Identifying drug effects in a cardiac model of electrophysiology using kernel-based parameter estimation methods

Andreas Oslandsbotn[*†‡] Alexander Cloninger[‡] Nickolas Forsch[†]

Abstract

Computational models of cardiac electrophysiology hold great potential in the drug discovery process to bridge the gap between *in vitro* and *in vivo* preclinical trials. Current methods for solving inverse problems in cardiac electrophysiology are limited by their accuracy, scalability, practicality, or a combination of these. We investigate the feasibility of using kernel methods to solve the inverse problem of estimating the parameters of ionic membrane currents from observations of corresponding action potential (AP) traces. In particular, we consider AP traces generated by a cardiac cell action potential model, which mimics those obtained experimentally in measurable *in vitro* cardiac systems. This proof-of-concept study aims to improve existing pipelines for identifying drug effects in "heart-on-a-chip" systems by introducing a new approach to solving the inverse problem.

Using synthetic training data from the 1977 Beeler-Reuter AP model of mammalian ventricular cardiomyocytes, we demonstrate our recently proposed boosted KRR solver (StreaMRAK), which is particularly robust and well-adapted for high-complexity functions. We show that this method is less memory demanding, estimates the model parameters with higher accuracy, and is less sensitive to parameter identifiability problems than existing methods, such as standard KRR solvers and loss-minimization schemes based on nearest neighbor heuristics.

1 Introduction

The current cardiac antiarrhythmic drug development pipeline is hindered by the cost and time required to develop and launch a new drug to market. Part of this barrier is attributable to the use of animal models for preclinical screening, which can sometimes fail to predict human cardiotoxicity [1]. For efficient drug development, measuring drug effects in human-based physiological systems, such as human-induced pluripotent stem cells (hiPSCs), is desirable. Furthermore, most antiarrhythmic agents interact with ion channels in the cellular and sub-cellular membranes to modulate ionic currents. While direct measurement of ionic current modulation has advanced from laborious, manual, low-throughput patch clamp techniques to higher-throughput, multi-cell platforms, these platforms require highly specialized laboratory expertise and high initial expense for the instrumentation [2, 3]. Meanwhile, the cardiac transmembrane potential, commonly referred to as the action potential (AP), is optically accessible using live cell fluorescence microscopy [4, 5]. The AP and its characteristic shape are determined by the biophysical dynamics of these ionic membrane currents [6, 7], which makes it possible, at least in theory, to quantify ionic currents through observations of APs.

Mathematically, we can represent the ionic membrane currents as parameters p_i and represent them as a vector $p = (p_i, \dots p_d) \in \mathcal{P}$ in some parameter space $\mathcal{P}$ of dimension d. Meanwhile, the corresponding AP trace can be considered a point $v \in \mathcal{V}$ generated by an unknown function $v = f(p)$. Panel A in Figure 1 illustrates this setting.

A natural approach to deriving the currents from a measured AP is constructing a model $\widetilde{f}$ of their relationship, which approximates the actual function f. Numerous cardiac action potential models describe this relationship, typically as ODE or PDE systems [8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. However, existing AP models are constructed to model the forward map $v = f(p)$, while we are interested in the

[*]University of Oslo, Problemveien 7, 0315 Oslo, Norway
[†]Simula Research Laboratory, Kristian Augusts gate 23, 0164 Oslo, Norway
[‡]University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, United States

inverse map $p = f^{-1}(v)$. In particular, given a measured AP trace $w \in \mathcal{V}$, we are interested in the underlying parameters p that generated w.

This paper investigates the feasibility of using kernel methods to learn the inverse map of model-generated cardiac-cell action potential traces. The purpose is to identify simulated drug-induced changes to the ionic currents in measurable *in vitro* cardiac systems, such as cardiac microphysiological systems (MPS). In particular, our contribution is improving the inversion step in the drug-effect-identification pipeline developed in [18], [19], and [20] by introducing a new approach to solving the inversion problem. We show that our method estimates drug-induced changes with higher reliability than existing methods in a simplified cardiac AP model, where reliability is measured in terms of estimation accuracy and variability of accuracy over a physiological range of cardiac cells.

Standard numerical methods for solving this inverse problem aim to minimize a loss function $L(v, \widetilde{f}(p))$ using either gradient-based optimization, such as gradient descent, or direct search methods (non-gradient-based), such as Particle swarm, Pattern-search, and Nelder-Mead. The standard strategy of these methods is to search in the parameter space $\mathcal{P}$ iteratively, and for each candidate, $p \in \mathcal{P}$, solve the ODE model $\widetilde{f}(p)$ to compare with the target w through the loss $L(w, \widetilde{f}(p))$.

However, traditional numerical inversion methods are inefficient in the face of large quantities of measured AP. This is because only one AP trace can be inverted at a time, and the AP model $\widetilde{f}$ must be computed for every candidate parameter. Although the former can be alleviated by parallelization, the computational expense of solving the AP model constitutes a computational bottleneck that prevents the scalability of traditional numerical inversion methods for this setting. Moreover, this computational bottleneck is growing since cardiac electrophysiology models are becoming increasingly large and elaborate [21, 22] and, therefore, more expensive to compute.

The inversion step in the drug-effect-prediction pipeline proposed by [18] used an alternative to traditional inversion strategies. The main idea of this inversion strategy is to use an AP model $\widetilde{f}$ to generate a pre-computed database of samples $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$, where $v_i = \widetilde{f}(p_i)$, and then minimize a loss function $L(w, v_i)$ on these samples. The solution to this problem is the parameter corresponding to the voltage, which is the nearest neighbor of w in $\mathcal{D}_n$ (measured under the metric defined by L). The advantage of this approach is that the computational burden is moved from the prediction step to a pre-computing phase, and the prediction of parameters for different w can reuse the same database. Consequently, the algorithm is significantly more scalable when we need to estimate the parameters of several measured AP traces. Furthermore, the minimization of L has access to all samples upfront, which offers a global overview that reduces the vulnerability to local minima.

The challenge with this approach is that the accuracy is directly linked to the density of pre-computed samples, a problem that is not present in more traditional inversion techniques. In particular, the accuracy of out-of-sample parameter estimations can only be as good as the distance to the nearest parameter in the pre-computed samples $\mathcal{D}_n$. In other words, accuracy would be expected to vary across the parameter domain as a function of the distance to samples in $\mathcal{D}_n$. Furthermore, the number of samples necessary to cover a d-dimensional region grows exponentially with the dimension [23]. Therefore, the training set $\mathcal{D}_n$ will need to be very large to prevent high variability in estimation accuracy, especially for models with many parameters. This translates to a significant upfront cost of computing $\mathcal{D}_n$. With a memory requirement of $\mathcal{O}(n)$, extensive resources must be dedicated to storing the pre-computed data set.

Highly reliable estimations are paramount in this clinically motivated study and approach. In this regard, sensitivity to local minima is undesirable as it can lead to significant errors and misleading characterizations of antiarrhythmic drugs. Likewise, considerable variability in accuracy over the parameter domain is also undesirable since the method must be robust over a physiological range of cardiac AP phenotypes.

In this study, we seek an inversion scheme that is scalable, memory efficient, and offers high reliability in parameter estimations: *scalable* in the sense that it efficiently estimates the underlying parameters of large quantities of AP traces; *memory efficient* in the sense that it requires little dedicated memory to compute the inversion and to store the trained model; and *reliable* in the sense that it has high accuracy and high consistency (low variability) over a physiological range of cardiac cell parameters.

The contribution of our proposed method is illustrated in Figure 1, where we compare it to existing inversion approaches. In particular, we present an alternative to the nearest-neighbor loss-function minimization approach used in [18], with increased accuracy and consistency. Furthermore, our proposed method reduces memory requirements from $\mathcal{O}(n)$ to $\mathcal{O}(L\sqrt{n})$ at the cost of an upfront computational

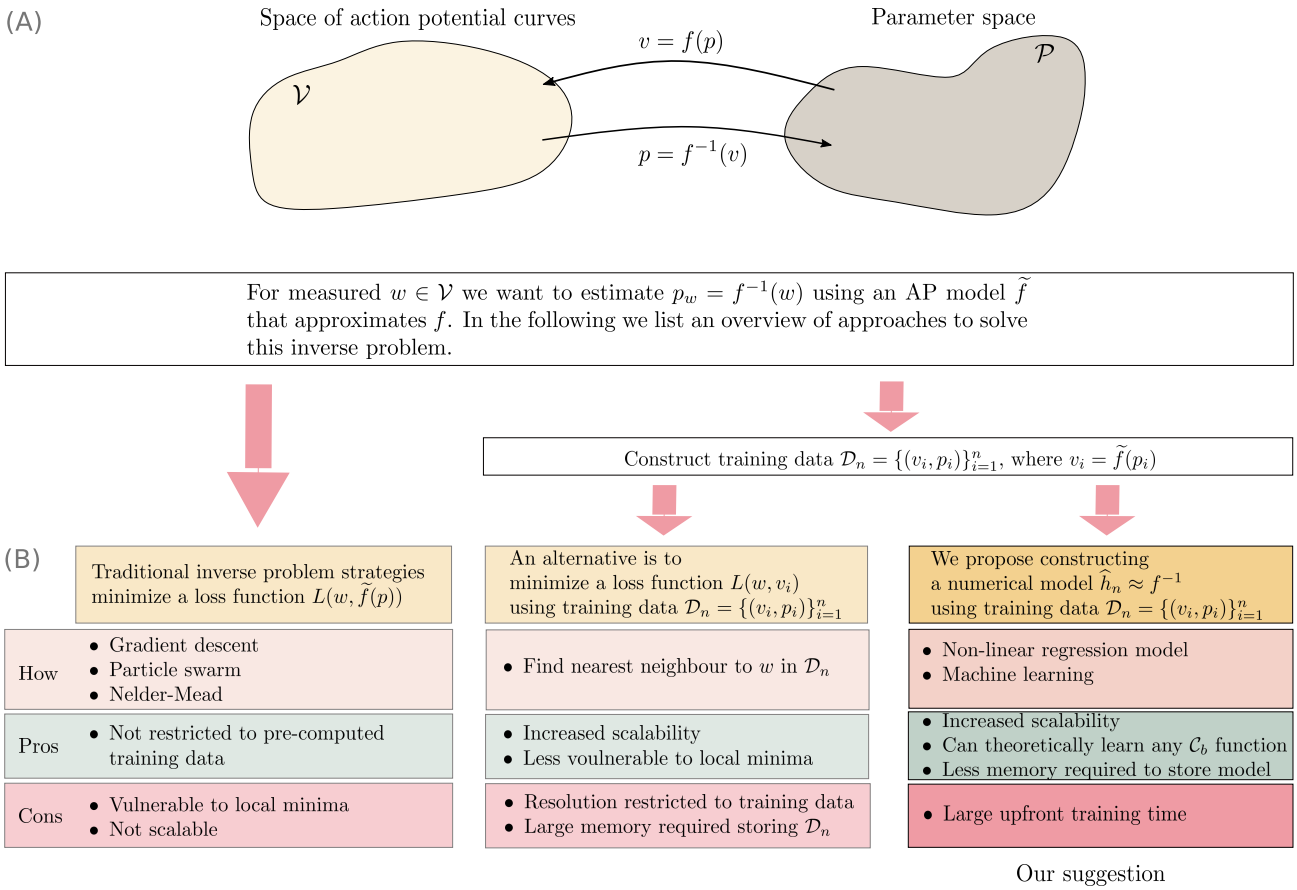


Figure 1: Panel (A) illustrates the mapping f and its inverse f^{-1} between the parameter space $\mathcal{P}$ and the voltage space $\mathcal{V}$ and defines the inverse problem we want to solve. Panel (B) compares our method with existing inverse problem strategies. Here $\mathcal{C}_b$ is the set of bounded and continuous functions.

training time. Here L is a constant independent of n.

