Physics-based Deep Spatio-temporal Metamodeling for Cardiac Electrical Conduction Simulation

Hao Yan, Xinyu Zhao, Zhiyong Hu, Dongping Du

Abstract—**Modeling and simulation have been widely used in both cardiac research and clinical study to investigate cardiac disease mechanism and develop new treatment design. Electrical conduction among cardiac tissue is commonly modeled with a partial differential equation, i.e., reaction-diffusion equation, where the reaction term describes cellular excitation and diffusion term describes electrical propagation. Cellular excitation can be modeled by either detailed human cellular models or simplified models such as the FitzHugh-Nagumo model; electrical propagation can be simulated using either bio-domain or mono-domain tissue model. However, existing cardiac models have a great level of complexity, and the simulation is often time-consuming. This paper develops a new spatiotemporal model as a surrogate model of the time-consuming cardiac model. Specifically, we propose to investigate the auto-regressive convolutional neural network (AR-CNN) and convolutional long short-term memory (Conv-LSTM) to model the spatial and temporal structure for the metamodeling. Model predictions are compared to the one-dimensional simulation data to validate the prediction accuracy. The metamodel can accurately capture the properties of the individual cardiac cell, as well as the electrical wave morphology in cardiac fiber at different simulation scenarios, which demonstrates its superior performance in modeling and the long-term prediction.**

I. Introduction

Spatio-temporal meta-modeling for the nonlinear partial differential equations plays an important role in different applications. Typical practice is to develop a surrogate model to replace the time-consuming physical model for efficient modeling and simulation. Spatio-temporal meta-modeling has been widely applied in various applications [1]. However, to the best of our knowledge, few studies have been done to develop metamodels for cardiac modeling and simulation.

Spatio-temporal meta-modeling of cardiac electrical dynamics is challenging due to the complexity and the multiscale characteristic of a cardiac model. The mathematical model of heart integrates cellular models with tissue model to describe the electrical propagation among cardiac muscles. Cardiac cell model describes the cell membrane potential as a function of time, which is commonly defined as action potential (AP) [2], by an ordinary differential equation. The spatiotemporal propagation of electrical waves in cardiac tissue is modeled by a reaction-diffusion equation where the reaction term describes the cellular excitation and the diffusion term represents cell-to-cell interactions. There are generally two groups of cardiac cell models, i.e., detailed models and simplified models. In detailed models, AP is formulated as a function of ion channel gating which is further modeled by a set of ordinary differential equations and supporting equations. Detailed cardiac models often include hundreds of equations, and the simulation can be very time-consuming. To reduce the computational cost, simplified models such as FitzHugh–Nagumo (FHN) model are developed [3], which consists of two coupled equations to capture cardiac depolarization and repolarization. Although the FHN model shows great advantages in reducing computational time, when it goes to higher organizational scale, e.g., tissue and organ scales, the simulation is still very time-demanding. Furthermore, the FHN equation cannot be used directly in many cases since it involves some unobservable variables (e.g. gating variables), which cannot be measured directly in the experiments.

Metamodeling has been a popular approach that helps reduce model complexity, estimate the unobservable variables, and overcome computational challenges [4]. Metamodels are constructed in various studies to learn the hidden dynamics in spatio-temporal processes. It is worth mentioning that the cardiac electrical propagation is a spatio-temporal process with cellular AP changing in a nonlinear fashion in the temporal domain and electrical waves propagating in the spatial domain. This spatio-temporal process is regulated by a nonlinear partial differential equation (PDE). The objective of this study is to develop a computationally efficient metamodel to replace the time-consuming simulation model of cardiac electrical propagation. This can be realized by learning the hidden dynamics of the spatio-temporal process from simulation data generated by the cardiac models.

Studies have been done in the literature to learn covariance structures of spatio-temporal data driven by PDEs. Gaussian Process (GP) model has been a popular choice to extract information from high-dimensional data [5]. For example, Raissi et al developed a framework based on GP to represent the underlying laws of physics expressed by time-dependent and nonlinear PDEs from experimental data [6]. Sigrist et al. defined a GP through a stochastic partial differential equation to cope with large spatio-temporal data set [7]. Alvarez et al. developed a hybrid approach using GP and differential equations to combine data-driven modeling with a physical model of the system [8]. Despite the fact that previous studies have contributed a significant amount of knowledge in the PDE-driven metamodeling, existing techniques cannot be directly applied to model the spatio-temporal propagation of electrical waves in the cardiac system due to the following two

*Research supported by National Science Foundation.

Hao Yan is with the Arizona State University, Tempe, AZ 85281 USA (corresponding author) phone: (480) 727-0556; e-mail: haoyan@asu.edu.

