

OPEN

# Reaction diffusion system prediction based on convolutional neural network

Angran Li<sup>1</sup>, Ruijia Chen<sup>2</sup>, Amir Barati Farimani<sup>1\*</sup> & Yongjie Jessica Zhang<sup>1\*</sup>

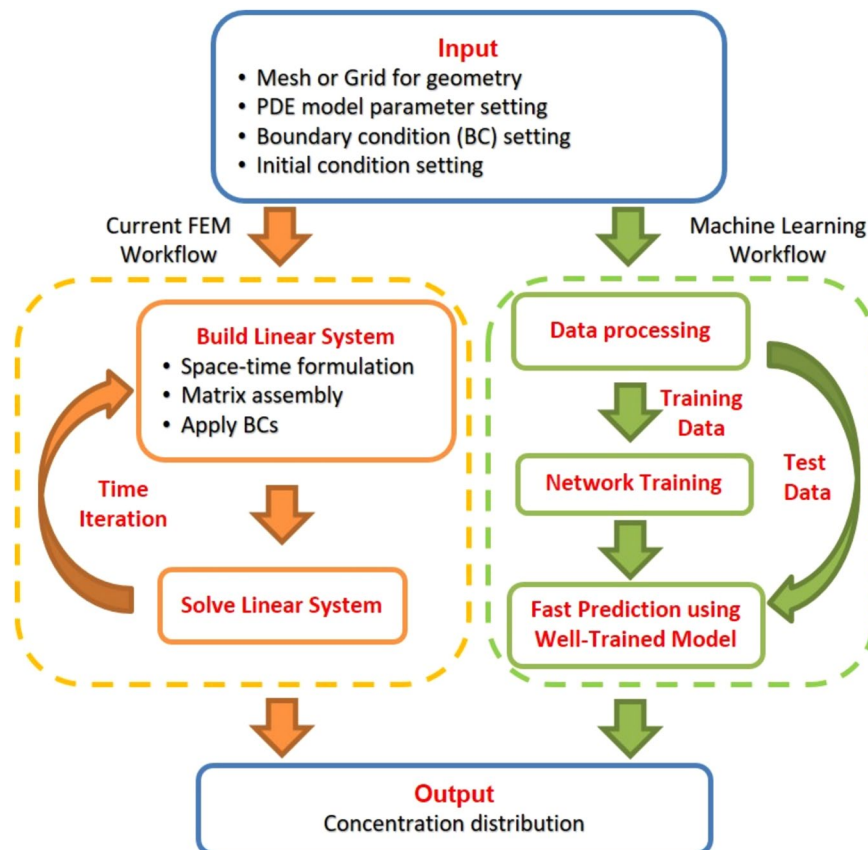
The reaction-diffusion system is naturally used in chemistry to represent substances reacting and diffusing over the spatial domain. Its solution illustrates the underlying process of a chemical reaction and displays diverse spatial patterns of the substances. Numerical methods like finite element method (FEM) are widely used to derive the approximate solution for the reaction-diffusion system. However, these methods require long computation time and huge computation resources when the system becomes complex. In this paper, we study the physics of a two-dimensional one-component reaction-diffusion system by using machine learning. An encoder-decoder based convolutional neural network (CNN) is designed and trained to directly predict the concentration distribution, bypassing the expensive FEM calculation process. Different simulation parameters, boundary conditions, geometry configurations and time are considered as the input features of the proposed learning model. In particular, the trained CNN model manages to learn the time-dependent behaviour of the reaction-diffusion system through the input time feature. Thus, the model is capable of providing concentration prediction at certain time directly with high test accuracy (mean relative error < 3.04%) and 300 times faster than the traditional FEM. Our CNN-based learning model provides a rapid and accurate tool for predicting the concentration distribution of the reaction-diffusion system.

Reaction-diffusion systems have attracted a considerable amount of attention in recent years. They arise naturally in various chemistry models to describe the spatiotemporal concentration change of one or more chemical species which involve both local chemical reaction and diffusion simultaneously. Chemical reactions transform substances from one to another and the diffusion process causes substances to spread out over the spatial domain. The reaction-diffusion system consists of a set of partial differential equations (PDEs) to represent the behaviour of each chemical species individually. The solution of reaction-diffusion systems also presents many interesting phenomena such as spatial patterns, moving fronts or pulses and oscillations<sup>1–4</sup>. Therefore, it has been used to study complex chemical or biological processes including multiple chemical components such as combustion theory<sup>5,6</sup>, calcium dynamics<sup>7</sup> and nerve impulse propagation<sup>8</sup>. The diversity of spatial patterns in reaction-diffusion systems also inspires the biological pattern formation study<sup>9–11</sup> and spatial ecological study<sup>12,13</sup>.