In the following, we summarise the main aspects of our contribution. Let $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$ be a set of known training examples generated by an AP model $\widetilde{f}$ that approximates f.

- We propose to think of the inverse problem as a non-linear regression task, where the goal is to learn an approximation $\widehat{h}_n$ to the inverse map f^{-1} using $\mathcal{D}_n$ as training samples.
- Using our recently developed streaming multi-resolution adaptive kernel algorithm StreaMRAK [24], we demonstrate with synthetic experiments how we can construct a kernel model $\widehat{h}_n \approx f^{-1}$ which estimates the model parameters with increased accuracy and reduced variability compared to alternative regression models and nearest neighbor search approaches.
- For the analysis of our AP models, we propose a geometric approach to identifiability analysis, which supplements the tool developed in [19].

The regression model we propose, StreaMRAK, is a particular kernel method based on kernel ridge regression (KRR), developed to enable more efficient learning of functions with multiple modalities, low complexity in some regions and high complexity in others. StreaMRAK draws inspiration from wavelets and boosting principles to construct a multi-resolution regression function that incrementally improves predictions from past iterations. Furthermore, StreaMRAK is designed to avoid the exponential increase in training samples required for high-dimensional data, provided that the high-dimensional data has a lower-dimensional structure.

Kernel methods are among the most popular non-parametric learning algorithms, and KRR is the go-to algorithm for regression. The attraction of these algorithms is their strong theoretical foundation

[25, 26]. Furthermore, certain classes of kernels are universal kernels, meaning they can approximate bounded and continuous functions on a domain $\mathcal{X}$ arbitrarily well [27]. Despite significant computational demands in their traditional form [26], algorithmic improvements to KRR have made significant advances in recent years; [28, 29, 30, 31, 32, 24]; with application in a wide array of fields ranging from finance to bio-medicine [33, 34, 35, 36].

To use a non-parametric learning algorithm, such as StreaMRAK, is a natural choice since the target function f^{-1} is assumed to exhibit non-linear and complex behavior, and the only information we have on its shape and structure comes from training data generated by the action potential model $\widetilde{f}$. Non-parametric learning has the advantage that it does not rely on pre-determined shape assumptions and can therefore learn highly complex and non-linear functions of unknown shapes as long as sufficient data is provided.

To demonstrate our approach, we run our method and others on several synthetic experiments generated from the 1977 Beeler-Reuter (BR) mathematical model of mammalian ventricular cardiomyocytes [37]. The BR model was the first ventricular cell AP model to be developed and includes four different ionic currents: a fast inward sodium (Na^+) current, a slow inward current carried primarily by calcium (Ca^{2+}) ions, a time-dependent outward current, and a time-independent potassium (K^+) current. Many cardiac AP models of various cell types have been developed since the BR model was first introduced, incorporating more detailed electrophysiology through the representation of more specific currents, both in the cellular and sub-cellular membranes and compartments. With more detailed models come more parameters and variables which must be differentiated and resolved when solving the inverse problem. In this proof-of-concept study, we have selected a model which is easy to interpret and analyze while still containing currents that are of interest for studying the electrophysiological effects of drugs [38]. The learning algorithms demonstrated in this study can be generalized, in a straightforward manner, to other more realistic AP models with more parameters, states, and variables, such as a model of cardiomyocytes derived from human-induced pluripotent stem cells incorporating multiple intracellular compartments [39, 20].

2 Methods and Theory

In this section, we offer a detailed description of the method we propose for learning the inverse map between AP traces and the underlying parameters. We start by describing how we generate the synthetic training data. In particular, we present the 1977 Beeler-Reuter model, which is the AP model we consider in this study. Section 2.2 describes the kernel model we propose for the non-linear regression and the training scheme used to fit the model. Meanwhile, Section 2.3 briefly recaptures the loss-minimization schemes we compare with, and Section 2.4 offers an outline of specific parameter identifiability tools that we use to analyze the AP model for better understanding of the estimation results. Finally, Section 2.5 gives an overview of the study and the experiments we conduct.

2.1 Generating synthetic data from the AP model

The method we propose requires access to a set of training data $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$, where $p_i \in \mathbb{R}^d$ is a d dimensional vector of parameters representing different ionic membrane currents and v_i is a time series of transmembrane potential (voltage) v_i, which we refer to as the action potential (AP). The training data can be generated either from *in-vito* measurements or numerically solving an AP model. Although the transmembrane potential is a continuous function of time, in practice, whether we measure the AP traces *in-vito* or generate them from a system of ODEs, we need to sample the traces at a finite time grid. We denote this time grid $[t_1, \ldots, t_k, \ldots t_T]$, which we scale such that $t_1 = 0$ and $t_T = 1$. We think of these AP traces v_i as T dimensional vectors in the sub-space $\mathcal{V}_T \subset \mathbb{R}^T$, where each entry v_{ij} corresponds to a time step t_j. See panel (C) in Figure. 2 for an illustration.

Now, training samples generated by an AP model with the parameter set $p = (p_1, \ldots, p_d)$ restrict predictions to these parameters. For a given AP model $\widetilde{f}$, we consider some physiologically viable region $\mathcal{P} \subseteq \mathbb{R}^d$ from which we sample n parameters uniformly, and solve the BR model on each sample, generating the training data $\mathcal{D}_n = \{(\widetilde{f}(p_i), p_i) : p_i \sim \text{Uni}(\mathcal{P}), \text{ for } \quad i = 1, \ldots n\}$. See step (A)-(B) in Figure 2. In the remainder of this paper, we use the 1977 Beeler-Reuter AP model of mammalian ventricular cardiomyocytes to generate the training samples. We will now give a brief description of this model.

2.1.1 The action potential model

The Beeler-Reuter model is formulated as a system of ODEs relating the time derivative of the action potential to a set of 10 parameters $p = (p_1, \dots p_{10})$ representing specific membrane currents. Table 1 summarizes these parameters and the ionic currents they represent. Once solved numerically, the model outputs a time series of transmembrane potential (voltage) v_i, which we refer to as the AP.

Table 1: Parameters in the Beeler-Reuter model. The first column lists the parameter names. The second contains scaling constants, and the last column contains a description.

Parameters	Scaling	Description
E_{Na}	5.e+1 mV	Sodium reversal potential (mV)
g_{Na}	4.e-2 mS/mm^2	Parameter for the sodium (Na^{2+}) ionic current component (mS/mm^2)
g_{Nac}	3.e-5 mS/mm^2	Parameter for the sodium (Na^{2+}) ionic current component (mS/mm^2)
g_s	9.e-4 mS/mm^2	Parameter for the slow inward ionic current component (mS/mm^2)
$I_{stim-amp}$	5.e-1 $\mu A/mm^2$	Stimulus current amplitude ($\mu A/mm^2$)
$I_{stim-period}$	5.e+5 ms	Stimulus current period (ms)
$I_{stim-pulse}$	1.e+3 ms	Stimulus current duration (ms)
$I_{stim-end}$	1.e+0 ms	Stimulus current end time (ms)
$I_{stim-start}$	1.e+1 ms	Stimulus current start time (ms)
C	1.e-2 $\mu F/mm^2$	Cell capacitance ($\mu F/mm^2$)

The ODEs that make up the Beeler-Reuter model are described mathematically in [40]. However, instead of working directly with the ODE equations of the Beeler-Reuter model, it is sufficient for our purposes to consider the model as a mapping from the parameter space $\mathcal{P}$ to the space of AP traces $\mathcal{V}_T$, we defined this mapping in Definition 2.1. See also step (B) in Figure 2 for an illustration on how we use the model.

Definition 2.1 (The Beeler-Reuter function)**.** *Under the setting described above, the Beeler-Reuter model corresponds to a vector-valued function $F_T : \mathcal{P} \to \mathcal{V}_T$. In particular,*

$$v_i = F_T(p_i),$$

where $v_i = (v_{i1}, \dots, v_{ik}, \dots, v_{iT}) \in \mathcal{V}$ is the AP curve that corresponds to the parameter choice $p_i = (p_{i1}, \dots, p_{ij}, \dots, p_{id}) \in P$. Furthermore, the action potential at time t_k is

$$v_{ik} = F_T(p_i)(t_k) = f_T(p_i; t_k),$$

where each time step t_k defines a specific functional relationship between the parameters and the AP.

2.1.2 Pre-processing of the AP traces

We use single pulse AP traces v_i sampled over a fixed time interval to build the AP models. The time interval is chosen to be sufficiently large for all variations of the AP traces to re-polarize to the resting state. Furthermore, for comparability, we align the resting state for all traces. After these pre-processing steps, we consider the resulting single-pulse AP traces v_i as T dimensional vectors in the sub-space $\mathcal{V} \subset \mathbb{R}^T$.

2.1.3 Pre-processing of the parameters

The parameters of the Beeler-Reuter model are of different units and magnitudes. For a stable numerical analysis, we scale the parameters to unitless quantities using the scaling constants in the third column in Table 1. Under this scaling, the parameter $p = (1, \dots, 1)$ corresponds to what we define as a baseline biophysical cell with its corresponding AP trace. This way, a 0.1 perturbation of any parameters corresponds to a 10% change in their magnitude, etc.

2.1.4 Selecting parameters for the drug effect identification

For the drug effect identification analysis, we select from the Beeler-Reuter model the parameters associated with the fast inward Na^+ current and the slow inward Ca^{2+}-type current. These are ionic currents typically examined when studying the anti- and pro-arrhythmic effects of cardiac drugs [38] and are the most natural choices among the parameters captured by the BR model. For the Na^+ current, since g_{Na} and g_{Nac} modulate the same current, we have combined these into the same parameter, g_{Na}, which effectively modulates the conductance of the Na^+ current. Similarly, g_s modulates the conductance of the Ca^{2+}-type current.