Xinyu Zhao is with the Arizona State University, Tempe, AZ 85281 USA e-mail: xzhao119@asu.edu.

Zhiyong Hu, and Dongping Du are with Texas Tech University, Lubbock, TX 79409, USA, email: dongping.du@ttu.edu

reasons: 1) most of the current techniques are limited to linear or stationary process, but many PDE equations such as electrical propagation in heart tissue can be nonstationary and nonlinear; 2) many of algorithms such as GP models are not computationally efficient, which cannot be used for real-time prediction in meta-modeling.

Recently, deep neural networks such as convolutional neural networks [9] and recurrent neural networks [10] are developed to model the complex nonlinear spatial structures and temporal structures, respectively. Furthermore, the use of GPU computing is also able to greatly speed up the prediction time, which can be used for real-time purposes. Specifically, physics-driven deep learning methodology has been proposed in the literature, which includes PDE-net [11], Physics-informed deep learning [12], and Deep Galerkin Method [13]. However, these methods either assume the PDE network structure is known or it is derived from a pure-data driven approach, which causes limited long-term prediction accuracy. Moreover, their use in modeling of the cardiac electrical propagation simulation has not been studied yet. Therefore, we propose to develop neural network model-based meta-modeling techniques for cardiac modeling and simulation.

The rest of the paper is organized as follows. Section II introduces the methodology used for meta-modeling, and Section III provides a case study to show the performance of the proposed method, which is followed by Section IV Conclusion.

II. Methodology

In this section, the simplified model of cardiac electrical propagation will be firstly introduced in Section II. A, which is used to generate spatio-temporal simulation data. Further, the metamodel of the cardiac model is presented in section II. B. We propose to use two neural network architectures, i.e., regularized autoregressive model convolutional neural network (AR-CNN), and regularized convolutional long short-term memory (Conv-LSTM). In addition, optimization procedures are given in section C.

A. Cardiac Model

In this study, FHN model is used to calculate the AP of individual cardiac cell, and mono-domain tissue model is adopted to simulate the electrical wave propagation on a one-dimensional (1D) cell string [3]. Define the AP of any cardiac cell on the 1D cell string as μ, then the change of μ in the spatial and temporal domain follows a PDE defined as follows.

$$\frac{d\mu}{dt} = c_1\mu(\mu - a)(1 - \mu) - c_2\mu v + I_{st} + D\frac{\partial^2\mu}{\partial s^2} \quad (1)$$

$$\frac{dv}{dt} = b(\mu - dv) \quad (2)$$

where t is time, μ is a fast variable that describes the membrane potential, i.e., action potential (AP), v is a slow variable that corresponds to the gating variable, a, b, c_1, c_2, and d are model parameters, which are set to 0.13; 0.013, 0.26, 0.1, 1.0, respectively. D is the conduction velocity which is chosen as 1.0, and s is the spatial location of a cell in the 1D cell string. I_{st} is a stimulation current that triggers cell excitation and electrical propagations. In this study, spatial location and frequency of I_{st} are set to different values to generate stimulation data. For more details about the simulation setup, please refer to the simulation section.

B. Proposed Model

One of the challenges of using (1) and (2) directly is that v is not observable. It is often not possible to directly use (1) and (2) for model prediction. In this section, we will develop the spatio-temporal autoregressive model for the complex spatio-temporal dynamics to simulate the partial differential equation (PDE) that governs the system. We begin with the spatio-temporal function $\mu(s,t)$ and assume that this function follows certain spatio-temporal dynamic propagation as $\frac{\partial\mu}{\partial t} = f\left(\mu, \frac{\partial\mu}{\partial s}, \frac{\partial^2\mu}{\partial s^2}\right) + I_{st}$, where I_{st} is the external stimulation. Here, we assume that the external stimulation I affects the $\mu(s,t)$ in a linear fashion. This information is also reflected in (1) from our physics models.

In reality, we only have discrete measurements at time $t = 1, \cdots, T$. We use $\mu_t(s)$ to represent the profile measurement $\mu(s,t)$ at time t. To discretize the PDE on the time dimension, we will use the Euler's method $\mu_{t+1}(s) = \mu_t(s) + f\left(\mu_t(s), \frac{\partial\mu_t(s)}{\partial s}, \frac{\partial^2\mu_t(s)}{\partial s^2}\right) + I_t(s)$ [14]. To discretize the PDE on the spatial dimension, we use μ_t to represent the vector $\left(\mu(1,t), \cdots, \mu(S,t)\right)^T$. We assume that the observed data $y_t = \mu_t + e_t$, where $e_t \sim N(0, \sigma^2)$ is the noise. Since $\frac{\partial\mu_t}{\partial x} \approx \mu_t(s+1) - \mu_t(s)$ and $\frac{\partial^2\mu_t}{\partial x^2} \approx \mu_t(s+2) - 2\mu_t(s+1) + \mu_t(s)$. In another word, both $\frac{\partial\mu_t}{\partial s}$ and $\frac{\partial^2\mu_t}{\partial s^2}$ can be represented by the convolutional operations with $k_1 = [1, -1]$ and $k_2 = [1, -2, 1]$. It is worth noting that the direct estimation on μ_{t+1} based on the Euler's equation is not possible due to the unknown dynamic propagation $f(\cdot)$. This inspires us to emulate the μ_t directly from the data by using nonlinear parametric models with convolutional operators g to represent the complex nonlinear spatio-temporal system dynamic as (1):

