Comparing Surrogate Models for Action Potential Features in Ischemic Conditions

Yan B. Werneck[1], Lucas T. Oliveira[1], Anders Nygren, Brodie A. J. Lawson, Kevin Burrage[3], Bernardo M. Rocha[1,2], Joventino O. Campos[2], Rodrigo W. dos Santos[1,2],

[1]Computational Modeling Graduate Program - Federal University of Juiz de Fora, Juiz de Fora, Brazil
[2]Department of Computer Science - Federal University of Juiz de Fora, Juiz de Fora, Brazil
[3]Queensland University of Technology
[4]Schulich School of Engineering - University of Calgary

Abstract

Electrophysiological action potential models have become useful in clinical applications, offering deep insights into the behavior of complex biological phenomena when combined with tools such as uncertainty quantification and sensitivity analysis. However, their application is often hindered by high computational costs, particularly when extensive model sampling is required for fitting and deploying patient-specific models. This study explores the use of three kinds of surrogate models - Neural Networks, Polynomial Chaos Expansions and Gaussian Processes - as efficient alternatives to reduce computational overhead while maintaining accuracy in modeling action potential in ischemic conditions.

1. Introduction

Understanding the behavior of cellular action potential models is essential, as these models integrate biophysical mechanisms, thereby offering significant explanatory and predictive power. This predictive capability is particularly useful in cardiac electrophysiology, as it allows the simulation of various scenarios under different conditions.

To enhance the credibility of simulations, it is crucial to assess the relevance of each parameter employed by cardiac models. This can be accomplished through the application of techniques such as uncertainty quantification and sensitivity analysis [1]. However, as model complexity increases, these tools require large datasets to yield stable results, making the use of direct simulations or real data computationally prohibitive.

Emulators emerge as a viable solution for generating the large datasets required for these analyses, offering substantial advantages in terms of computational cost and execution time compared to direct model simulations. Among the main surrogate models used in the cardiac modeling context, the following stand out: Polynomial Chaos Expansions (PCE), as explored by Del Corso et al. (2020) to perform global sensitivity analysis and uncertainty quantification in a monodomain model of cardiac electrophysiology [2]; Gaussian Processes (GP), applied by Coveney et al. (2021) to reproduce restitution curves, thereby facilitating Bayesian calibration and sensitivity analysis of clinical models from noisy data [3]; and finally, Neural Networks (NN), as investigated by Pagani and Manzoni (2021), which integrated this technique with reduced order modeling to emulate complex simulations, facilitating uncertainty quantification and sensitivity analysis of parameters in electrophysiological models.

In this context, this study utilizes the model proposed by ten Tusscher [4], a well-established and relatively simple human action potential model, under disease condition [5], to evaluate the performance of various surrogate models.

2. Methods

This study evaluates the ability of emulators to capture action potential (AP) dynamics under disease conditions and across varying cell physiologies. The models have two core components based on parameterizations of the ten Tusscher 2004 (TT) model [4], as defined in Table 1: an ischemic model representing different degrees of acidosis, hypoxia and hyperkalemia conditions, as defined in [5]; and a cell diversity model that incorporates ionic conductance variations to account for diverse AP waveforms within a population. The resulting AP morphologies are characterized by four key quantities of interest (QoIs): dV/dt_{max} (maximum upstroke velocity), APD50 and APD90 (AP durations at 50% and 90% repolarization), and $V_{\mathtt{rest}}$ (resting potential). To minimize the impact of transient effects, QoIs are measured after 20 AP cycles, with 1 second between stimulus.

We evaluate three types of regression models as emula-

tors to model the relation between parameters and AP features: Gaussian Processes (GP), Neural Networks (NN), and Polynomial Chaos Expansions (PCE). We consider two degrees of complexity: Model A, a disease model that includes only three ischemic parameters, ranging from 0 to 1 scaling the degree of acidosis, hypoxia and hyperkalemia; and Model B, that extends this base ischemic model by adding scaling coefficients to base ten Tusscher [4] conductance values to account for cell diversity within a population, for a total of 12 parameters. The solution space of both models is shown in Figure 1. Each model is used to generate datasets to train the emulators, sampling uniformly each of the parameters and running the base TT model. Training set sizes range from 100 to 5,000 samples, and the validation set has 100,000 samples. Key performance metrics included training/inference times and memory usage, and accuracy, in the form of the Mean Absolute Relative Error :

$$MARE_Q = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i^Q - \hat{y}_i^Q}{y_i^Q}\right| \quad (1)$$

for a dataset with N samples a QOI Q. Models with iterative training (NN and GP) were allowed to train until convergence (with tolerance = 1×10^{-6} and only stopping after 1000 iterations with no improvement).