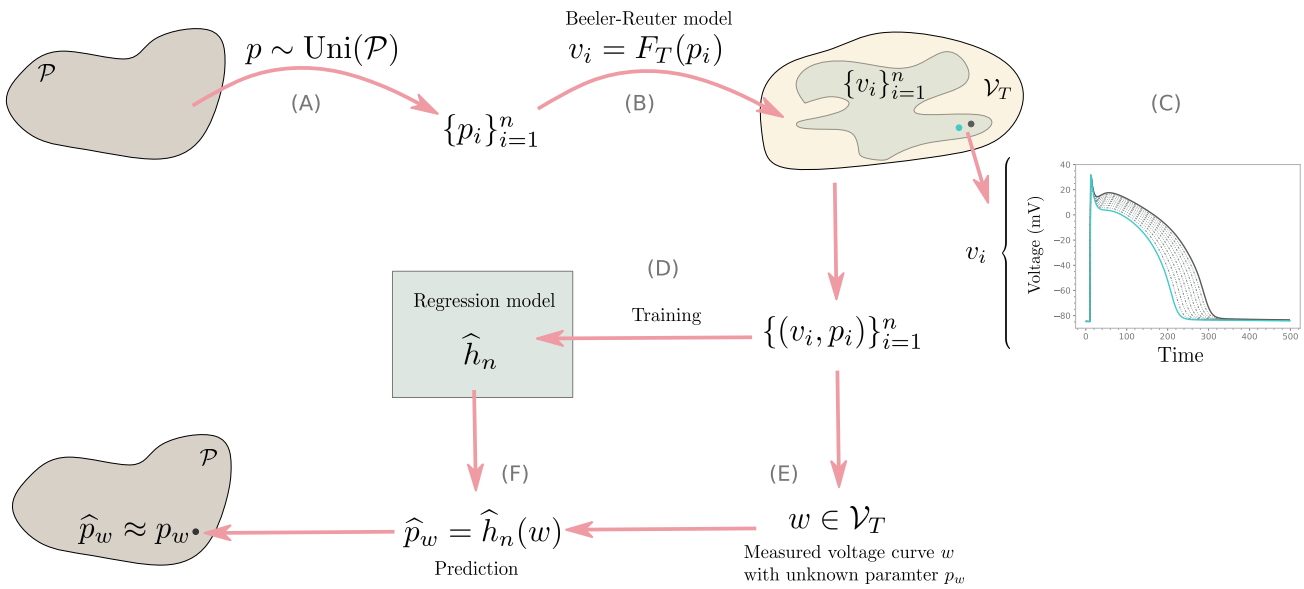


Figure 2: Illustration of our method

2.2 Constructing the kernel models

Having generated the training data $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$ as described in the previous section, we are in a position to construct our non-linear regression model. This modeling strategy corresponds to the rightmost column in panel (B) in Figure 1. We note that the modeling method proposed here is not restricted to training data generated from the Beeler-Reuter model. The training data $\mathcal{D}_n$ might also come from solving an alternative action potential model or from experimental data measured in the lab.

In this section, we outline the main principles of kernel methods as the underlying regression model we use. We then offer a brief sketch of the specific kernel model we propose, namely StreaMRAK [24], and an alternative kernel algorithm FALKON [32], which we compare with. For more details on these kernel methods, we refer to Appendix 6.5 and the references therein.

2.2.1 A note on the inverse map

We begin by making a short remark on the inverse map we set out to model. We are interested in constructing a non-linear regression model that can approximate the inverse map f^{-1}, illustrated in panel (A) in Figure 1. Since our training data is generated by the Beeler-Reuter model $v = F_T(p)$, we are, in practice, building a model that approximates $p = F_T^{-1}(v)$. In Appendix 6.3, we discuss the invertibility of this mapping in more detail. Furthermore, each of the parameters we consider p_j, $j \in 1, \ldots, d$ are independent (orthogonal), which means they can be described by independent mappings from the voltage space. We represent each of these mappings as

$$p_j = F_T^{-1}(v)(j) = f_T^{-1}(v; j). \tag{2.1}$$

2.2.2 Defining the model

Appendix 6.5 gives a more detailed description of the kernel model. Here we restrict ourselves to presenting the main building blocks. The model we propose to approximate $f_T^{-1}(v; j)$ corresponds to a

linear combination of kernels $k(v, v_i)$ centered on the training data:

$$\widehat{h}_n(v; j) = \sum_{i=1}^{n} \alpha_{ij} k(v, v_i), \tag{2.2}$$

where our choice of $k(v, v_i)$ is the Gaussian kernel

$$k(v, v_i) = \exp\left(- \frac{\|v - v_i\|^2}{2\sigma^2} \right).$$

The advantage of this model is that k is a universal kernel [27], meaning that $\widehat{h}_n$ can approximate bounded and continuous functions on a domain $\mathcal{X}$ arbitrarily well provided sufficient training data. Furthermore, since the model is non-parametric, we can capture highly complex functions without any background knowledge of the functional characteristics. We note that StreaMRAK introduces a multi-resolution formulation of the model in (2.2). Details on this are given in Appendix 6.5.

2.2.3 Learning the coefficients

Training the kernel model from Eq. (2.2) corresponds to finding an optimal set of regression coefficients α_{ij}. We will use two recently proposed algorithms for this purpose: The large-scale kernel method FALKON [32] and the multi-resolution adaptive kernel method StreaMRAK [24]. FALKON is rooted in the optimization scheme known as kernel ridge regression (KRR), see Eq. (6.1) in the appendix, but introduces several improvements to speed up computations. Meanwhile, StreaMRAK utilizes FALKON as a base solver but introduces a boosted version to construct a model that can learn highly complex functions with varying modalities; large local derivatives in some regions and derivatives near zero in others.

We mention that both FALKON and StreaMRAK use sub-sampling of the training data to reduce computational complexity. In FALKON, random sub-sampling, also known as Nystrom sub-sampling, is used to select a subset of training points on which the regression is performed. Meanwhile, StreaMRAK uses a cover tree to construct an epsilon cover in the AP space $\mathcal{V}$. The epsilon cover enforces a minimal distance of ϵ between each AP trace $v_i \in \mathcal{V}$ and also tries to maintain a maximal distance of ϵ. This has the advantage of removing training data that contribute with similar information (training points that are very close) but simultaneously ensuring that training points are close enough to maintain sufficient interpolation properties.

2.2.4 Parameter estimation

The trained model $\widetilde{h}_n$ can be used to estimate the underlying parameters of measured AP traces $w \in \mathcal{V}_T$ for which the corresponding set of parameters p_w is unknown. These AP traces can be, for example, time series measurements of the cardiac transmembrane potential using live cell fluorescence microscopy, as described in [5]. In this regard, it is crucial that the time grid underlying the training data is the same as for the AP traces w measured *in-vito*. If this is not the case, then the AP traces are not comparable. In other words, we must ensure the measured AP traces w belong to the same space $\mathcal{V}_T$ as the training data. Should the time grids differ initially, it is possible to align them by time interpolation, which is possible because the AP traces are smooth functions of time.

Provided $w \in \mathcal{V}_T$, we can make estimations of the corresponding parameters using the model $\widehat{h}_n$, such that $p_w = \widehat{h}_n(w)$. This is illustrated in step (E)-(F) in Figure 2. We note that for our synthetic experiments, we do not use AP traces measured experimentally. Instead, we generate a separate out-of-sample test set $\{w_i\}_{i=1}^n$ using the BR model on which we demonstrate our approach.

2.3 The nearest neighbor loss-minimization inversion schemes

For comparison with the regression strategy, we implement two variations of the nearest neighbor loss function minimization strategy; see the middle column in panel (B) in Figure 1. These strategies also make use of a precomputed training set $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$, and then aim to minimize a loss function $L(w, v_i)$ over $\mathcal{D}_n$, where $w \in \mathcal{V}_T$ is as in Section 2.2.4. Essentially, this corresponds to finding the nearest neighbor of w among the $v_i \in \mathcal{D}_n$, when "distance" is measured using L.

We consider two loss functions or "distance metrics" for finding the nearest neighbor. The simplest loss function is $L_{eucl}(w, v_i) = \sqrt{\sum_{k=1}^{T}(w_k - v_{ik})^2}$, which simply measures the euclidean distance between AP traces in $\mathcal{V}_T$. In addition, we use a loss function $L_{apf}(w, v_i) = \sum_j H_j(w, v_i)^2$ that measures the distance using specific action potential features (APF) H_j, such as AP duration. A complete list of APF and detailed descriptions can be found in S3.1 of [20].

2.4 Identifiability analysis

We are interested in learning the relationship between action potential traces and the parameters of the underlying ionic currents. However, as noted in [19], typical AP models have parameters that result in similar or overlapping changes, and the effect of perturbations along different directions in parameter space can vary considerably. Because of this, we expect that the effect of some parameters is more challenging to learn than others. By knowing which directions are expected to be hard to learn, one can account for that in the design of the learning model. Furthermore, knowing the parameter identifiability is also valuable when assessing the reliability of predictions.

In this section, we review an existing strategy, proposed in [19], for determining parameter identifiability based on spectral analysis of a matrix that incorporates the currents and their time dependency. Furthermore, we suggest a supplementary tool based on the Laplacian eigenmaps method [41], which offers a geometric understanding of the mapping between parameters and AP traces.

2.4.1 Spectral analysis

[19] proposed a tool for analyzing the identifiability of ionic currents in a given AP model using spectral analysis. Let I_j^k be membrane current j at time step $t_k = k\Delta t$ and form the matrix

$$A = \begin{pmatrix} I_1^1 & \dots & I_N^1 \\ \vdots & & \vdots \\ I_1^T & \dots & I_N^T \end{pmatrix}.$$

The total membrane current can then be written as $I_{tot} = A\mu$, where $\mu \in \mathbb{R}^N$ is the vector of only ones.

Under the assumption that the AP is determined by I_{tot}, the identifiability of each current component, I_j, can be determined by considering the singular value decomposition of A. In particular, I_{tot} and therefore the AP is only affected by the currents I_J that project onto the singular vectors associated with non-zero singular values. Because of this, any current in the null space of A can be considered non-identifiable.

Furthermore, as observed by [19], the change in AP when perturbing along a singular vector is proportional to the corresponding singular value. Consequently, currents projecting along singular vectors with small singular values have less effect on the AP, and changes in these currents are, therefore, harder to detect. Because of this, the size of singular values can be used to understand the identifiability of the membrane currents incorporated in an AP model. We note that the identifiability properties of different AP models vary as seen from [19, 20].