Numerical methods such as Finite Difference Method (FDM) and Finite Element Method (FEM) are widely used to derive the approximate solution of the reaction-diffusion system<sup>14–16</sup>. The current FEM workflow for solving a time-dependent reaction-diffusion system is shown in Fig. 1: (i) generating a mesh corresponding to the geometry; (ii) setting up the finite element model by specifying diffusion and reaction coefficients and the initial and boundary conditions; and (iii) iteratively assembling and solving the linear equations until the expected computational error is reached. The complexity and time-consuming nature of the current FEM workflow makes it hard to rapidly generate output feedback to the input setting. This limits its application to simulate a reaction-diffusion system which occurs in a large spatial domain and requires long time to reach the steady state. Without rapid feedback, it is also inconvenient to synthesize the desired spatial pattern which needs multiple input adjustments of the system.

To address limitations in the current FEM workflow, we use deep learning to study spatial distribution of the reaction-diffusion system and provide quick and accurate prediction given different parameters (Fig. 1). In recent years, deep learning has been proven successful in areas of speech recognition<sup>17,18</sup> and computer vision<sup>19,20</sup>. In particular, different convolutional neural networks (CNN) are designed and lead to extensive application in

<sup>1</sup>Carnegie Mellon University, Department of Mechanical Engineering, Pittsburgh, 15213, United States. <sup>2</sup>Carnegie Mellon University, Department of Electrical and Computer Engineering, Pittsburgh, 15213, United States. \*email: [barati@cmu.edu](mailto:barati@cmu.edu); [jessicaz@andrew.cmu.edu](mailto:jessicaz@andrew.cmu.edu)



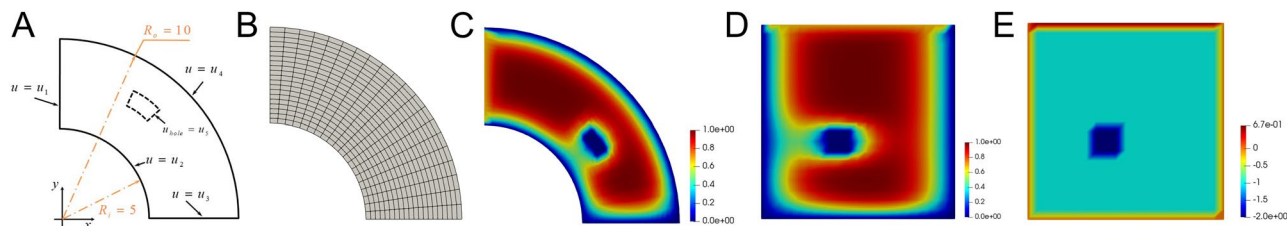
**Figure 1.** The current FEM workflow versus a machine learning-based solution for concentration prediction of a reaction-diffusion system.

image recognition and segmentation<sup>21,22</sup> as well as video classification<sup>23</sup>. The use of encoder-decoder architecture in designing neural networks is a major trend now to establish straightforward mapping between input features and output results. The architecture has been widely used in CNN and achieves great performance in machine translation<sup>24–26</sup>, image semantic segmentation<sup>27</sup> and road scene understanding<sup>28</sup>. The practical success of deep learning in artificial intelligence also inspires new algorithm in solving high-dimensional PDEs<sup>29</sup> and discovering PDEs following the laws of physics<sup>30</sup>. In addition, the combination of machine learning and FEM is also attractive to improve the performance of the conventional FEM workflow<sup>31,32</sup> and develop data-driven solvers<sup>33</sup>. A deep learning model was designed to directly estimate stress distributions of the aorta<sup>34</sup>. The conditional generative adversarial network (*cGAN*) was also applied in a data-driven paradigm for rapid inference, modeling and simulation of transport phenomena<sup>35</sup>.

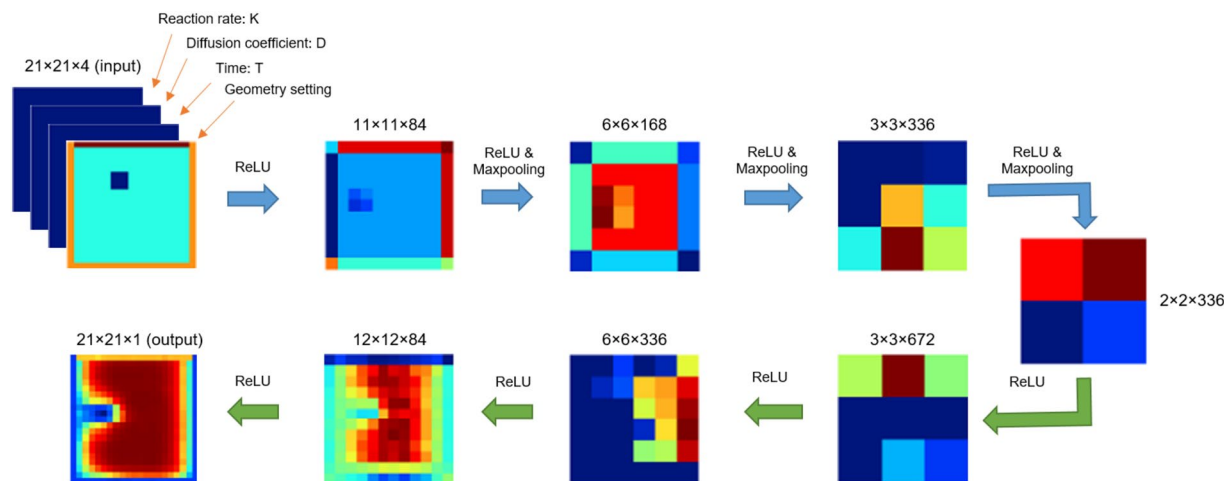
In this paper, we propose to predict the spatial distribution result of a specific reaction-diffusion system using an encoder-decoder based CNN. The reaction-diffusion system is based on Zeldovich equation controlled by several simulation parameters. In our CNN model, the encoder is defined using a four-layer CNN that takes the input of different simulation parameters, boundary conditions, geometry configurations, and output a reduced feature vector. The decoder is another four-layer CNN that takes the feature vector from the encoder and predicts the concentration distribution. The well-trained learning model can predict concentration distributions within 1 second with an average error of 3.04% compared to the traditional FEM. By introducing time as the input feature, our model learns the time-dependent behaviour of the reaction-diffusion system and predicts dynamic concentration distributions within a certain time range. Our model also manages to give accurate prediction even when the distribution is complex or changes rapidly due to complex geometry and simulation settings. Therefore, our CNN-based learning model provides a rapid and accurate tool for predicting the concentration distribution of the reaction-diffusion system.