$$\mu_{t+1} = \mu_t + g(y_1, \cdots, y_t; \theta), y_{t+1} = \mu_{t+1} + I_{t+1} + e_t, \quad (1)$$

where θ are the model parameters. Here $g(\cdot)$ links the difference $\mu_{t+1} - \mu_t$ from the original data $y_1, \cdots, y_t$. To estimate θ, we propose a penalized regression model to minimize $l(\theta)$ as follows:

$$l(\theta) = \sum_{t=1}^{T}(\| y_{t+1} - \mu_{t+1} - I_{t+1} \|^2 + \lambda\mu_{t+1}^T R\mu_{t+1}) \quad (2)$$

where $\|\cdot\|$ is the L_2 norm operator, and λ are tuning parameters controlling the spatio-temporal smoothness and sparsity of the anomaly. The Matrix R is the regularization matrix that controls the smoothness of the mean function μ_t. For example, one popular choice for R is that $R = D^T D$, where D is the second order differential operator as $D = \begin{bmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & & \\ & & 1 & -2 & 1 \end{bmatrix}$.

To optimize the loss function, we propose to use the backpropagation architecture to optimize the parameters $\theta^{(k)}$ in each iteration. For more details about how to derive the gradient is discussed in the next subsection. Finally, stochastic gradient techniques can be used to update the model parameter

θ [15] based on a mini-batch of samples in the k^{th} iteration as follows:

$$\theta^{(k+1)} = \theta^{(k)} - \frac{1}{B}\sum_{i_k=1}^{B} \frac{\partial l\left(\theta; y_t^{(i_k)}\right)}{\partial \theta}\Big|_{\theta^{(k)}}. \quad (3)$$

C. Model Architecture and Optimization Techniques

In this section, we will discuss two different neural architectures that are inspired by the common PDE function architectures.

1) Convolutional Neural Network-based Autoregressive Model (AR-CNN)

Model Architecture: As mentioned above, the PDEs can be normally emulated by the CNNs. Here, we propose to learn two CNNs g_1, g_2 to emulate μ_t as $\mu_{t+1} = \mu_t + g_1(y_t; \theta)$ and $\mu_t = g_2(y_t; \theta)$. Here, $g_2(\cdot)$ is a denoising CNN, which aims to remove the noise from the measurement data y_t. $g_1(\cdot)$ is a prediction CNN, which aims to learn the differential dynamic from the spatio-temporal data. This architecture is motivated by the Euler's equation, as introduced in Section II. B [14].

For the CNN architecture, we propose to apply the same padding after each convolutional layer, where the size of the output is kept the same after each spatial convolutional layer. To simulate the PDE boundary conditions, we propose to use the replication padding, where the input tensor is padded using replication of the input boundary.

Inference and Optimization Techniques: Here, we propose to learn the $g_1(\cdot)$ and $g_2(\cdot)$ from the following loss functions

$$l(\theta) = \sum_{t=1}^{T} \| y_t - g_2(y_t; \theta) - I_t \|^2 + \sum_{t=1}^{T} \| y_{t+1} - g_2(y_t; \theta) - g_1(y_t; \theta) - I_{t+1} \|^2$$

The gradient of the function can be derived as:

$$\begin{aligned}\frac{\partial l(\theta)}{\partial \theta} &= \sum_{t=1}^{T} \frac{\partial}{\partial \theta} \| y_t - g_2(y_t; \theta) - I_t \|^2 \\ &\quad + \frac{\partial}{\partial \theta} \sum_{t=1}^{T} \| y_{t+1} - g_2(y_t; \theta) - g_1(y_t; \theta) - I_{t+1} \|^2 \\ &\quad + \lambda_2 \frac{\partial}{\partial \theta} g_2(y_t; \theta)^T R g_2(y_t; \theta) \\ &= 2\sum_{t=1}^{T} \frac{\partial}{\partial \theta} (g_2(y_t; \theta) - y_{t+1} - I_t) g_2'(y_t; \theta) \\ &\quad + 2\left(\sum_{k=1}^{2} g_k(y_t; \theta) - y_{t+1} - I_{t+1}\right) \sum_{k=1}^{2} g_k(y_t; \theta) \\ &\quad + 2\lambda_2 g_2'(y_t; \theta) R g_2(y_t; \theta)\end{aligned}$$