2.4.2 Geometrical analysis

We propose to use Laplacian eigenmaps (LE) [41, 42] as a geometrical tool to probe the relationship between AP traces and parameters. The LE is a well-established method for interrogating the structure of non-linear and complex point clouds and is supported by a solid theoretical foundation. In particular, LE can be considered a non-linear counterpart of principal component analysis.

We can think of the set of AP traces $\{v_i\}_i^n \in \mathcal{V}_T$ as a point cloud in $\mathcal{V}_T$, generated from a set of uniformly sampled parameters $p_i \in \Omega$ from some parameter domain $\Omega \in \mathcal{P}$. The main principle of LE is to construct the graph Laplacian L on $\{v_i\}_i^n$, which then incorporates the structure of the point cloud. To gain access to this structure, it is sufficient to solve for the singular vectors $\{u_i\}_{i=1}^n \in \mathbb{R}^n$ and singular values σ_i of L. In particular, with $\sigma_1 \leq \dots, \sigma_n \leq$, we have that u_1 corresponds to the direction of the most extensive spread. Furthermore, with r being the rank of L, only the r first singular vectors are relevant. For a detailed description of how to construct L, we refer to [41].

The similarity to the spectral analysis proposed in [19] is apparent, with the difference that we now consider the spectral components of the graph Laplacian L instead of the current matrix A. Furthermore,

using the r first singular vectors $\{u_i\}_{i=1}^r$ we can generate an r-dimensional embedding of $\{v_i\}_i^n$, where the singular vectors act as a new set of coordinates for the AP traces.

The AP traces $v_i \in \mathbb{R}^T$ are encoded as T dimensional vectors, where T is the number of time steps. Meanwhile, the rank r of L is directly related to the number of independent parameters that generated $\{v_i\}_i^n$. Although an AP model considers a significant amount of independent parameters, the number of independent parameters is typically significantly less than T, which means $r \ll T$. More importantly, restricting the parameters space to only a few parameters, $r \leq 3$, allows us to visualize the embedding.

2.5 Overview of study

In this study, we compare the drug-effect-estimation capabilities of the proposed kernel model StreaMRAK with four loss-minimization schemes similar to those used in [18, 20]. In particular, we consider two loss functions: the standard euclidean norm in the voltage space and a loss function constructed from action potential features (APF); see Section 2.3 for more details. For each loss function, we do a 1-nearest-neighbor search (Eucl-1-nn and Apf-1-nn) and a 10-nearest-neighbor search (Eucl-10-nn and Apf-10-nn). Furthermore, we also include the kernel model FALKON [32] in the comparison.

Estimating drug effects comes down to estimating the underlying model parameters of AP traces. In particular, to measure the effects of a specific drug dose, we need to estimate the model parameters of the untreated cell and the model parameters of the cell exposed to the drug, i.e., paired measurements and estimates. From these estimates, we can then quantify the drug effect by the perturbation magnitude and direction in the parameter space. Because of this, we will mainly focus on evaluating how the inversion schemes perform in terms of parameter estimations. However, it is implicit that these results translate directly to how the inversion schemes will perform in terms of drug-effect estimation.

In the next section, we define the measures used in this study to compare the different inversion schemes. The subsequent sections describe the synthetic training data and the experiments we conduct.

2.5.1 Measures of reliability

When exploring drug mechanisms in heart-on-chip systems, the reliability of parameter estimates is vital for making consistent and accurate estimates of the drug effects. In particular, an inversion scheme that provides high accuracy on average but has large variability over the parameter domain is undesirable because failure to estimate the effect of a particular drug dose can have severe consequences. When comparing the reliability of the inversion schemes, we, therefore, consider both the accuracy and how the accuracy varies over physiologically relevant regions of the parameter space. Let $\mathcal{A} \subset \mathcal{P}$ be such a physiologically relevant region in the parameter space. We then quantify the accuracy as the root mean square error (RMSE) of the parameter estimations in $\mathcal{A}$. Similarly, we quantify the variability in the estimate by the standard deviation (Std) and by the maximum error over $\mathcal{A}$.

2.5.2 Training data

Using the 1977 Beeler-Reuter AP model of mammalian ventricular cardiomyocytes, we generate a synthetic data set of $N = 6000$ sample pairs $\mathcal{D} = \{(v_i, p_i)\}_{i=1}^N$, on which we train our models. The source code used for this purpose is taken from [43]. The training samples are distributed uniformly over the parameter domain $\mathcal{P} = [0.2, 2]^2$ where the first parameter is the conductance of the rapid and excitatory inward sodium ionic current, g_{Na}, and the second parameter is the conductance of the slow inward ionic current, g_s, which primarily consists of calcium ions. Furthermore, we assume that $p_{(0)} = (g_{Na,0}, g_{s,0}) = (1.0, 1.0)$ are the parameters of the untreated cell, which represents a baseline biophysical cell and standard cardiac action potential.

2.5.3 Outline of experiments

Our experiments are designed to test the parameter estimation accuracy and variability over physiologically relevant regions to get insights into how the various inversion schemes will perform as tools for estimating drug effects.

The first experiment we conduct is covered in Section 3.1 and compares the inversion schemes as tools for drug-effect estimation by considering four simulated drugs $[A, B, C, D]$ whose perturbation directions θ are indicated in the right panel in Figure 3. For each drug, we consider a series of parameters

$p_{\theta,r} = (\theta, r)$ for varying radii r. From the root mean square and standard deviation over these parameter estimations, we quantify the expected accuracy and variability in the drug-effect estimation for each of the simulated drugs.

The three subsequent experiments are designed to better understand how the inversion schemes perform in terms of accuracy and consistency over the parameter domain. The first experiment, covered in Section 3.2, gives an overview by inspecting the heterogeneity of the estimation errors over the parameter domain $\mathcal{P}$. Meanwhile, in the two succeeding sections, Section 3.3 and 3.4, we present experiments that inspect the accuracy as a function of specific directions and distances away from the baseline biophysical cell at $p_{(0)}$.

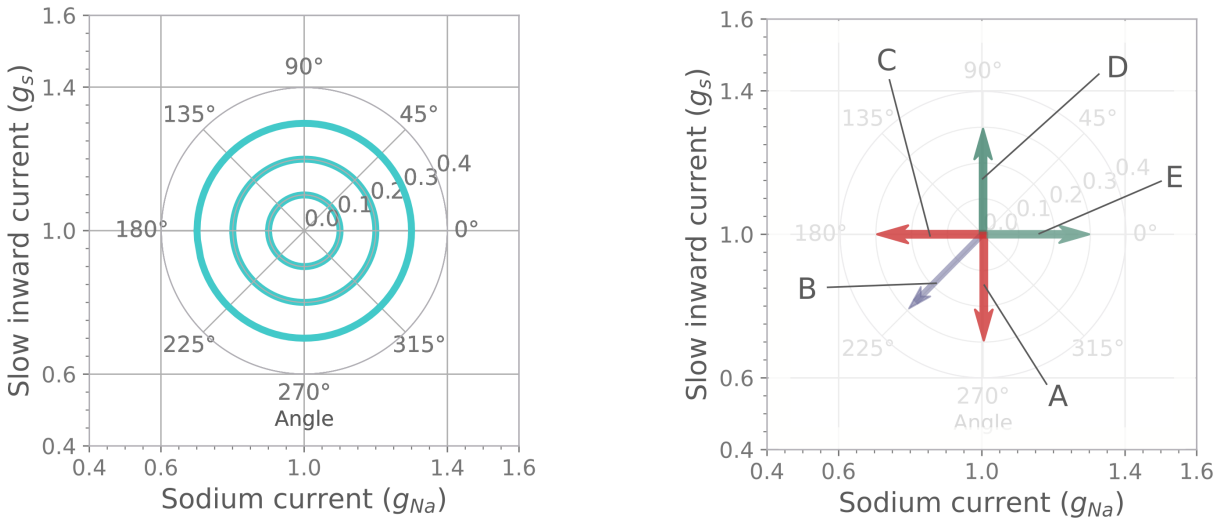


Figure 3: Direction and magnitude of perturbations in parameter space for simulated drug effects on the standard cardiac action potential. The left panel shows the three perturbation magnitudes $r = \{0.1, 0.2, 0.3\}$ for directions uniformly distributed on $[0°, 360°]$. The right panel indicates the drug effect of 5 different drugs; A and C are representatives of Ca^{2+} and Na^{+} current inhibitors, respectively; similarly, D and E are representatives of Ca^{2+} and Na^{+} current agonists, and B is a mix of a Ca^{2+} and a Na^{+} current inhibitors.

The left panel in Figure 3 shows three concentric circles in the parameter space. The experiment we conduct in Section 3.3 estimates the parameters from the corresponding AP traces along each of these circles, thereby examining how the estimation accuracy varies with the direction in parameter space measured from $p_{(0)}$. This experiment allows us to see if certain parameter combinations are more difficult to estimate than others and to what degree this is the case for the different inversion schemes.

Applying and measuring the response of a range of drug doses is an essential part of drug development. It is, therefore, of interest to know how the accuracy of the inversion schemes varies with drug dosage. Increasing the dose of specific drugs results in perturbations in the parameter space with increasing magnitude away from the baseline biophysical cell. The experiment we conduct in Section 3.4 examines this by studying how the accuracy varies with the distance from $p_{(0)}$ along the drug directions A, B, and C, as illustrated in the right panel in Figure 3. This experiment allows us to compare the reliability of the inversion schemes, both in terms of estimation accuracy and in terms of consistency, along relevant physiological regions of the parameter space.

We finish the study with a performance analysis in Section 3.5, where we consider the time complexity and parameter estimation capability as a function of model size.

An implementation of StreaMRAK and FALKON, along with experiments that reproduce the performance experiments can be found in the gitHub repository [44].

3 Results

This section compares the parameter estimation capabilities of StreaMRAK with FALKON and the loss-minimization schemes. We consider the experiments outlined in Section 2.5.3 and finish the section with

a performance analysis on time complexity and parameter estimation capability.

3.1 Experiment one: Identifying the effect of simulated drugs

To demonstrate the use of StreaMRAK as a method for drug-effect estimation, we consider four simulated drugs $[A, B, C, D]$ whose perturbation directions are indicated in Figure 3. Let $p_{\theta,r} = (\theta, r)$ be the polar representation of a parameter, centered at $p_{(0)}$. For each drug, we run 100 experiments for a fixed angle θ and sample the perturbation radius $r \sim \mathcal{N}(0.2, 1 \times 10^{-2})$, namely a 20% perturbation with standard deviation ±5%.