Gaussian Process. Gaussian Processes (GPs) are non-parametric models that define distributions over functions. Given training data $\{\mathbf{x}_{\text{train}}, \mathbf{y}_{\text{train}}\}$, the GP posterior mean prediction for a new input $\mathbf{x}$ is:

$$\hat{\mathbf{y}} = m(\mathbf{x}) + \mathbf{k}(\mathbf{x}, \mathbf{x}_{\text{train}})\mathbf{K}^{-1}_{\text{train,train}}(\mathbf{y}_{\text{train}} - \mathbf{m}(\mathbf{x}_{\text{train}})). \quad (2)$$

where $m(\mathbf{x})$ is the mean function and $k(\mathbf{x}, \mathbf{x}')$ is the kernel function [6]. The matrix $\mathbf{K}_{\text{train,train}}$ represents the covariance among training points, training usually involves solving it's associated linear system via Cholesky decomposition. Thus GP training scales as $O(N^3)$, while inference scales as $O(N^2)$, yielding the predictions of both mean and variance. Three GP models are considered. GP_S, employing a constant mean function with an RBF kernel. GP_M, that incorporates a linear mean and a composite RBF-Matérn ($\nu = 2.5$) kernel, and GP_L, that uses a constant mean combined with a composite RBF-Matérn ($\nu = 1.5$)-linear kernel. GPs are implemented using Gpytorch [7], which provides a framework for efficient inference and training using GPUs, including conjugate gradient approximation instead of the decomposition when the training set is larger than 2000 samples.

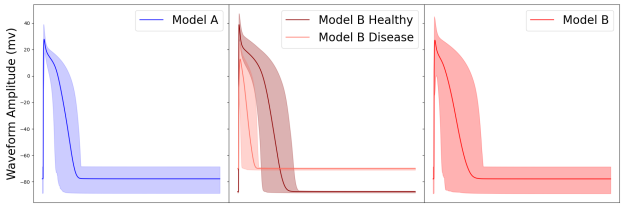

Figure 1. AP waveforms for Models A and B. (left) Model A with all parameters varying from 0 to 1 (disease-free to maximum disease). (center) Model B with conductance coefficients ranging from 0.5 to 1.5, ischemic parameters fixed at 0 or 1. (right) Model B with ischemic parameters varying from 0 to 1, and conductances from 0 to 1.5.

Table 1. Model parameters mapped to ten Tusscher parameters values (Input). Ischemia model parameters range from 0 to 1 corresponding to healthy and diseased conditions, scaling linearly. Cell population model has conductance parameters ranging from 0.5 to 1.5 and are coefficients that directly multiply TT parameters to capture diverse AP waveforms.

Model	**Parameter**	**Input**	**Base Value**	**Diseased Value**
Ischemia model	Acid	g_{Na}	15.0	11.0
		g_{Cal}	0.175	0.131
		K_i	138.0	125.0
	Hypox	ATP	5.40	4.05
	Hyper	K_0	5.60	10.0
Cell Diversity model	**Parameter**	**Input**	**Base Value**	**Range**
	g_{NaC}	g_{Na}	g_{Na}	$0.5 - 1.5$
	g_{CalC}	g_{Cal}	g_{Cal}	$0.5 - 1.5$
	g_{K1C}	g_{K1}	5.40	$0.5 - 1.5$
	g_{KrC}	g_{Kr}	96.0	$0.5 - 1.5$
	g_{KsC}	g_{Ks}	245.0	$0.5 - 1.5$
	g_{toC}	g_{to}	294.0	$0.5 - 1.5$
	g_{bCaC}	g_{bCa}	0.592	$0.5 - 1.5$
	g_{pCaC}	g_{pCa}	g_{pCa}	$0.5 - 1.5$
	g_{p_K}	g_{pK}	0.0146	$0.5 - 1.5$

Neural Network Emulators. Neural Networks (NNs) approximate functions via composite transformations [8]:

$$\hat{\mathbf{y}} = \sigma(W^{(L)}(\sigma(W^{(L-1)}\cdots\sigma(W^{(1)}\mathbf{x}+\mathbf{b^{(1)}})+\mathbf{b^{(L-1)}}))+\mathbf{b^{(L)}}), \quad (3)$$

where $W^{(i)}$ and $\mathbf{b^{(i)}}$ are layer parameters. We explore: NN_S (1 layer, 16 neurons), NN_M (2 layers, 32 neurons each), and NN_L (4 layers, 64 neurons), using a SILU function for σ. Parameters are optimized by fitting the training data using Adam (lr=$1e^{-4}$) and the L_2 loss. The number of parameters and thus training size mainly scales with network size, requiring iterative updates through backpropagation of a loss function. Inference scales only with model size (the number of layers and neurons), consisting of sequential matrix-vector multiplication at each layer. Neural networks are trained and evaluated using performance-focused custom PyTorch [9] code, leveraging GPUs.