## Results

**FEM simulation of the reaction-diffusion system.** In this paper, we study a one-component reaction-diffusion system inside a one-fourth arc pipe in 2D domain (Fig. 2A). To consider the effect of different geometry configurations, we introduce a hole with fixed size at different locations of the pipe (black dash box in Fig. 2A). We perform FEM simulations to obtain the spatiotemporal distribution results (Fig. 2C) of different geometries and parameter settings. The FEM results are then processed to become the input dataset of our model (Fig. 2D,E). See the section of Method for more details about the problem setting and data generation.



**Figure 2.** The reaction-diffusion system in our model. (A) The problem setting of the reaction-diffusion system. Black dash box represents a hole with a fixed size. (B) The quadrilateral mesh used in FEM simulations. (C) The concentration result in the physical domain. (D) The concentration result in the parametric domain. (E) The boundary condition visualized in the parametric domain.



**Figure 3.** The convolutional neural network with encoder-decoder architecture. Blue and green arrows represent encoding and decoding, respectively. A representative output for each layer is shown for both encoding and decoding layers.

**Network architecture.** In our model, we design and implement a CNN since it can extract features from our tensor-format input data. We refer to the convolutional encoder-decoder architecture to design the network as shown in Fig. 3. The input four-channel tensor depicting geometry, boundary conditions and parameter settings are processed by a series of convolutional operations in the encoder network to downsample the tensor and obtain a low-dimensional representation. Then, the decoder network reverses the operations in the encoder by a series of deconvolutional operations to upsample the reduced representation and generate the predicted solution. In Fig. 3, the top row (encoder network) shows how the optimized features are extracted through a series of convolutional layers. For instance, in the  $11 \times 11$  feature map, a region looking like the hole is distinguishable from the other interior region and the four boundary edges show different colors from the interior region indicating different boundary values. After one convolutional operation, the resulting  $6 \times 6$  feature map becomes abstract but the hole feature is still noticeable. Therefore, it is believed that optimal information is continuously extracted layer by layer so that the geometry and parameter settings are transformed to the distribution results properly. The bottom row (decoder network) in Fig. 3 shows how the extracted features are reconstructed in the corresponding distribution results.

To select the appropriate activation function in the network, we test networks for three widely used activation functions: *ReLU*<sup>36</sup>, *PReLU*<sup>37</sup> and *ELU*<sup>38</sup>. The test results illustrate that all three activation functions can achieve similar accuracy while ReLU needs the shortest time in training (see Supplemental Information; Fig. S1). Therefore, we employ ReLU as the activation function in both the encoder and the decoder.

Moreover, *maxpooling* is used in the encoder process to drop unnecessary features to prevent overfitting and keep transition and rotation invariance of features. During the learning process, we use the Mean Square Error (MSE) as the loss function to adjust the weights of all layers,

$$MSE = \frac{1}{M \times N} \sum_{j=1}^M \sum_{i=1}^N (u_{i,j}^P - u_{i,j}^G)^2, \quad (1)$$

where  $u_{i,j}^P$  is the  $i^{\text{th}}$  entity of the  $j^{\text{th}}$  predicted concentration matrix, while  $u_{i,j}^G$  is the  $i^{\text{th}}$  entity of the  $j^{\text{th}}$  ground truth concentration matrix. In our training process, we set the learning rate to be 0.0001. We also monitor the loss function value in Equation (1) for each training epoch to make sure the training converges.