Table 2: Estimation results of the algorithms w.r.t. the drug effects of the Ca^{2+} blocker, drug A, and the Ca^{2+} enhancer, drug D. The "Pred" column is the predicted parameters (p_1, p_2), while "RMSE" is the root mean square error of the estimation and "Std" is the standard deviation.

	Calcium blocker (A)			**Calcium enhancer (D)**		
Perturbation	(1.0, 0.8)	–	–	(1, 1.2)	–	–
Algorithms	Pred	RMSE	Std	Pred	RMSE	Std
Streamrak	(0.996, 0.800)	(4.e-3, 4.e-4)	(4.e-3, 7.e-4)	(0.994, 1.201)	(7.e-3, 7.e-4)	(7.e-3, 1.e-3)
Falkon	(1.02, 0.798)	(2.e-2, 2.e-3)	(2.e-2, 4.e-3)	(1.017, 1.197)	(2.e-2, 2.e-3)	(2.e-2, 3.e-3)
Eucl-1-nn	(1.028, 0.799)	(2.e-2, 2.e-3)	(2.e-2, 3.e-3)	(1.026, 1.194)	(4.e-2, 6.e-3)	(4.e-2, 6.e-3)
Eucl-10-nn	(1.043, 0.797)	(2.e-2, 2.e-3)	(3.e-2, 4.e-3)	(1.034, 1.194)	(5.e-2, 4.e-3)	(5.e-2, 7.e-3)
Apf-1-nn	(1.026, 0.811)	(3.e-2, 1.e-2)	(3.e-2, 9.e-3)	(1.026, 1.218)	(5.e-2, 3.e-2)	(5.e-2, 3.e-2)
Apf-10-nn	(0.966, 0.793)	(5.e-2, 1.e-2)	(5.e-2, 1.e-2)	(0.984, 1.190)	(3.e-2, 2.e-2)	(4.e-2, 2.e-2)

Table 3: Estimation results of the algorithms w.r.t. the drug effects of the Na^+ blocker, drug C, and the mixed current blocker, drug B. For column descriptions, see Table 2.

	Sodium blocker (C)			**Mixed blocker (B)**		
Perturbation	(0.8, 1.0)	–	–	(0.859, 0.859)	–	–
Algorithms	Pred	RMSE	STD	Pred	RMSE	Std
Streamrak	(0.798, 1.000)	(5.e-3, 4.e-4)	(5.e-3, 6.e-4)	(0.863, 0.858)	(5.e-3, 4.e-4)	(5.e-3, 6.e-4)
Falkon	(0.813, 0.997)	(1.e-2, 3.e-3)	(1.e-2, 2.e-3)	(0.885, 0.856)	(3.e-2, 3.e-3)	(2.e-2, 4.e-3)
Eucl-1-nn	(0.807, 0.997)	(1.e-2, 2.e-3)	(1.e-2, 3.e-3)	(0.828, 0.860)	(3.e-2, 2.e-3)	(3.e-2, 2.e-3)
Eucl-10-nn	(0.831, 0.994)	(3.e-2, 5.e-3)	(3.e-2, 8.e-3)	(0.884, 0.855)	(3.e-2, 2.e-3)	(3.e-2, 4.e-3)
Apf-1-nn	(0.772, 0.988)	(2.e-2, 1.e-2)	(2.e-2, 8.e-3)	(0.844, 0.852)	(3.e-2, 1.e-2)	(3.e-2, 8.e-3)
Apf-10-nn	(0.819, 1.006)	(2.e-2, 7.e-3)	(2.e-2, 7.e-3)	(0.908, 0.874)	(4.e-2, 1.e-2)	(4.e-2, 1.e-2)

All drugs considered in this experiment correspond to a perturbation from the unperturbed state $p_0 = (g_{Na}, g_s) = (1.0, 1.0)$, but in different directions. Drug A is a Ca^{2+}-type current inhibitor (such as verapamil), drug C is a Na^+ current inhibitor (such as quinidine), drug D is a Ca^{2+}-type current agonist (such as BAYK8644), drug E is a Na^+ current agonist (such as veratridine), and drug B is a mix of drugs A and C (such as a combination of quinidine and verapamil, or flecainide); [45]; [46]; [47]; [48]; [49]; [50].

The estimation results for each of these drugs are summarised in Table 2 and Table 3. The root mean square error (RMSE) and the standard deviation (Std) are calculated over the 100 experiments. Meanwhile, the Pred column shows estimates of a specific perturbation whose magnitude is shown in the first row. Again, StreaMRAK outperforms the other methods with lower RMSE for all drugs. The estimation of the Na^+ current parameter is consistently worse than the parameter for the slow inward current across all algorithms.

The Beeler-Reuter model is used to generate the AP traces corresponding to the predicted parameters. This provides an idea of which part of the AP gives rise to the estimation error. Figure 4 depicts how the AP traces relate to the trace of the actual perturbation (solid blue line). The AP trace corresponding to the unperturbed parameters is included for reference. There are, in particular, two regions where a discrepancy is present between the predicted and actual perturbations. These are the peak

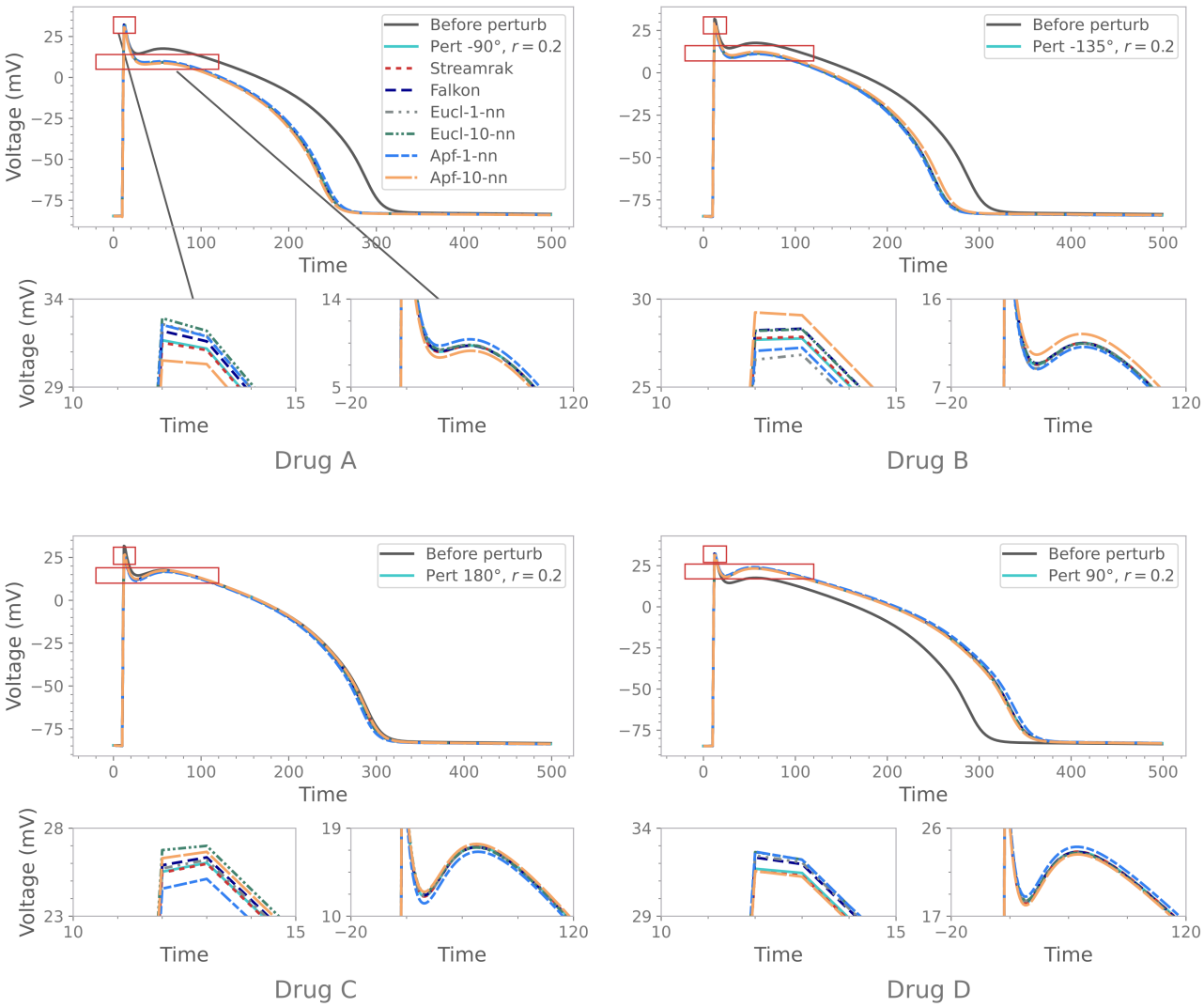


Figure 4: Comparison of how well the algorithms predict the drug effects of the drugs A, B, C, D from Figure 3. In each sub-figure, the upper panel shows the perturbation and the estimation from each algorithm. The unperturbed trace is included for reference. The lower panels highlight the first and the second "peak" in the AP trace.

of the depolarization, mainly determined by the rapid Na^+ current and the plateau before the final repolarization phase.

The Na^+ inhibitor (Drug C) mainly affects the depolarization peak as seen from panel (Drug C) in Figure 4. Furthermore, the close-up in this panel shows that StreaMRAK gives a better fit to this peak compared to the other methods. Meanwhile, the Ca^{2+}-type drugs A and D mainly affect the plateau and the repolarization phases, and these changes are more pronounced than the change in the depolarization peak. From the panels of drugs A and D, one can see that all methods give a close fit to the plateau and the repolarization phases. However, the close-up reveals that StreaMRAK has the tightest fit. Furthermore, the approximation of the depolarization peak is again better for StreaMRAK.

3.2 Experiment two: Domain error

This experiment studies the heterogeneity of the estimation error over the parameter domain $\mathcal{P} = [0.2, 0.2]^2$ to quantify the reliability of parameter estimations from StreaMRAK with respect to the other methods. The reliability over a region is related to both the absolute accuracy of the estimate and how the estimation accuracy varies within this region, with higher variations meaning lower reliability.

To generate the test data for this experiment, 3000 points are sampled uniformly from $\mathcal{P}$. The corresponding voltage curves are then solved for using the Beeler-Reuter model. For each test trace, w_i, the underlying parameter vector $p_{w_i} \in \mathcal{P}$ is estimated and compared with the actual parameter p_i

from the test set. In this manner, the square error is calculated for each sample in the test data; the distribution of errors over $\mathcal{P}$ are shown in Figure 5.