where both $g_1'(y_t; \theta)$ and $g_2'(y_t; \theta)$ can be computed by the back-propagation through the network. It is possible to use the weight-sharing techniques to let $g_1(y_t; \theta)$ and $g_2(y_t; \theta)$ share a few layers to reduce the number of parameters needed to train the model. In the noiseless case, we can simply set $g_2(y_t; \theta) = y_t + I_t$, since $e_t = 0$ in (1). Finally, the stochastic gradient descent algorithm (3) can be used for the parameter update.

Long-term Prediction: We will discuss how to enable the long-term prediction for AR-CNN model. Here we denote $\mu_{t_0}(t_0 + \Delta t)$ as the prediction of $\mu(t_0 + \Delta t)$ at time t_0. For longer term prediction, the following method can be used. We know that $\hat{\mu}(t_0) = g_1(y_{t_0}; \theta) + I_{t_0}$ and $\hat{\mu}(t_0 + i + 1) = I_{t_0+i+1} + \hat{\mu}(t_0 + i) + g_2(y_{t_0+i}; \theta)$. Therefore, we can derive the following formula for the long-term prediction.

$$\hat{\mu}_{t_0}(t_0 + \Delta t) = g_1(y_{t_0}; \theta) + \sum_{i=1}^{\Delta t - 1} g_2(\hat{\mu}_{t_0+i}; \theta) + I_{t_0+\Delta t}$$

2) Convolutional Long Short-Term Memory Networks Model (Conv-LSTM)

LSTM is one type of recurrent neural network that is designed to learn the long-term dependencies. They are widely used in a large variety of problems such as time-series prediction [16] and natural language processing [17]. However, the LSTM method is not suitable to model the spatio-temporal propagation since it uses the fully connected transition matrices on the hidden state, which cannot take advantage of the spatial neighborhood structure during the temporal transition and could potentially lead to the overfitting [18]. In contrast, Conv-LSTM method is proposed in [18] to model this local propagation via the convolutional operator as follows:

$$\begin{aligned} i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\ C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\ o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\ H_t &= o_t \circ \tanh(C_t) \end{aligned}$$

Here, '$*$'denotes the convolution operator and '$\circ$' denotes the Hadamard product. $h_t = (C_t, H_t)$ are the memory state of Conv-LSTM at time t. In another word, LSTM is a function to link the data and previous system state as:

$$o_t, h_t = g(y_{t-1}, h_{t-1}), \ \mu_t = f_{yo}(o_t) + \mu_{t-1} + I_t.$$

Again, motivated by the Euler's equation, the LSTM model $f_{yo}(\cdot)$ is used to model the difference between μ_t and μ_{t-1}. Furthermore, another convolution neural network f_{yo} can be used to link o_t and future prediction y_t.

Inference and Optimization Techniques: Here, the loss functions can be defined in (2).

Both $\frac{\partial f(\mu_t;\theta)}{\partial \theta}$ and $\frac{\partial \mu_t}{\partial \theta}$ can be computed from the backpropagation. For example, $\frac{\partial f(\mu_t;\theta)}{\partial \theta} = \frac{\partial f(\mu_t;\theta)}{\partial \mu_t}\frac{\partial \mu_t}{\partial \theta} + \frac{\partial f(\mu_t;\theta)}{\partial \theta}$. $\frac{\partial \mu_t}{\partial \theta}$ can be computed recursively as $\frac{\partial \mu_t}{\partial \theta} = \frac{\partial \mu_{t-1}}{\partial \theta} + \frac{\partial f(\mu_{t-1};\theta)}{\partial \theta}$. In reality, to be able to derive $\frac{\partial \mu_t}{\partial \theta}$, we will have to derive the results from $\frac{\partial \mu_1}{\partial \theta}$, $\frac{\partial \mu_2}{\partial \theta}$, ⋯, $\frac{\partial \mu_t}{\partial \theta}$, which increase the computational complexity dramatically for large t. Normally, truncated backpropagation can be applied to cut the gradient flow in the latest few measurements to decrease the computational complexity.