**Model evaluation.** We use the mean absolute error (MAE) and the mean relative error (MRE) as two performance metrics to evaluate the accuracy of the predicted concentration distributions. For each predicted result, the MAE is defined by

$$MAE = \sqrt{\frac{1}{N} \sum_{i=1}^N (u_i^P - u_i^G)^2}, \quad (2)$$

where  $N$  denotes the number of entities in the output matrix,  $u_i^P$  and  $u_i^G$  denote the  $i^{\text{th}}$  entity in the prediction matrix and the ground truth matrix, respectively. For each predicted result, the MRE is defined by

$$MRE = \frac{MAE}{\max \|u^G\| - \min \|u^G\|} \times 100\%, \quad (3)$$

where  $\max \|u^G\|$  and  $\min \|u^G\|$  denote the maximum and minimum values from the ground truth matrix, respectively.

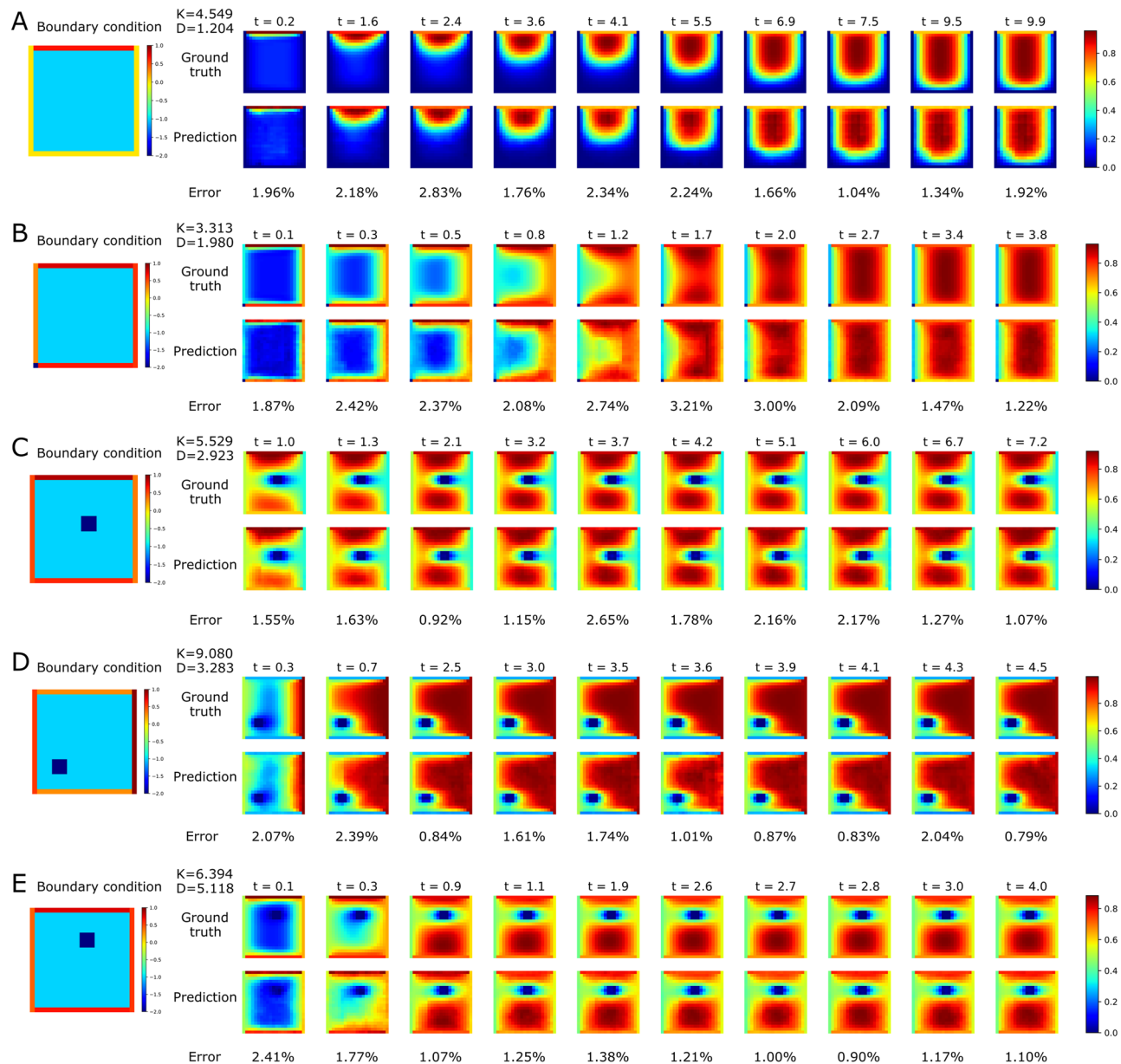
We first study the effect of the original state and main parameters  $D$ ,  $K$  and time  $t$  on the dynamic concentration pattern of the reaction-diffusion system. The dataset consists of 2,500 matrices, each of which was 21 by 21. After establishing the architecture of our neural network, we train the network by randomly selecting 75% samples as the training data. Our models are trained for 100 epochs and the MSE loss converges to 0.0035 at the end of training. Then, we evaluate the performance of the model using the rest 25% samples as the test dataset. To demonstrate the performance of our model, we select several prediction results from the test dataset that display a variety of distribution profiles and compare with their corresponding ground truth in Fig. 4. By comparing the predicted results with the corresponding boundary conditions, we observe that our model captures the distribution feature at boundary (high concentration at top edge in Fig. 4A and low concentration at left edge in Fig. 4B) and gives the prediction following the boundary condition. We also find in Fig. 4C–E that the fixed-size hole is accurately restored in the prediction results. The comparison of time-serial results in each example of Fig. 4 also shows that the dynamic change of distribution can be predicted with good accuracy. Even when the distribution changes dramatically in a short time ( $t = 0.3$  to  $0.7$  in Fig. 4D or  $t = 0.1$  to  $0.9$  in Fig. 4E), the model is still capable of providing convincing prediction with around 1% accuracy decrease.