Panel (A)-(B) in figure 5 shows the domain error of the StreaMRAK and FALKON estimations. Both StreaMRAK and FALKON are based on training a non-linear regression model, which interpolates between the samples in the training data $\mathcal{D}$. However, the estimation errors of StreaMRAK are more consistently low compared to FALKON. In particular, FALKON has higher variability and more significant MSE along the domain borders than StreaMRAK.

Panel (C)-(F) shows the domain error of the four loss-minimization algorithms Eucl-1-nn, Eucl-10-nn, Apf-1-nn, and Apf-10-nn. The domain errors of the APF-loss and Euclidean-loss-based methods are relatively comparable; the domain errors of the APF-based methods are slightly larger in magnitude than those based on the euclidean loss function.

By comparing the domain error of the kernel methods (A)-(B) with that of the loss-minimization schemes (C)-(F), one can see the difference between building a regression model of the data, which extends estimations beyond the training samples in $\mathcal{D}_n$, and minimizing a loss, which is restricted to samples in $\mathcal{D}_n$. The model-based schemes StreaMRAK and FALKON have significantly more consistent estimations over the parameter domain $\mathcal{P}$. Furthermore, the domain error of StreaMRAK is also considerably lower, whereas, for FALKON, this is only true in some local regions of the parameter domain.

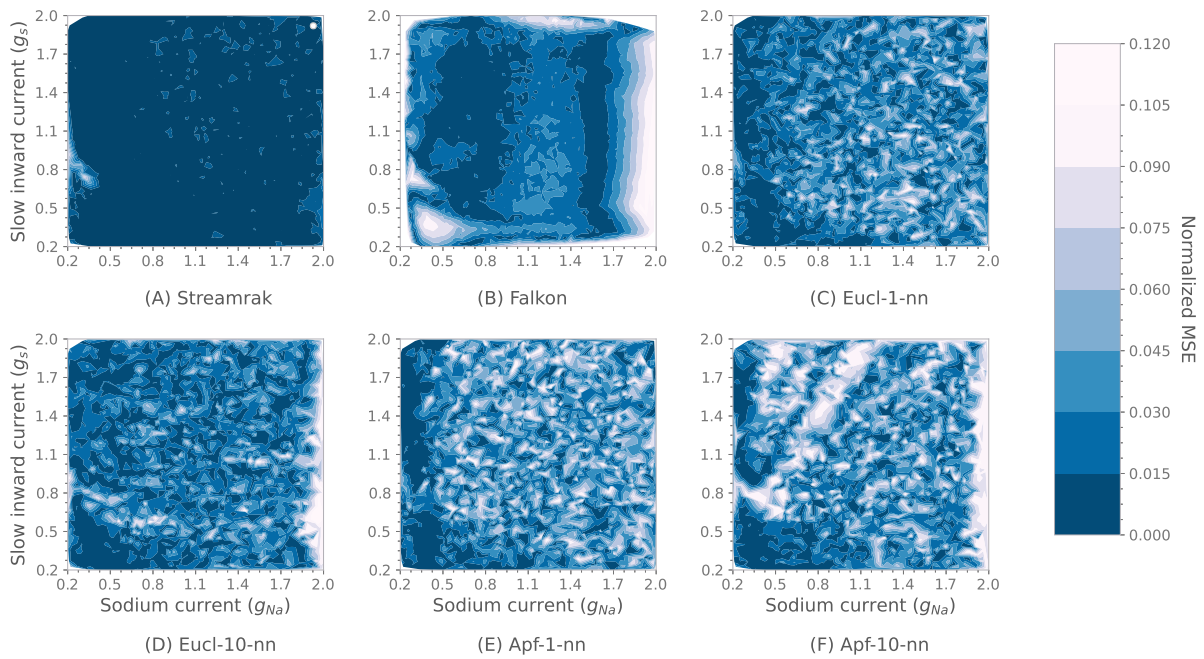


Figure 5: Experiment from Section 3.2. Distribution of estimation error over the domain $P = [0.2, 2]^2$.

3.3 Experiment three: Accuracy as a function of perturbation direction

This experiment studies how the accuracy of the inversion methods varies as a function of direction in parameter space relative to the baseline biophysical cell at $p_{(0)}$.

To generate the test data, parameters are sampled uniformly along three concentric circles centered at $p_{(0)} = (1.0, 1.0)$ with radius $0.1, 0.2$, and 0.3, respectively. For each set of parameters $(g_{Na,i}, g_{s,i})$ from these perturbation circles, the corresponding AP trace v_i is generated using the Beeler-Reuter model. The underlying parameters of these AP traces are then estimated using each of the six algorithms. By assuming that $p_{(0)}$ corresponds to the untreated case, the three circles can be considered as drug perturbations of $10\%, 20\%$, and 30% in all directions in the parameter domain as shown in the left panel in Figure 3.

The parameter estimates from the six algorithms are shown in Panel (A)-(C) in Figure 6 for each of the three perturbation magnitudes, respectively. The accuracy of the parameter estimation from the two kernel methods, StreaMRAK and FALKON, is consistently high throughout the perturbation directions

Figure 6: Estimation of perturbations with increasing magnitude for directions in interval $[0°, 360°]$. Here panel (A) corresponds to perturbation radius $r = 0.1$ from the unperturbed $p = (1.0, 1.0)$ (Smallest perturbation circle in Figure 3). Panel (B) corresponds to perturbation radius $r = 0.2$ (Middle perturbation circle in Figure 3). Panel (C) corresponds to perturbation radius $r = 0.3$ (Largest perturbation circle in Figure 3).

and magnitudes. Meanwhile, the accuracy of the estimations from the euclidean-loss-minimization algorithms Eucl-1-nn, Eucl-10-nn is high around the 90° and 270° axes, which are directions dominated by perturbations of the slow inward (Ca^{2+}-type) current. However, when perturbations of the Na^+ current are introduced, the parameter estimation accuracy of these methods drops. We observe similar behavior for the APF-loss-minimization algorithms Apf-1-nn and Apf-10-nn; however, the regions with the best estimation accuracy are slightly shifted away from the vertical axes.

We have observed that the estimation accuracy of the loss-minimization algorithms drops as the magnitude of the perturbation of the Na^+ current parameter g_{Na} increases. Furthermore, perturbations of g_{Na} affect the estimation accuracy of g_s. Meanwhile, the regression models exhibit more confidence in identifying parameters. Notably, for StreaMRAK and FALKON, the estimation accuracy of each of the parameters g_s, g_{Na} does not depend on the perturbation direction and magnitude.

3.4 Experiment four: Accuracy as a function of perturbation magnitude

This experiment studies how prediction accuracy varies with the perturbation magnitude away from the baseline biophysical cell at $p_{(0)}$. For this purpose, the three-drug directions A, B, and C introduced in Section 3.1 are considered; See Figure 3. To generate the test data, 20 evenly spaced samples are made on the interval $[0, 0.3]$ along each of the three simulated drug directions. The estimation results are shown in Figure 7.

Panel (A) in Figure 7 shows the estimation error of the g_{Na} parameter. The parameter estimation accuracy of StreaMRAK and FALKON are significantly more consistent across the perturbation range $[0, 0.3]$ for each of the three directions. To quantify this, we refer to Table 4 and Table 5 in the supplementary, which shows the three metrics (RMSE, Max abs. error, Standard deviation) over the interval $[0, 0.3]$.

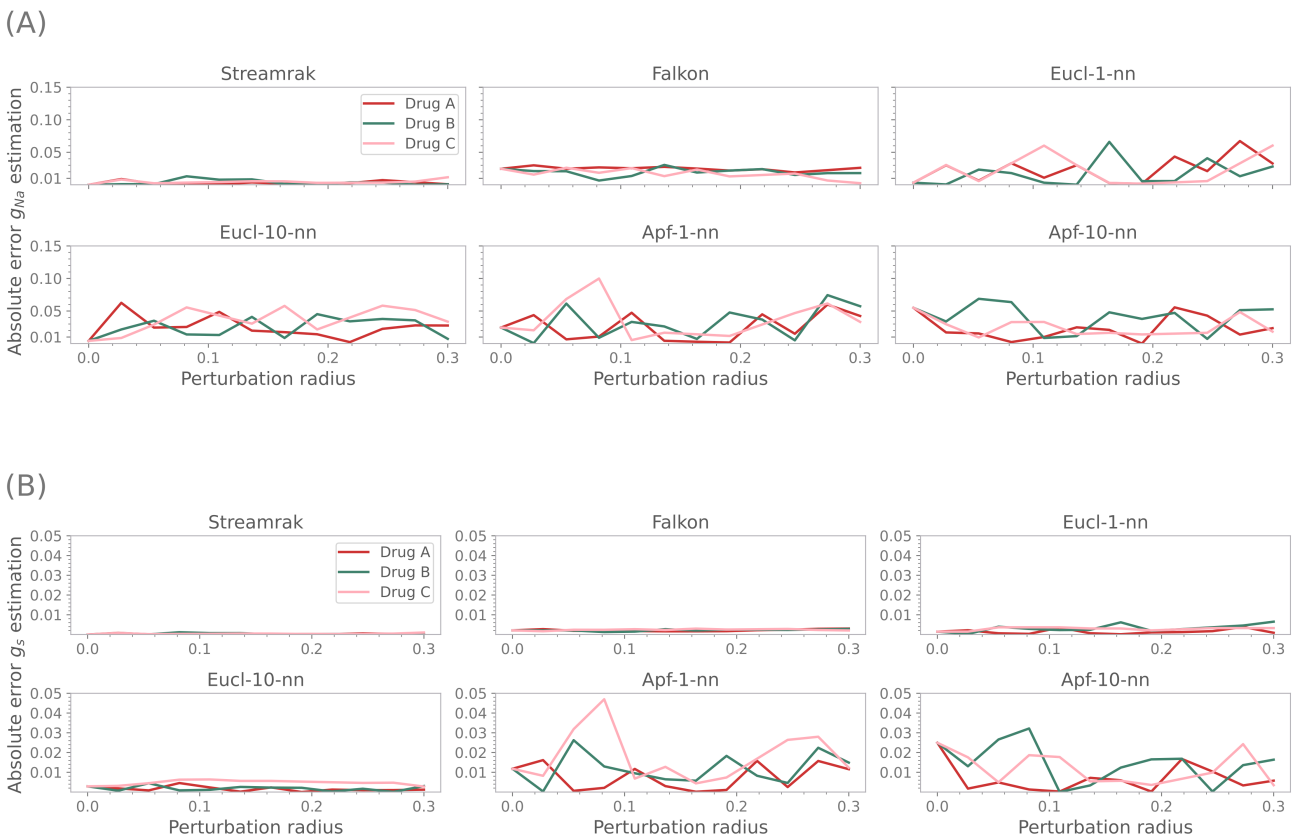


Figure 7: Experiment from Section 3.4. Figure showing parameter estimation accuracy, as measured by the absolute error, along the three selected directions A, B, and C as indicated in Figure 3. For each direction, 20 samples are evenly spaced on the interval $[0, 0.3]$. The red line is direction A, the green line is B, and the pink line is C. Panel (A) shows the estimation accuracy of the g_{Na} parameter, and panel (B) shows the estimation accuracy of the g_s parameter.