***Long-term Prediction*:** Similarly, we would like to discuss how to enable long-term prediction for the Conv-LSTM model. Unlike the AR-CNN model, the predicted value $\hat{\mu}_{t_0}(t_0+1)$ requires all values $y_i, i=1,\cdots t_0$. We divide the long-term prediction into two phases: the warm-up phase and the prediction phase. In the warm-up phase, we will start with y_0 or some value from $y_{i'}, i'<t_0$ to learn a more accurate memory state representation h_i from the original data. In the warm-up phase, the y_i is known for $i=1,\cdots t_0$, therefore, y_i can be used as input for the Conv-LSTM model. In the prediction phase, y_i is not known for $i=t_0+1,\cdots,t_0+\Delta t$. In this case, we propose to use the future prediction $\hat{\mu}_i$ for $i=t_0+t,\cdots,t_0+\Delta t$. This long-term prediction can also combine the stimulation information I_t. If we don't know where and when the future stimulation is, we can set $I_i=0$.

Algorithm: Long-term prediction for Conv-LSTM

Warm up:

$$o_{i+1}, h_{i+1} = g(y_i, h_i),$$
$$\hat{\mu}_{i+1} = f_{yo}(o_i) + \hat{\mu}_i + I_{i+1}, \quad i=1,\cdots t_0.$$

Prediction:

$$o_{i+1}, h_{i+1} = g(\hat{\mu}_i, h_i),$$
$$\hat{\mu}_{i+1} = f_{yo}(o_i) + \hat{\mu}_i + I_{i+1}, \quad i=t_0+1,\cdots,t_0+\Delta t.$$

III. Case Study

A case study is done using simulation data of electrical wave propagation on a one-dimensional cell cable. The performance of the proposed model is evaluated and compared, which are presented in this Section.

A. Simulation Setup

First, we performed simulations on a one-dimensional cell array (200 cells) using the FHN model and the mono-domain tissue model. Samples (i.e., cell membrane potentials) generated by simulations are used to learn the metamodel. Specifically, two simulation protocols are considered, which are described as follows:

- *Case 1*: One stimulation at a variable frequency of 1/12 to 1/4 Hz in a 1/40Hz increment is given to the left end of the cell array, which triggers electrical waves propagate to the other end of the cable. In addition, more experiments were done by moving the stimulation to the right of the cable in a step of 10 cells. For example, stimulation is given at the 180th cell in a frequency of 1/12Hz (See Fig. 1 left).
- Case 2: Two periodic stimulations at a variable frequency of 1/12 to 1/4 Hz in a 1/40Hz increment are given at different locations. The two stimulations are at a variable distance of 20 cells to 200 cells with a 20-cell increment. For example, the stimulations are given at the 20st cell and the 100th cells (see Fig. 1 right).

We have generated 105 samples in case 1 and 275 samples in case 2. Each sample contains 5000 measurements in 1ms time step. Here, we show two examples of the generated spatio-temporal processes in Fig. 1. Finally, we divide 80% samples for training and the rest 20% for testing.

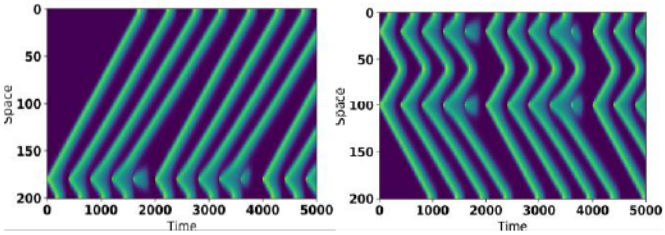

Fig 1. Simulated data for 2 different scenarios in the testing samples

B. Result Comparison

To evaluate the performance of the proposed method, we compared the prediction accuracy of the proposed models on the cell action potential (AP), i.e., $\mu(s,t)$. mean square error (MSE) of the predicted mean for Δt ms were computed as $\frac{1}{NTn_x}\sum_{i=1}^{N}\sum_{t=1}^{T}||\hat{\mu}_{i,t_0}(t_0+\Delta t)-\mu_{i,t_0}(t_0+\Delta t)||^2$, where T is the length of the sequence in the testing data, N is the number of testing samples, and n_x is the length of the spatial dimensions. For the benchmark method, we will compare with the widely used auto-regressive model, where the autoregressive time series model (AR) is applied in each spatial location independently.

The prediction accuracy of both AR-CNN and Conv-LSTM models are calculated and compared using testing data from both the two cases. For case 1, a stimulation at a frequency of 1/2Hz is given to the 180th cell of the cable, which generates a series of electrical waves propagating to both ends of the cable. For case 2, a more complicated scenario is considered, where two stimulations are given at the 40th cell and the 80th cell in a frequency of 1/2Hz. Two electrical waves are generated and propagate toward each other and to both ends of the cable respectively. The result of the MSE of Δt=5ms, 10ms, 50ms, 100ms, and 200ms is shown in Table 1. We would also like to report the computational time for the proposed AR-CNN and Conv-LSTM method, which takes only 0.7ms and 0.8ms for each prediction, which is computationally efficient.

