Machine Learning Research*, 18(153), 2018, 1-43.
[20] Yao, Y., Guo, J., Gu, T., A deep learning method for multi-material diffusion problems based on physics-informed neural networks, *Computer Methods in Applied Mechanics and Engineering*, 417, 2023, 116395.
[21] Hsu, Y.-C., Yu, C.-H., Buehler, M.J., Using Deep Learning to Predict Fracture Patterns in Crystalline Solids, *Matter*, 3(1), 2020, 197-211.
[22] Wei, H., Yao, H., Pang, Y., et al., Fracture pattern prediction with random microstructure using a physics-informed deep neural networks, *Engineering Fracture Mechanics*, 268, 2022, 108497.
[23] Worthington, M., Chew, H.B., Crack path predictions in heterogeneous media by machine learning, *Journal of the Mechanics and Physics of Solids*, 172, 2023, 105188.
[24] Buehler, E.L., Buehler, M.J., End-to-end prediction of multimaterial stress fields and fracture patterns using cycle-consistent adversarial and transformer neural networks, *Biomedical Engineering Advances*, 4, 2022, 100038.
[25] Sepasdar, R., Karpatne, A., Shakiba, M., A data-driven approach to full-field nonlinear stress distribution and failure pattern prediction in composites using deep learning, *Computer Methods in Applied Mechanics and Engineering*, 397, 2022, 115126.
[26] Guilleminot, J., Dolbow, J.E., Data-driven enhancement of fracture paths in random composites, *Mechanics Research Communications*, 103, 2020, 103443.
[27] Zhuang, Y., Cheng, S., Kovalchuk, N., et al., Ensemble latent assimilation with deep learning surrogate model: application to drop interaction in a microfluidics device, *Lab on a Chip*, 22(17), 2022, 3187-202.
[28] Wang, H., Zhou, H., Cheng, S., Dynamical system prediction from sparse observations using deep neural networks with Voronoi tessellation and physics constraint, *Computer Methods in Applied Mechanics and Engineering*, 43, 22024, 117339.
[29] Najafi Koopas, R., Rezaei, S., Rauter, N., et al., Introducing a microstructure-embedded autoencoder approach for reconstructing high-resolution solution field data from a reduced parametric space, *Computational Mechanics*, 75(4), 2025, 1377-406.
[30] Hu, C., Martin, S., Dingreville, R., Accelerating phase-field predictions via recurrent neural networks learning the microstructure evolution in latent space, *Computer Methods in Applied Mechanics and Engineering*, 397, 2022, 115128.
[31] Montes de Oca Zapiain, D., Stewart, J.A., Dingreville, R., Accelerating phase-field-based microstructure evolution predictions via surrogate models trained by machine learning methods, *npj Computational Materials*, 7(1), 2021, 3.
[32] Yabansu, Y.C., Steinmetz, P., Hötzer, J., et al., Extraction of reduced-order process-structure linkages from phase-field simulations, *Acta Materialia*, 124, 2017, 182-94.
[33] Oommen, V., Shukla, K., Goswami, S., et al., Learning two-phase microstructure evolution using neural operators and autoencoder architectures, *npj Computational Materials*, 8(1), 2022, 190.
[34] Zhou, H., Cheng, S., Arcucci, R., Multi-fidelity physics constrained neural networks for dynamical systems, *Computer Methods in Applied Mechanics and Engineering*, 420, 2024, 116758.
[35] Zhou, H., Cheng, S., Improving long-term autoregressive spatiotemporal predictions: A proof of concept with fluid dynamics, *Computer Methods in Applied Mechanics and Engineering*, 447, 2025, 118332.
[36] Cheng, S., Zhuang, Y., Kahouadji, L., et al., Multi-domain encoder–decoder neural networks for latent data assimilation in dynamical systems, *Computer Methods in Applied Mechanics and Engineering*, 430, 2024, 117201.
[37] Ronneberger, O., Fischer, P., Brox, T., U-net: Convolutional networks for biomedical image segmentation, *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.
[38] Graves, A., Mohamed, A.R., Hinton, G., Speech recognition with deep recurrent neural networks, *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.
[39] Palangi, H., Deng, L., Shen, Y., et al., Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4), 2016, 694-707.

[40] Hewamalage, H., Bergmeir, C., Bandara, K., Recurrent Neural Networks for Time Series Forecasting: Current status and future directions, *International Journal of Forecasting*, 37(1), 2021, 388-427.
[41] Hochreiter, S., Schmidhuber, J., Long Short-Term Memory, *Neural Computation*, 9(8), 1997, 1735-80.
[42] Yu, Y., Si, X., Hu, C., et al., A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures, *Neural Computation*, 31(7), 2019, 1235-1270.
[43] Hua, Y., Zhao, Z., Li, R., et al., Deep Learning with Long Short-Term Memory for Time Series Prediction, *IEEE Communications Magazine*, 57(6), 2019, 114-119.

ORCID iD

Wenda Yan https://orcid.org/0009-0000-3456-6246
Hongzhi Wang https://orcid.org/0009-0009-9657-0588
Shuisheng Zhang https://orcid.org/0009-0001-4162-7227
Jian Zhang https://orcid.org/0009-0003-8076-6099

How to cite this article: Yan W., et al., Deep Learning Aided Surrogate Model for Real Prediction of Diffusion Process, *J. Appl. Comput. Mech.*, xx(x), 2025, 1–10. https://doi.org/10.22055/jacm.2025.48691.5429