The same analysis is done for the g_s parameter, where we see an improvement in the performance of Eucl-1-nn and Eucl-10-nn. Meanwhile, Apf-1-nn and Apf-10-nn, based on the APF-loss from [20], have large fluctuations in estimation accuracy also for g_s. Again, using the three metrics (RMSE, Max abs. error, Standard deviation), we quantify the difference in reliability; see Table 4 and Table 5 in the supplementary.

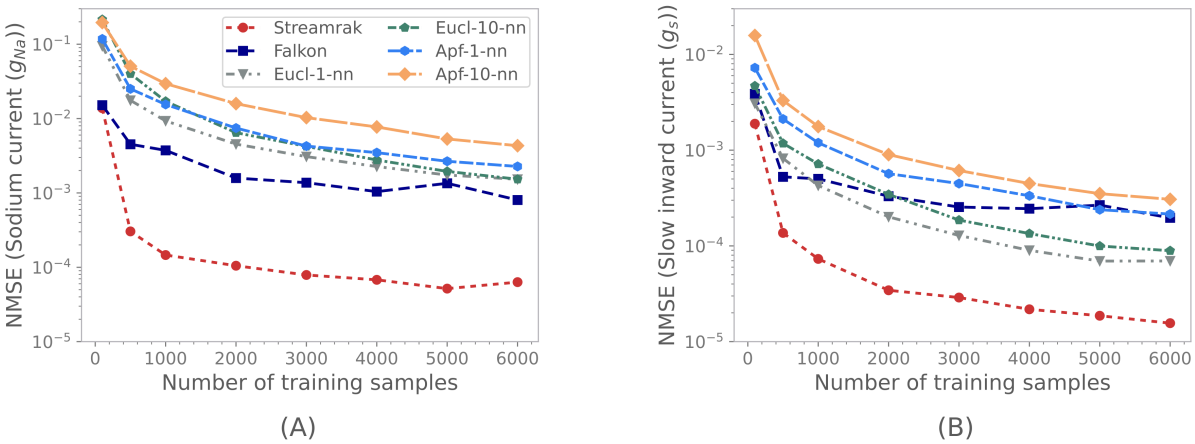
Finally, for all methods, the error and variability are greater for the g_{Na} parameter than for the g_s parameter.

3.5 Performance analysis

To compare the overall performance and scalability of the methods, their parameter estimation capability and their time complexity are analyzed as a function of the training data size. The MSE is taken over all test estimations for each parameter to quantify the parameter estimation capability. The test data is the same as in Section 3.2 with 3000 points sampled uniformly from $\mathcal{P} = [0.2, 2]^2$. Meanwhile, for the training data, a model is constructed on each of the sample sizes $n \in [100, 500, 1000, \ldots, 6000]$.

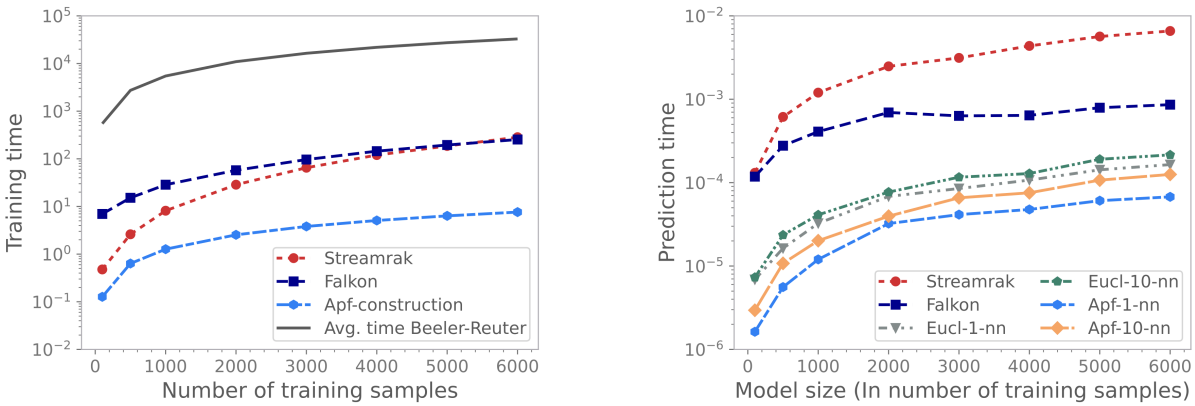
Consider the MSE over the test estimations for each parameter; the result is shown in Figure 8, where panel (A) is the MSE of the g_{Na} estimations and panel (B) the MSE of the g_s estimations. For both parameters, as the number of training samples increases, the MSE is reduced across all methods, and the error from StreaMRAK is consistently an order of magnitude lower than the rest. Furthermore, we see how StreaMRAK achieves a low estimation error with relatively small training sample sizes (2000 - 3000 samples). Meanwhile, the other methods do not reach a similar level of MSE even when doubling the sample size.

Figure 9 (A) compares the training time usage of the algorithms as a function of the training sample size. A notable advantage of the loss-minimization algorithms Eucl-1-nn, Eucl-10-nn, Apf-1-nn, and Apf-



(A) (B)

Figure 8: MSE of the estimations over the parameter domain $P = [0.2, 2]^2$. Panel (A) shows the MSE of estimations of the g_{Na} parameter (Na^+ current). Panel (B) shows the MSE of estimations of the g_s parameter (Ca^{2+}-like current).



(A) (B)

Figure 9: Comparison of time complexity of the algorithms. Panel (A) shows the training time of Streamrak and Falkon along with the time to construct the action potential features for the apf-k-nn algorithm. The solid dark line is the average time to solve for the AP traces of N samples using the Beeler-Reuter ODEs. Panel (B) shows the average estimation time, averaged over 2897 test samples.

10-nn is that they do not require any training time because they only rely on "memorizing" the training data. Although we need to consider the time required to construct the action potential features for each AP trace for Apf-1-nn, and Apf-10-nn, it is clear that the training time of StreaMRAK and FALKON is significant in comparison. However, as seen in Figure 8, StreaMRAK achieves a lower estimation error with far fewer samples. Because of this, the cost of generating the training samples must also be taken into account. This cost is indicated as the solid black line in Figure 9 (A), and it is clear that it is several magnitudes higher than the training times.

Figure 9 (B), compares the average prediction (parameter estimation) time of each method as the model size (number of training samples) increases. Both StreaMRAK and FALKON are slower than the loss-minimization schemes.

4 Discussion

In Section 3 we compared two regression-based learning models StreaMRAK and FALKON with four nearest-neighbor loss-minimization schemes Eucl-1-nn, Eucl-10-nn, Apf-1-nn, and Apf-10-nn. Our ex-

periments show that StreaMRAK estimates drug effects with higher accuracy — roughly an order of magnitude higher — than both FALKON and the loss-minimization schemes. Furthermore, we have seen that the accuracy of parameter estimates of the regression models is more consistent across the studied parameter combinations than the loss-minimization schemes suggesting a greater degree of reliability for the kernel methods, both in terms of accuracy and consistency.

In the remainder of this section, we will discuss the results from Section 3 in more detail. In particular, we will focus on two key aspects: Reliability, which we discuss in Section 4.1, and Computational performance, which we discuss in Section 4.2. We end the discussion with a note on the identifiability of currents in the Beeler-Reuter model; see Section 4.3.

4.1 Accuracy and reliability of methods

In this section, we take a closer look at how the estimation accuracy of the methods compares across different combinations of parameters. From Figure 5, we see a first indication that StreaMRAK, and to some degree FALKON, have a more consistent estimation accuracy than the loss-minimization schemes. Furthermore, from Panel (A)-(C) in Figure 6, we see that parameters along specific directions in parameter space are more difficult to estimate. In particular, this is especially true for the loss-minimization schemes Eucl-1-nn, Eucl-10-nn, Apf-1-nn, and Apf-10-nn.

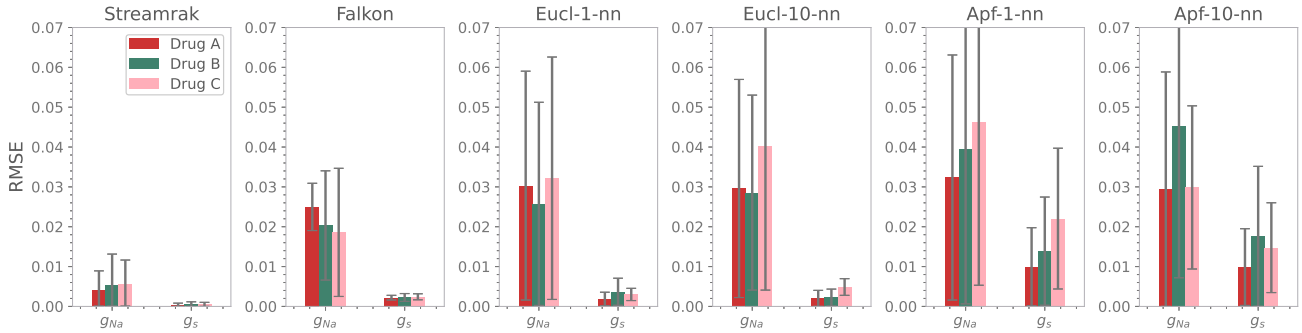


Figure 10: Figure showing the RMSE with error bars of the parameter estimations along three different perturbation directions on the interval $[0, 0.3]$. The perturbation directions correspond to the directions of Drugs A (red), B (green), and C (pink) defined in Figure 3.

We take a closer look at this observation by considering statistics of the experiment presented in Figure 7 in Section 3.4. We summarise these statistics in Figure 10, which shows the mean of the absolute error and the standard deviation for perturbation radii sampled uniformly on $[0, 0.3]$ for each of the three representative drugs A, B, and C. Consider first Panel (A) in Figure 10. Our first observation is that the error of StreaMRAK is significantly smaller than for the other methods both for the g_{Na} and the g_s estimations. More importantly, we observe that the estimation error for each of the three drugs is more consistent for StreaMRAK than for the other methods. Furthermore, we also see that the standard deviation within a specific drug is more significant for FALKON and the loss-minimization schemes compared to StreaMRAK. The consistency that StreaMRAK exhibits is valuable for the purpose of identifying drug effects for different populations of cultured cardiac cells and for different types of drugs. For both estimates of the baseline model parameterization and the perturbation under the effects of a simulated drug, StreaMRAK performed the best regarding the absolute prediction accuracy and the consistency of the estimate on repeated tests.