In addition, we study the influence of the  $K/D$  value, which is the ratio between the reaction and diffusion process, on the concentration distribution results and the training of our model. We first plot the error vs  $K/D$  on test data (see Supplemental Information; Fig. S2) and find the  $K/D$  value of most test samples falls in the range of  $[0, 500]$ . Then, we compare several time-serial samples with the same boundary condition to investigate the effect of the  $K/D$  value. Given the same boundary condition as in Fig. 4A, the concentration distribution varies with the  $K/D$  value (see Supplemental Information; Fig. S3). When the  $K/D$  ratio increases, the reaction becomes dominant in the reaction-diffusion system and the time needed to reach steady state drops quickly. The large  $K/D$  value leads to the wide spread of the substance and further decreases the accuracy of our model, as shown in Fig. S3A,B. The hole inside the geometry also brings difficulty to the prediction by making the distribution more complex. To study the influence of hole location on the prediction accuracy, we plot the error vs the location of hole plot on test data (see Supplemental Information; Fig. S4). The plot shows that the test sample has poor performance when the hole locates closer to the domain boundary. We also find that the  $K/D$  values are similar in Fig. S3C,D, but different locations of the hole contribute to two completely different distribution profiles in which the wide-spread distribution shows lower accuracy. Therefore, our model can be further improved to handle the complexity of distribution profiles due to different input simulation parameters and geometries.

We then evaluate the MRE on each sample of the test dataset and the trained model can achieve an average error of 3.04% with the maximum error around 30% (see Supplemental Information; Fig. S5). In addition, the error of most samples are located near the mean value line. By using the defined MRE, we also identify five worst prediction results in the test dataset (see Supplemental Information; Fig. S6). We observe that all the worst predictions (with an average error of 29.94%) contain a hole in the geometry, which indicates that our model tends to have lower accuracy in the system with complex distribution because of the hole.

We also compare the accuracy and computational performance of our algorithm with our FEM solver on a PC with a 2.93 GHz quad core CPU and 16 GB RAM. Since the analytical solution for Equation (5) is unknown, we run our FEM solver in a fine  $81 \times 81$  grid to obtain the baseline solution for accuracy evaluation. To compare the computation time, we run the FEM simulation and CNN prediction to solve the reaction-diffusion equation with the same input configurations for 1,000 time steps. The comparison results of two FEM models and two CNN models are summarized in Table 1. We can observe that the simulation time of the full CNN model (0.155s) is two orders of magnitude faster than the full FEM model (46s) while the error only increases by 1.8%. We then coarsen the input mesh of the full FEM model and obtain a simplified FEM model with similar error as the full CNN model (7.56% vs 7.13%), but the full CNN model is still much faster than the simplified FEM model (16s). We also compare our full CNN model with a simplified CNN model to show the computational behaviour of our model. The simplified CNN model handles the dataset created by FEM simulations on a coarse  $11 \times 11$  grid (see Supplemental Information; Fig. S7). We find that the error increases to 11.37% but the simulation time does not improve much, which suggests that the CNN model is more suitable for large size problems to achieve better performance improvement.

Note that deep learning is not the only way to predict the solution of PDEs, classical fitting methods can also provide a good approximation. Here, we compare our method with the solution estimated by Fourier series. We randomly pick three sets of time-serial data and each set has the same  $D$  and  $K$  values. The approximate solution  $u_{\text{approx}}$  of the Fourier series form is given by



**Figure 4.** The concentration distribution comparison with five different simulation configurations. (A,B) The same geometry without hole; and (C–E) geometries with hole at three different locations. The examples are selected in test dataset to show the prediction performance for different distribution profiles. For each configuration, the ground truth results, predicted results and errors are shown from top to bottom.

Model	Training time	Simulation time	Mean error computed using the baseline solution
Full FEM model in Fig. 2	0	46s	5.74%
Full CNN model in Fig. 3	9.3 hrs	0.155s	7.56%
Simplified FEM model (15 × 15 grid)	0	16s	7.13%
Simplified CNN model in Fig. S7	7.6 hrs	0.112s	11.37%

**Table 1.** Comparison between CNN models and FEM models.

$$u_{approx} = a_0 + \sum_{m=1}^M \sum_{n=1}^N (a_m \cos(m\omega_x x) + b_m \sin(m\omega_x x))(c_n \cos(n\omega_y y) + d_n \sin(n\omega_y y)), \tag{4}$$

where  $M, N$  are set to be 4 and  $a_0, a_m, b_m, c_n, d_n, \omega_x, \omega_y$  are to-be-determined parameters. We adopt the least-square method to fit the data at each time step and compare with the prediction results from our model (see Supplemental Information; Fig. S8). We find that deep learning prediction has higher accuracy than the Fourier series approximation. Though Fourier series can estimate the location of the hole (Fig. S8C), it fails to capture the size of the hole ( $t = 0.4$  in Fig. S8B) and sometimes it even fails to capture the existence of the hole (Fig. S8A). The possible reason is that the classical fitting method needs a good approximation function to obtain high accuracy and Fourier series may not be a good choice to fit the data. Whereas the structure of neural network can be easily modified to fit for more complex distribution and yield better prediction.