Polynomial Chaos Expansion. Polynomial Chaos Expansion (PCE) represents outputs as weighted sums of orthogonal polynomials [10]:

$$\hat{\mathbf{y}} = \sum_{i=0}^{P} c_i \Phi_i(\mathbf{x}), \quad (4)$$

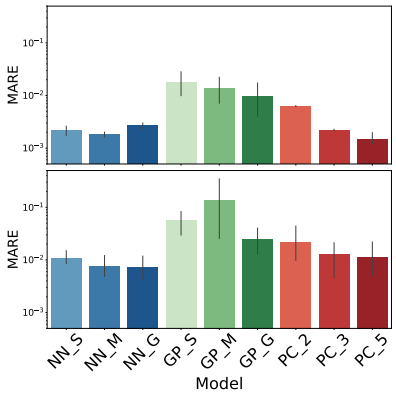

Figure 2. Comparison of MARE (average of all QoIs) for emulators of the models A (top) and B (bottom).The plots illustrate variation for surrogates trained with more or less data (0.1K to 5K samples) with the black lines.

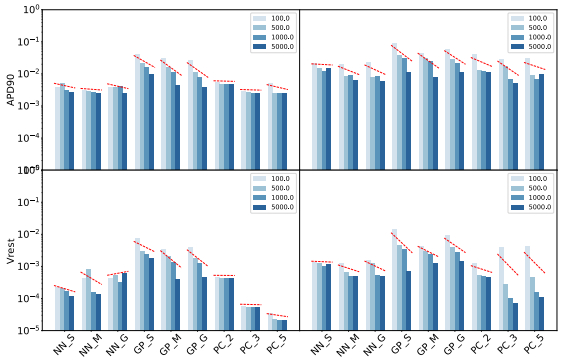

Figure 3. MARE for APD90 and Vrest of models A (left) and B (right), showing results by training size and the impact of increasing set size (dashed line indicates the slope).

where $\Phi_i(\mathbf{x})$ are basis polynomials of degree P, and c_i are regression coefficients. Higher-degree expansions capture more variance but increase complexity. We use Legendre polynomials of degrees 2, 3, and 5. Number of model parameters and thus training cost scales polynomially with input dimensionality, making it ideal for low-dimensional problems but impractical for high-dimensional due to an exponential growth in basis functions. Inference scales with the number of basis terms, requiring polynomial evaluations and summations. Training and inference are done using ChaosPy [11] on a single CPU core.

3. Results

Results indicate that Model B was consistently more challenging to emulate than Model A. Model B surrogates exhibited MARE values are approximately ten times greater than for Model A, both when considering the average of QoIs (Figure 2), and when QoIs were considered individually (Figure 3). Notably, surrogates of Model B benefited significantly more from increasing the number of training samples, suggesting that the increase in model complexity, due to higher dimensionality, can be mitigated by larger training datasets (see red lines of Figure 2). However, this improvement was only observed in models with relatively higher expressive capacity, such as higher-order PCEs and large NNs. Interestingly, this effect is most pronounced in PCEs. For Model B, the PCE exhibits significantly greater expressive power than for Model A, enabling it to fit a larger amount of training data. This is because in PCE, the number of surrogate parameters, a good proxy for expressive capabilities, scales polynomially with the problem's dimensionality. In contrast, due to their reliance on the covariance matrix, GP benefit from an increase in training data across both models.

The performance of PCEs and NNs was similar, but as QoI or model complexity increases, N+he most accurate emulator for Model A, while NN_L was the best for Model B. Notably, while PC_5 and PC_3 yielded similar results to the neural networks, they required 5000 training samples for optimal results, while increasing the number of samples further than 500 for the networks had little effect. This suggests that even if model expressiveness is comparable for models of about the same size magnitude (PC_5 has around 6000 parameters and NN_G 13000), NNs are better at interpolating data from scarce training sets.

Conversely, GPs exhibited the worst performance in both models, despite showing a noticeable improvement in accuracy as the number of training samples increased. This trend was particularly promising in Model A, where the best GPs matched other emulators, even though requiring more training samples. GPs, however, proved to be less effective than the other surrogates as complexity increased.

Key trade-offs to emulator accuracy are the time to train and the inference speed of the surrogates. During training, PCEs were the most cost-effective, training in milliseconds with minimal memory usage and using only a single cpu core, while using a GPU, NNs required a few seconds and GPs anywhere from tens to up to hundreds of seconds depending on training set size. Figure 4 shows the balance between time for each trained surrogate to process 100,000 samples and their accuracy. Inference time was shortest for NNs due to fully parallelized computations on a RTX 4070 GPU. While GPs also benefited from GPU acceleration, inference cost is higher but also yields the variance in the prediction. For PCEs, while the use o single core had comparatively less impact during training, that is in large part serial, serial inference (around $4GHz$) is much slower than the other surrogates, this however could be mitigated by using a multi-core approach.