TABLE I. TABLE I: LONG-TERM PREDICTION ACCURACY IN THE NOISELESS CASE

Case 1			
Δt (ms)	**AR-CNN**	**Conv-LSTM**	**AR**
5	**2.8e-5 (1e-5)**	3.9e-5 (1e-5)	2.7e-4 (7e-5)
10	9.0e-5 (3e-5)	**4.1e-5 (1e-5)**	1.1e-3 (3e-4)
50	1.9e-3 (1e-3)	**5.0e-4 (2e-4)**	0.023 (0.03)
100	5e-3 (1e-3)	**1.0e-3 (6e-4)**	0.064 (0.02)
200	1e-2 (4e-3)	**4e-3 (6.2e-4)**	0.140 (0.04)
Case 2			
Δt	**AR-CNN**	**Conv-LSTM**	**AR**
5	**3.8e-5 (2e-5)**	3.9e-5 (1e-5)	2.7e-4 (7e-5)
10	8.5e-5 (9e-5)	**4.0e-5 (1e-5)**	1.1e-3 (3e-4)
50	1.6e-3 (2e-3)	**5.0e-4 (2e-3)**	0.02 (0.03)
100	5.9e-3 (1e-3)	**1.0e-3 (7e-4)**	0.06 (0.02)
200	0.013 (8e-3)	**1.5e-3 (6e-4)**	0.14 (0.04)

From Table I, we can conclude that in both cases, conv-LSTM out-performs AR-CNN and AR model. When Δt is small (i.e. Δt=5ms), AR-CNN model outperforms conv-

LSTM model by a small margin. This is because the spatial information dominates the predictions in the short-term prediction. However, if we increase the Δt to 10ms, the MSE of conv-LSTM does not change too much for both Case 1 and 2. However, the mean squared error (MSE) of AR-CNN method doubles. This shows the importance of adding the temporal information in the prediction. Finally, AR performs the worst since it does not have the capacity to consider the complex nonlinear spatial or temporal information. Further, the MSE of AR-CNN and CLSTM method with Δt from 1ms to 250ms are shown in Fig. 2. The MSE of AR is not shown in Fig. 2 since its performance is not comparable to the other two model as shown in Table I.

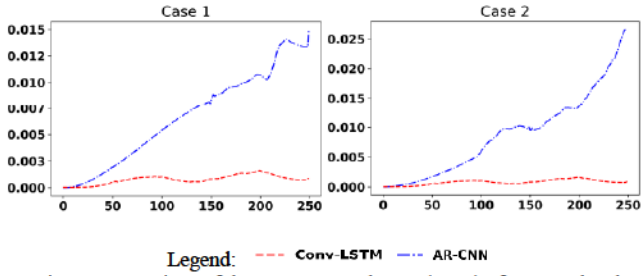

Fig. 2. Comparison of the mean squared error (MSE) of 250ms ahead prediction of AR-CNN and Conv-LSTM.

Fig. 3 shows the predicted electrical waves on the cell array at Δt =200ms ahead at time $t_0 = 3000$ (i.e. t_0 is shown as origin 0 in Fig. 3), i.e. $\mu_{t_0}(t_0 + \Delta t)$ for both case 1 and case 2. As seen in Case 1, both the Conv-LSTM and the AR-CNN models can provide an accurate estimation of the first three waves. However, Conv-LSTM model shows a better performance in predicting the magnitude of the fourth wave. It is worth noting that the AR-CNN model generates prolonged wave tails in each cycle as compared to the Conv-LSTM model and the true data. This could be the reason that AR-CNN model fails to capture the 4th wave because when the wave begins, the 3rd wave by the AR-CNN model has not finished. Due to the refractoriness of the cardiac cell, new excitation cannot be initiated. For Case 2 (see Fig. 3 right), the left and right waves merge at the 60th cell. Only the Conv-LSTM method can accurately capture this merge event by predicting the magnitude and shape of each peak event accurately. Similar to Case 1, the AR-CNN model shows some refractoriness to frequent stimulations and fails to capture the peaks around cell 50 and cell 120.

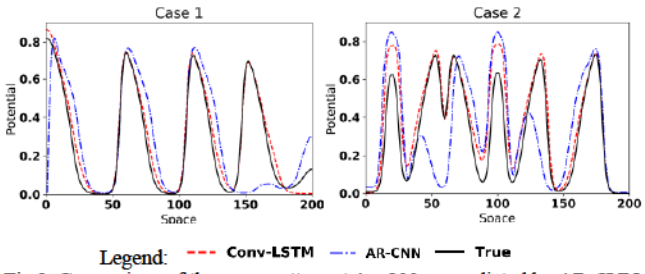

Fig 3. Comparison of the wave pattern at Δt=200ms predicted by AR-CNN and Conv-LSTM vs true data.