## Discussion

In this paper, we develop a CNN model to study the specific one-component reaction-diffusion system based on the data from the FEM simulation. The network adopts the encoder-decoder architecture with five feature parameters sorted into different input channels of CNN, namely the diffusion coefficient, the reaction rate, time, boundary conditions and the geometry configuration. Given input parameters, our model can provide comparatively accurate and two orders of magnitude faster spatiotemporal distribution prediction compared to the FEM simulation results. The crucial features like the location of the hole and boundary conditions are accurately captured by learning the meaning of different labels from input geometry information. More surprisingly, the CNN model can even predict accurate distribution result at certain time without iterative process, which we believe that the embedded time channel instructs the model to learn the time-dependent behaviour of the reaction-diffusion system. Overall, our well-trained CNN model can provide efficient spatiotemporal distribution prediction of the specific reaction-diffusion system. Compared to FEM simulation, Our deep learning model avoids the expensive solving and iteration processes and generates distribution result directly. When designing a reaction-diffusion process with desired distribution result, the rapid feedback from the design tool is quite important and our model is undoubtedly competent for this task. In addition, our method can also be extended to study more complex reaction-diffusion systems using dataset from experiment observation other than FEM simulation. In this case, more parameters should be introduced in the network architecture to fit for the increasing complexity of the reaction-diffusion system. Suppose the model is trained on the experimental data, the model evaluation may be difficult due to the limited amount of data with ground truth. One common way is to perform cross validation on training data with the limitation on completely ignoring the testing data. Other approaches such as reverse testing framework<sup>39</sup> and intersection-validation<sup>40</sup> are suitable in this situation to perform model evaluation utilizing the testing data.

In summary, our model shows multiple novel aspects within both the representation of the data and the architecture of the network used. The introduction of time as a channel for inference, the representation of solid boundaries and geometries into CNN, the conditioning of the diffusion constant or any other material properties and usage of that in combination with other parameters such as time are some of new representation aspects of the project. In addition, using encoder-decoder architecture for time-dependent prediction with multiple channels of the data and their interdependencies is another architectural advance we have performed in this article. Compared to most finite element models or any engineering modeling, we are in essence approximating functions and since neural networks are universal function approximators, all models can be considered as neural networks. However, the convolution operation and the robust feature detection along with the architecture of U-Net which is an encoder-decoder network made it possible to create a surrogate model using neural networks which is faster than conventional FEM models. Our work also has its limitation which we will address in our future work. Based on results from our model, we find that the learning performance is influenced by the  $K/D$  ratio and the geometry configuration. Our model becomes unstable when more features are included, which suggests that we need to increase the complexity of the network to improve the performance. Regarding to the geometry configuration, we only include one hole with fixed rectangle shape but different locations in our dataset and holes with more complex shape should also be included in our model. In addition, we only consider a simple one-fourth arc pipe geometry in 2D domain and we will include more complex geometry to improve the applicability of our model. Despite these limitations, our model directly predicts the dynamic concentration distribution of the reaction-diffusion system and provides a powerful tool for further study in this field.

## Methods

**Problem setting.** To specify the problem, we assume a one-component reaction-diffusion system inside a one-fourth arc pipe in 2D domain (Fig. 2A). This geometry is selected to derive asymmetric concentration distributions and introduce the learning of geometry features to the model. The arc pipe has the inner radius  $R_i = 5$  and the outer radius  $R_o = 10$ . The concentration change of one chemical substance in space and time can be described using the following reaction-diffusion equation,

$$\frac{\partial u}{\partial t} = D\Delta u + R(u), \quad (5)$$

where  $D$  is the diffusion coefficient,  $u = u(x, y, t)$  is the concentration of the substance,  $R(u)$  is the reaction term. Here we select  $R(u) = Ku^2(1 - u)$ , where  $K$  is the reaction rate and this equation is also referred to as the Zeldovich equation<sup>41</sup> used in the combustion theory.

The initial condition is set to be  $u(x, y, 0) = 0$ . We apply the Dirichlet boundary condition at four edges of the pipe (Fig. 2A),

$$u(x, y, t) = \begin{cases} u_1, & x = 0, \\ u_2, & r = R_i, \\ u_3, & y = 0, \\ u_4, & r = R_o, \end{cases} \quad (6)$$

where  $r = \sqrt{x^2 + y^2}$  is the distance between the point  $(x, y)$  to the center of the arc pipe.