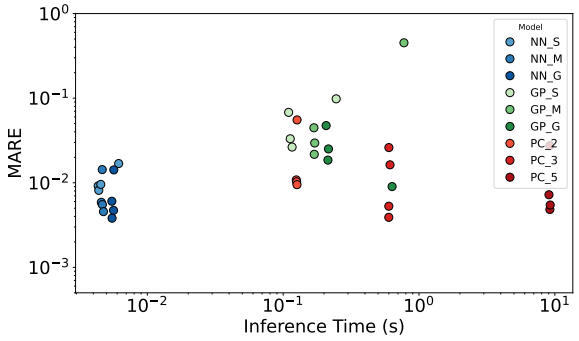

Figure 4. Cost-effectiveness comparison of surrogate models for Model A (MARE vs. inference time for 100,000 samples). Running 100,000 TT models in parallel on the same GPU takes 2450 s.

4. Conclusions

In this study, we evaluated the effectiveness of three surrogate modeling techniques—Polynomial Chaos Expansions, Neural Networks and Gaussian Processes—for emulating cardiac electrophysiological ischemia models under diverse cell conditions. Our results show a direct relationship between model complexity, emulator accuracy and training sample requirements. Furthermore, while PCE models were enough to yield good accuracy results, their lack of GPU support makes them very slow, although results could scale much better with multi-core support, even if only using cpus. In that sense, NN models outperform all others, achieving more than a million samples per second, while maintaining accuracy comparable to the best-performing models. Although GPs also utilize GPU acceleration, O(N^2) inference scaling makes it much slower, this however is balanced by it also inherently yielding the uncertainty associated with the predictions, which would require Monte Carlo sampling of the other surrogates.

This study highlights the efficiency of emulators in modeling AP dynamics. Once trained, surrogate inference achieved up to a million-fold speedup over the ten Tusscher model, with relative errors below 1%.While the initial cost of generating sufficient training data and optimization is equivalent to approximately 2000 model runs, this investment is increasingly offset as sample size requirements grow, making emulators especially valuable for large-scale tasks like parameter optimization and sensitivity analysis.

Acknowledgments

The authors thanks the Minas Gerais State Research Support Foundation (FAPEMIG), "Coordenaçãao de Aperfeiçoamento de Pessoal de Nível Superior" (CAPES), "Empresa Brasileira de Serviços Hospitalares" (Ebserh), National Council for Scientific and Technological Development (CNPq), SINAPAD Santos-Dumont, and Federal University of Juiz de Fora (UFJF) for funding this work.

References

[1] Campos JO, Sundnes J, dos Santos RW, Rocha BM. Uncertainty quantification and sensitivity analysis of left ventricular function during the full cardiac cycle. Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences 2020;378(2173):20190381.

[2] Winkler B, Nagel C, Farchmin N, Heidenreich S, Loewe A, Dössel O, Bär M. Global sensitivity analysis and uncertainty quantification for simulated atrial electrocardiograms. Metrology 2023;3(1):1–28. ISSN 2673-8244. URL https://www.mdpi.com/2673-8244/3/1/1.

[3] Coveney S, Corrado C, Oakley JE, Wilkinson RD, Niederer SA, Clayton RH. Bayesian calibration of electrophysiology models using restitution curve emulators. Frontiers in Physiology 2021;12. ISSN 1664-042X.

[4] Ten Tusscher KHWJ, Noble D, Noble PJ, Panfilov AV. A model for human ventricular tissue. American Journal of Physiology Heart and Circulatory Physiology 2004; 286(4):H1573–H1589. PMID: 14656705.

[5] Lawson BAJ, Oliveira RS, Berg LA, Silva PAA, Burrage K, dos Santos RW. Variability in electrophysiological properties and conducting obstacles controls re-entry risk in heterogeneous ischaemic tissue. Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences May 2020;378(2173):20190341.

[6] Williams Christopher K. R. CE. Gaussian processes for machine learning. MIT press Cambridge, MA, 2006.

[7] Gardner JR, Pleiss G, Bindel D, Weinberger KQ, Wilson AG. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. Advances in Neural Information Processing Systems NeurIPS 2018;31.

[8] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Networks 1989;2(5):359–366. ISSN 0893-6080.

[9] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S. Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems NeurIPS 2019;32.

[10] Xiu D, Karniadakis GE. The wiener–askey polynomial chaos for stochastic differential equations. SIAM Journal on Scientific Computing 2002;24(2):619–644.

[11] Feinberg J, Pinkus A. Chaospy: An open source tool for designing methods of uncertainty quantification. Journal of Computational Science 2018;11:46–57.

Address for correspondence:

Yan Barbosa Werneck
São Mateus 872, Juiz de Fora, MG, Brazil
yanbwerneck@outlook.com