We also demonstrate one example for long-term prediction of Δt =1ms to 1500ms time-ahead prediction by showing the AP of the 60th cell at time t_0 =3000ms, i.e., $\mu_{t_0}(s = 60, t_0 + \Delta t)$, $\Delta t = 1, \cdots, 1500$ on the cell cable in Fig. 4, where the horizontal axis shows time (i.e. origin 0 in the horizontal axis is t_0) and the vertical axis shows the cell AP. For case 1, the proposed algorithm can predict the first two waves in 1000ms,

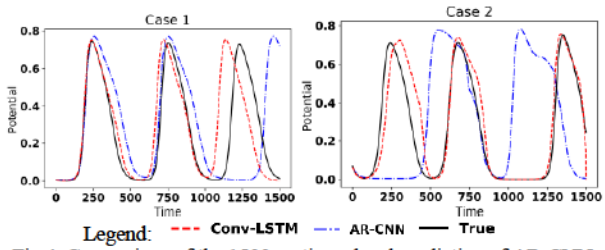

Fig 4. Comparison of the 1500ms-time ahead prediction of AR-CNN, Conv-LSTM, and true data. The origin 0 is the start of prediction time t_0

where Conv-LSTM (red dash line) shows better accuracy as compared to AR-CNN (blue dot line). For the third activation, Conv-LSTM still shows better performance than AR-CNN, although both models show larger discrepancy to the true data. It is worth mentioning that both models can learn the recurrent patterns of electrical excitations. The Conv-LSTM model captures a smaller cycle length as compared to the true cycle length, whereas the AR-CNN model learns a larger stimulation cycle as compared to the data. In addition, the AR-CNN model has prolonged AP durations (see Fig. 4 Case 1 for prolonged wave tails). This is consistent with the results observed in the previous section, where the AR-CNN model shows longer wave tails in the spatial domain (see Fig. 3 case 1). The proposed Conv-LSTM model can provide accurate predictions of AP in two consecutive cellular excitations in 1000ms, which is favorable to the AR-CNN model.

In case 2, a more complex scenario was considered, where two stimulations were given, and two electrical waves were generated and propagate toward each other and to the left and right end of the cell array. We investigated the performance of the two models and the results are shown in Fig. 4 (i.e. t_0 is shown as origin 0). Similarly, the AP of the 60th cell of the cable predicted by Conv-LSTM and AR-CNN were plotted and compared to the simulation data. 60th is selected because two waves merge together at the location. As see in Fig. 4, it is obvious that only the Conv-LSTM model is able to predict the three waves correctly, whereas the AR-CNN method is not able to capture this behavior. These findings show that Conv-LSTM architecture has much better accuracy for the long-term prediction in the complex cardiac system.

Further, we plot the spatio-temporal potential map for 200ms predicted by the AR-CNN method and Conv-LSTM method for both Case 1 and Case 2 in Fig. 5 (t_0 is shown as origin 0). As see in Fig. 5, the horizontal axis shows time, and the vertical axis shows the cell index. Different colors show different potential values, i.e., $\mu_{t_0}(s, t_0 + \Delta t)$ for different s and Δt. For case 1 (i.e. see left column of Fig. 5), a sequence of stimulations is given at the 180the cell prior t_0, which generate four waves propagate along cable. At $\Delta t = 150ms$, a new stimulation is given, which triggers a new electrical wave. The left column of Fig. 5 compares the predicted maps and the true data, where both the two models show good accuracy in predicting the first 3 waves (i.e., waves between cell 0 to cell 150). However, AR-CNN model fails to capture

the 4th wave between cell 150 and cell 200. The is possibly because AR-CNN model tends to lean prolonged AP durations and extended electrical wave tails from training data. This will further extend the excitation of each cell. Due to the refractoriness of cardiac cells, it is hard to generate new waves when the previous ones have not finished.