To consider the effect of different geometry configurations, we introduce a hole with fixed size at different locations of the pipe (black dash box in Fig. 2A). At the boundary of the hole, we also apply the Dirichlet boundary condition,

$$u_{hole} = u_5, \quad (7)$$

and we set  $u_5 = 0$  to simplify the problem.

**Data generation.** The dataset of reaction-diffusion system can be collected from experiments or computer simulation results. For fast and flexible dataset generation, we develop a FEM-based solver to solve Equation (5) and obtain dynamic concentration results for the system. We create a  $21 \times 21$  quadrilateral mesh (Fig. 2B) for the one-fourth arc pipe as the input of the FEM solver and one sample result is shown in Fig. 2C. Four types of simulation parameters are considered in our model, namely the diffusion coefficient  $D$ , the reaction rate  $K$ , time  $t$  and boundary values  $u_1 \sim u_4$ . To include the effect of geometry on the concentration result, we put a hole consisting of  $2 \times 2$  elements to 20 different locations inside the pipe and obtain 21 different geometries including one geometry without hole. For each geometry, we generate 500 different sets of  $D$ ,  $K$  and  $u_1 \sim u_4$  and solve the equation for 1,000 time steps (time step size  $\delta t = 0.01$  s).  $D$  and  $K$  are randomly selected in range  $[0, 10]$  and  $u_1 \sim u_4$  are randomly selected in range  $[0, 1]$ . The result of every 10 steps is extracted and we finally create the dataset containing 1,050,000 samples. Then we perform data processing to fit the dataset into our neural network with tensor data format.

For each FEM simulation result, we directly extract the nodal concentration value from the mesh as the input data instead of using the concentration color map. Since the single pipe geometry is topologically equivalent to a rectangle, we map the concentration distribution result from the physical domain (Fig. 2A) to a unit square in the parametric domain (Fig. 2D). In this case, the topological mapping between physical space coordinates  $(x, y)$  and parametric space coordinates  $(\xi, \eta)$  is defined as

$$\begin{cases} \xi = (r - R_i)/(R_o - R_i), \\ \eta = 2\theta/\pi, \end{cases}$$

where  $r = \sqrt{x^2 + y^2}$  is the distance between the point  $(x, y)$  to the center of the arc pipe and  $\theta = \arctan(y/x)$  is the angle between vector  $(x, y)$  and  $x$ -axis.

The nodal concentration values are then stored in a  $21 \times 21$  matrix as one channel of the data. In particular, the nodal concentration values are set to 0 inside the hole region. Regarding to the input boundary and initial condition information, we use a matrix with the same size and store the boundary values in it with the hole region setting to be  $-2$  and the region exterior to the hole setting to be  $-1$ , shown in Fig. 2E. For the other three feature parameters  $D$ ,  $K$  and  $t$  (scalars), we create three  $21 \times 21$  matrices to store each of them. The matrix representation of the feature parameters can be easily fitted in the convolutional operation during the encoding process. The usage of matrix representation also accounts for the possible spatial variant feature parameters. In particular, if  $D$  and  $K$  are invariant scalars, they can also be stored into two  $2 \times 2$  matrices and concatenated with the hidden layer obtained at the end of the encoding process (Fig. 3).

In summary, we generate four input matrices including the feature information (*i.e.* different geometries and parameter configurations) and one output matrix including the dynamic concentration result.

### Code and data availability

The source code for our neural network is available for download from a public software repository located at <https://github.com/truthlive/RDCNN>. All data generated during this study can be reconstructed by running the source code.

Received: 3 November 2019; Accepted: 10 February 2020;

Published online: 03 March 2020

### References

- Rinzel, J. & Terman, D. Propagation phenomena in a bistable reaction-diffusion system. *SIAM J. on Appl. Math.* **42**, 1111–1137 (1982).
- Turing, A. M. The chemical basis of morphogenesis. *Bull. Math. Biol.* **52**, 153–197 (1990).
- Szalai, I. & De Kepper, P. Turing patterns, spatial bistability, and front instabilities in a reaction-diffusion system. *The J. Phys. Chem. A* **108**, 5315–5321 (2004).
- Riaz, S. S. & Ray, D. S. Diffusion and mobility driven instabilities in a reaction-diffusion system: a review. *Indian J. Phys.* **81**, 1177–1204 (2007).
- Zeldovich, I. A. B., Barenblatt, G. I., Librovich, V. B. & Makhviladze, G. M. *Mathematical Theory of Combustion and Explosions* (Consultants Bureau, New York, NY, 1985).
- Poinsot, T. & Veynante, D. *Theoretical and Numerical Combustion* (RT Edwards, Inc., 2005).
- Means, S. *et al.* Reaction diffusion modeling of calcium dynamics with realistic ER geometry. *Biophys. J.* **91**, 537–557 (2006).

8. Manakova, N. A. & Gavrilova, O. V. Numerical study of the process of optimizing the propagation of a nerve impulse in a membrane for a three-component model. In *International Russian Automation Conference (RusAutoCon)*, 1–5 (IEEE, 2018).
9. Cross, M. C. & Hohenberg, P. C. Pattern formation outside of equilibrium. *Rev. Mod. Phys.* **65**, 851 (1993).
10. Kondo, S. & Asai, R. A reaction–diffusion wave on the skin of the marine angelfish pomacanthus. *Nature* **376**, 765 (1995).
11. Kondo, S. & Miura, T. Reaction-diffusion model as a framework for understanding biological pattern formation. *Sci.* **329**, 1616–1620 (2010).
12. Cantrell, R. S. & Cosner, C. *Spatial Ecology via Reaction-Diffusion Equations* (J. Wiley, 2003).
13. Tilman, D. & Kareiva, P. *Spatial Ecology: The Role of Space in Population Dynamics and Interspecific Interactions (MPB-30)*, vol. 30 (Princeton University Press, 2018).
14. Ramos, J. I. A review of some numerical methods for reaction-diffusion equations. *Math. Comput. Simul.* **25**, 538–548 (1983).
15. Roos, H.-G., Stynes, M. & Tobiska, L. *Robust Numerical Methods for Singularly Perturbed Differential Equations: Convection-Diffusion-Reaction and Flow Problems*, vol. 24 (Springer Science & Business Media, 2008).
16. Iida, M., Ninomiya, H. & Yamamoto, H. A review on reaction-diffusion approximation. *J. Elliptic Parabol. Equations* **4**, 565–600 (2018).
17. Hinton, G. *et al.* Deep neural networks for acoustic modeling in speech recognition *IEEE Signal Process. Mag.* **29** (2012).
18. Abdel-Hamid, O. *et al.* Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, Lang. Process.* **22**, 1533–1545 (2014).
19. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097–1105 (2012).
20. Witten, I. H., Frank, E., Hall, M. A. & Pal, C. J. *Data Mining: Practical Machine Learning Tools and Techniques* (Morgan Kaufmann, 2016).
21. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
22. Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440 (2015).
23. Karpathy, A. *et al.* Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1725–1732 (2014).
24. Cho, K. *et al.* Learning phrase representations using RNN encoder-decoder for statistical machine translation *arXiv preprint arXiv:1406.1078* (2014).
25. Cho, K., Van Merriënboer, B., Bahdanau, D. & Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches *arXiv preprint arXiv:1409.1259* (2014).
26. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
27. Badrinarayanan, V., Kendall, A. & Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561* (2015).
28. Yasrab, R. ECRU: An encoder-decoder based convolution neural network (CNN) for road-scene understanding. *J. Imaging* **4**, 116 (2018).
29. Han, J., Jentzen, A. & Weinan, E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci.* **115**, 8505–8510 (2018).
30. Raissi, M., Perdikaris, P. & Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **378**, 686–707 (2019).
31. Oishi, A. & Yagawa, G. Computational mechanics enhanced by deep learning. *Computer Methods in Applied Mechanics and Engineering* **327**, 327–351 (2017).
32. Capuano, G. & Rimoli, J. J. Smart finite elements: A novel machine learning application. *Comput. Methods Appl. Mech. Eng.* **345**, 363–381 (2019).
33. Kirchdoerfer, T. & Ortiz, M. Data-driven computational mechanics. *Comput. Methods Appl. Mech. Eng.* **304**, 81–101 (2016).
34. Liang, L. & Liu, M. A deep learning approach to estimate stress distribution: A fast and accurate surrogate of finite-element analysis. *J. The Royal Soc. Interface* **15**, 20170844 (2018).
35. Farimani, A. B., Gomes, J. & Pande, V. S. Deep learning the physics of transport phenomena *arXiv preprint arXiv:1709.02432* (2017).
36. Glorot, X., Bordes, A. & Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 315–323 (2011).
37. He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, 1026–1034 (2015).
38. Clevert, D.-A., Unterthiner, T. & Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs) *arXiv preprint arXiv:1511.07289* (2015).
39. Bhaskaruni, D., Moss, F. P. & Lan, C. Estimating prediction qualities without ground truth: A revisit of the reverse testing framework. In *IEEE 24th International Conference on Pattern Recognition (ICPR)*, 49–54 (2018).
40. Viinikka, J., Eggeling, R. & Koivisto, M. Intersection-validation: A method for evaluating structure learning without ground truth. In Storkey, A. & Perez-Cruz, F. (eds.) *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, vol. 84 of *Proceedings of Machine Learning Research*, 1570–1578 (2018).
41. Gilding, B. H. & Kersner, R. *Travelling Waves in Nonlinear Diffusion-Convection Reaction*, vol. 60 (Birkhäuser, 2012).

## Acknowledgements

The authors acknowledge support of the NSF grant CBET-1804929 and a collaborative research seed grant from Department of Mechanical Engineering, Carnegie Mellon University.

## Author contributions

A.L. and R.C. designed the model and carried out the calculations and analysis with the instruction from A.B.F. and Y.J.Z. All authors contributed to the writing, discussions and revisions of the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** is available for this paper at <https://doi.org/10.1038/s41598-020-60853-2>.

**Correspondence** and requests for materials should be addressed to A.B.F. or Y.J.Z.



**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020