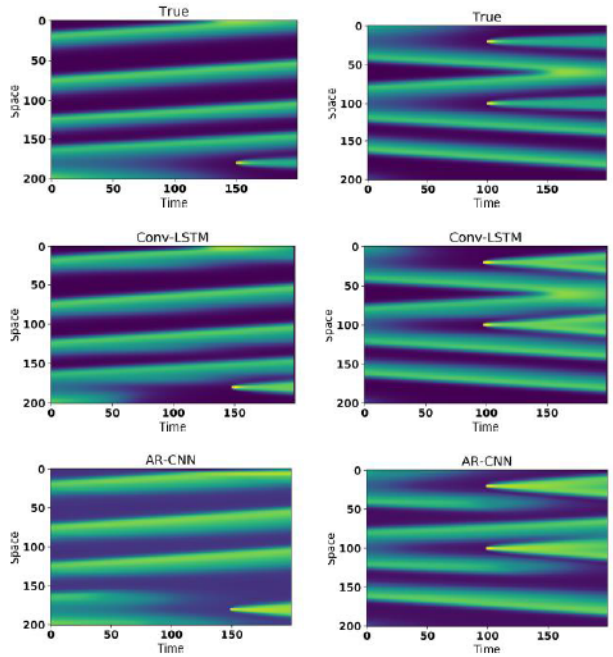

Fig 5. Comparison of 200ms-ahead prediction of AR-CNN, Conv-LSTM, and true data in the two simulation scenarios. Left: Case 1, Right: Case 2.

In case 2, as aforementioned, two stimulations are given to the 40th and the 80th cell, which generate electrical wave propagate in both directions. Multiple waves are formed prior to the starting time t_0=0 and conduct along the cell cable. In addition, new stimulations are given at Δt, where Δt =100ms, which initiate two waves. Given the potential at t_0 and the stimulation information, Conv-LSTM model and AR-CNN models are applied to prediction the spatio-temporal distribution of AP in 200ms. As seen in Fig. 5, similar results are observed, where the Conv-LSTM model can capture all the electrical waves in both temporal and spatial domains, whereas, the AR-CNN model cannot predict the second wave when new stimulations present (see cell 25-50 and cell 100-125 at 120ms to 200ms for AR-CNN model in Fig. 5 right column). This shows that the AR-CNN structure does not learn the dynamics of the merging events. In addition, both models tend to predict larger wave durations as compared to the true data (see cell 25 and cell 50 at 100ms to 200ms in Fig. 5 right column). This result is consistent with the observations in Fig. 3 case 2.

IV. Conclusion

Modeling and simulation have been widely used in both cardiac research and clinical study to investigate cardiac disease mechanism. In this work, we proposed a physics-based deep spatio-temporal regression approaches for more accurate meta-modeling of the system with two architectures, the Conv-LSTM model and the AR-CNN model. Both models have shown great prediction accuracy in the short time-horizon prediction, but the Conv-LSTM model shows a very good long-term prediction accuracy. These findings could be potentially used for better monitoring, modeling, and diagnosis of the clinical trial. For future work, we plan to study the meta-modeling in a higher-dimensional (e.g. 2D or 3D) and irregular spatial domain for better cardiac modeling.

Acknowledgment

This project is partially supported by NSF DMS-1830363, CMMI-1646664, and CMMI-1728338.

References

[1] A. Marrel, N. Perot, and C. Mottet, "Development of a surrogate model and sensitivity analysis for spatio-temporal numerical simulators," *Stochastic environmental research and risk assessment,* vol. 29, no. 3, pp. 959-974, 2015.

[2] C.-h. Luo and Y. Rudy, "A model of the ventricular cardiac action potential. Depolarization, repolarization, and their interaction," *Circulation research,* vol. 68, no. 6, pp. 1501-1526, 1991.

[3] E. M. Izhikevich and R. FitzHugh, "Fitzhugh-nagumo model," *Scholarpedia,* vol. 1, no. 9, p. 1349, 2006.

[4] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, and C. J. Wu, "Building surrogate models based on detailed and approximate simulations," *Journal of Mechanical Design,* vol. 128, no. 4, pp. 668-677, 2006.

[5] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, 2003, pp. 63-71: Springer.

[6] M. Raissi and G. E. Karniadakis, "Hidden physics models: Machine learning of nonlinear partial differential equations," *Journal of Computational Physics,* vol. 357, pp. 125-141, 2018.

[7] F. Sigrist, H. R. Künsch, and W. A. Stahel, "Stochastic partial differential equation based modelling of large space–time data sets," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 77, no. 1, pp. 3-33, 2015.

[8] M. Alvarez, D. Luengo, and N. Lawrence, "Latent force models," in *Artificial Intelligence and Statistics*, 2009, pp. 9-16.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[10] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[11] Z. Long, Y. Lu, X. Ma, and B. Dong, "Pde-net: Learning pdes from data," *arXiv preprint arXiv:1710.09668,* 2017.

[12] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations," *arXiv preprint arXiv:1711.10566,* 2017.

[13] J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," *Journal of Computational Physics,* vol. 375, pp. 1339-1364, 2018.

[14] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri, "Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations," *Applied Numerical Mathematics,* vol. 25, no. 2-3, pp. 151-167, 1997.

[15] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*: Springer, 2010, pp. 177-186.

[16] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches," in *Neural Nets WIRN Vietri-01*: Springer, 2002, pp. 193-200.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[18] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4580-4584: IEEE.