We also note that the standard deviation in the g_{Na} estimations is larger than for the g_s parameter, especially for FALKON and the loss-minimization schemes. In Section 4.3 we show how this can be attributed to a difference in identifiability of the two parameters, where g_{Na} has the lower identifiability.

4.2 Computational considerations

Although the regression models StreaMRAK and FALKON have higher accuracy and are more consistent in their parameter estimation, these regression models come with an upfront training cost, as opposed to the loss-minimization schemes. Theoretically, this cost is $\mathcal{O}(Ln\sqrt{n})$ for StreaMRAK [24], where L is

the number of resolution levels used in the model, and $\mathcal{O}(n\sqrt{n})$ for FALKON [32]. For the APF-based methods, there is admittedly a cost associated with computing the action potential features for each AP trace. However, as seen in Panel (A) in Figure 9, this cost is significantly less than the training time of both StreaMRAK and FALKON.

Nevertheless, from panel (A) in Figure 8, we see that StreaMRAK, and to some degree, FALKON, requires significantly fewer training points to achieve a small MSE. In fact, from Panel (A) in Figure 13 (See the Supplementary), we see the MSE as the number of training points is extended to $n = 60000$. Even then, for the g_{Na} parameter estimation, StreaMRAK has almost an order of magnitude lower MSE. Meanwhile, for the g_s parameter estimation, Eucl-1-nn and Eucl-10-nn catch up only after roughly $n = 35000$ training samples. This is somewhat expected because, with a high enough density of samples, any Voronoi cell (region closest to a point in a set than any other point in the set) will be approximately linear and small, which means minimizing the euclidean loss with the center of the Voronoi cell should have high accuracy. That StreaMRAK can do more for less is essential when we also consider the time it takes to construct the training samples by solving the AP model (Beeler-Reuter model). From Panel (A) in Figure 9, we see that solving the AP model has considerable time complexity. Furthermore, this time complexity is expected to increase for larger models. Consequently, the up-front training time is balanced by the reduced number of training samples for which the AP model must be solved.

Since the loss-minimization algorithms are based on minimizing a function over the training data, they must store all the training data, as the data is essentially their model. Therefore, the memory required for these algorithms is $\mathcal{O}(n)$. Furthermore, these models' precision (resolution) is directly linked to the density of training samples. Consequently, because the number of samples necessary to cover a d-dimensional region grows exponentially with the dimension, the training set $\mathcal{D}_n$ will need to be very large to make high-accuracy estimates for models with a large number of parameters.

The regression models, on the other hand, need the training data up-front to train the models. But once trained, significantly less memory is required for storage. For StreaMRAK this memory requirement is $\mathcal{O}(L\sqrt{n})$, while for FALKON it is $\mathcal{O}(\sqrt{n})$. Smaller memory translates to easier transfer and implementation due to lower memory requirements for the host computer system. Furthermore, because these methods interpolate between the training samples, fewer training samples are required to achieve good estimates, as observed in panel (A) in Figure 8.

In terms of parameter estimation time (prediction time), we see from Panel (B) in Figure 9 that the regression-based methods are slower, which is consistent with analytical estimation time calculations. For StreaMRAK the analytical estimation time is $\mathcal{O}(L\sqrt{n})$, while for FALKON it is $\mathcal{O}(\sqrt{n})$. Meanwhile, using a KD-tree, the nearest-neighbor schemes have an estimation time of $\mathcal{O}(d \log n)$ [51] (Using brute force, it is $\mathcal{O}(dn)$). Here d is the intrinsic dimensionality of the AP traces (which is strongly related to the number of independent parameters in the model).

4.3 A note on the identifiability of currents in the Beeler-Reuter model

We use the identifiability analysis techniques discussed in Section 2.4 to gain a deeper understanding of the results from Section 3. In particular, we are interested in why estimates of the g_s parameter have higher accuracy than the g_{Na} parameter. Furthermore, we want to understand why the consistency of the g_{Na} estimations is lower than for the g_s parameter and why this is especially true for the loss-minimization methods.

The Beeler-Reuter model incorporates three currents of interest: the sodium, slow-inward, and time-dependent-outward current, whereof the first two are parameterized by the g_{Na} and g_s parameters, respectively. Using the spectral analysis from [19], as described in Section 2.4.1, we can analyze the identifiability of these currents in the Beeler-Reuter model. Let $e = (1, 0, 0)$ represent the sodium current, $e = (0, 1, 0)$ the slow inward current, and $e = (0, 0, 1)$ the time-dependent outward current. From the analysis, we find the singular values $\sigma = (\sigma_1, \sigma_2, \sigma_3) = (707.8, 6.9, 8.1 \times 10^{-2})$ with singular vectors $v_1 = (7.956 \times 10^{-5}, 9.989 \times 10^{-1}, -9.917 \times 10^{-4})$, $v_2 = (9.997 \times 10^{-1}, -7.927 \times 10^{-5}, 2.226 \times 10^{-4})$ and $v_3 = (-2.268 \times 10^{-4}, 9.919 \times 10^{-4}, 9.987 \times 10^{-1})$. This means the current parameterized by g_s projects mainly along the largest eigenvalue direction, while g_{Na} is projected mainly along v_2 with singular value $\sigma_2 \ll \sigma_1$. Meanwhile, the time-dependent outward current is projected primarily along v_3 for which the eigenvalue is negligibly small. [19] observed that the change in the AP traces when perturbing along a singular vector is proportional to the corresponding singular value. Consequently, this analysis shows that for the Beeler-Reuter model, we can expect less identifiability for the g_{Na} parameter than for the

g_s parameter, which helps explain why the accuracy in estimating g_{Na} is lower than for g_s.

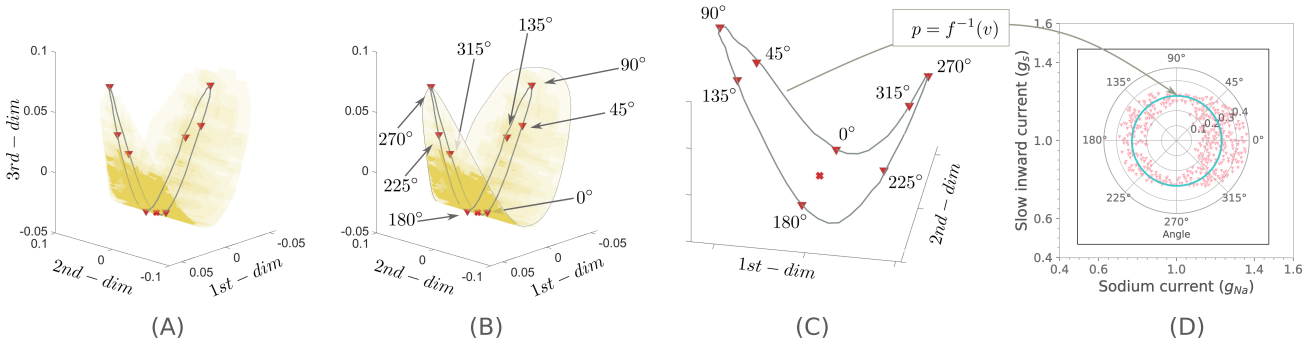


Figure 11: Panel (A) shows the 3-dimensional projected embedding of a ball in the space of action potential curves V (Embedding using the Laplacian Eigenmaps algorithm [41]). The grey ellipse in (A) corresponds to a circle of radius $r = 0.1$ centered at $p = (1, 1)$ in the parameter space. Panel (B) indicates how the directions in the parameter space map to the embedding. Panel (C) and (D) illustrate the relationship between a perturbation ring of radius $r = 0.3$ in parameter space (light blue circle in panel(D)) and the corresponding ellipse in V (grey ellipse in panel (C)). The pink triangles correspond to voltage curve samples from a band around the ellipse in panel (C) that are mapped back into the parameter space.

To supplement this analysis, we consider the geometrical analysis proposed in Section 2.4.2. We let the parameter domain Ω be the ball $\mathcal{B}(p_0, \delta) \in \mathcal{P}$ centered at $p_0 = (1, 1)$ with radius $\delta = 0.2$. Panel (D) in Figure 11 illustrates this ball. The corresponding AP traces lies on a 2-dimensional surface in $\mathcal{V}_T \in \mathbb{R}^T$, which we embed in 3 dimensions using Laplacian eigenmaps as shown by the yellow surface in Panel (A) in Figure 11. The grey ellipse in panel (A) indicates the embedded AP traces corresponding to the parameters on the blue circle in panel (D).

From Section 6.4 in the appendix, it is clear that the shape of the ellipse directly relates to the derivative of the inverse map f^{-1}. Along the g_{Na} axis, the ellipse is narrow, and therefore the derivative is large. Meanwhile, along the g_s axis, the derivative of f^{-1} is comparatively small. Clearly, high accuracy in voltage space is particularly vital in perturbation directions where small changes in AP-trace space significantly affect the underlying parameters. To illustrate this, consider AP traces from a narrow band along the ellipse. The corresponding parameters are shown as pink triangles in Panel (D). What is apparent is that, along the g_{Na} axis, the spread in parameters is considerable compared with the spread along the g_s axis. Consequently, because minor errors when matching the AP traces lead to substantial changes in the estimates of g_{Na}, we can expect the variability in the g_{Na} estimations to be more significant than for the g_s estimations.

The difficulty of learning the effect of the g_{Na} parameter is illustrated in Figure 12, which shows the AP traces as they vary when the parameters are perturbed along different directions from the initial $p_0 = (1, 1)$. Clearly, the g_{Na} parameter only leads to subtle changes in the AP traces, primarily affecting the depolarization peak. Consequently, if a learning model does not capture these subtle differences, it will lead to reduced estimation accuracy, which is precisely what we observe for the g_{Na} parameter.

For example, we consider the drug estimation in Figure 4, where we see that StreaMRAK is better at capturing critical features of the AP traces. For instance, a key identifier of Na^+ current inhibitors (such as quinidine) is the peak of depolarization. This drug effect is simulated by both drugs C and B, and we see from Figure 4 that for both drugs, StreaMRAK succeeds in capturing the depolarization peak, while the other methods fail to do so. Consequently, a significant error is observed in these methods' g_{Na} estimation by failing to capture this peak.

4.4 Complementary studies

By introducing a kernel-based regression scheme, our study improves the inversion step in the computationally aided drug-effect-estimation pipeline proposed by [18]. In particular, using synthetic and experimental data, [18] demonstrates how optically measured AP traces from human induced pluripotent stem cell-derived cardiomyocytes (hiPSC-CMs) can be inverted to gain information on drug effects in

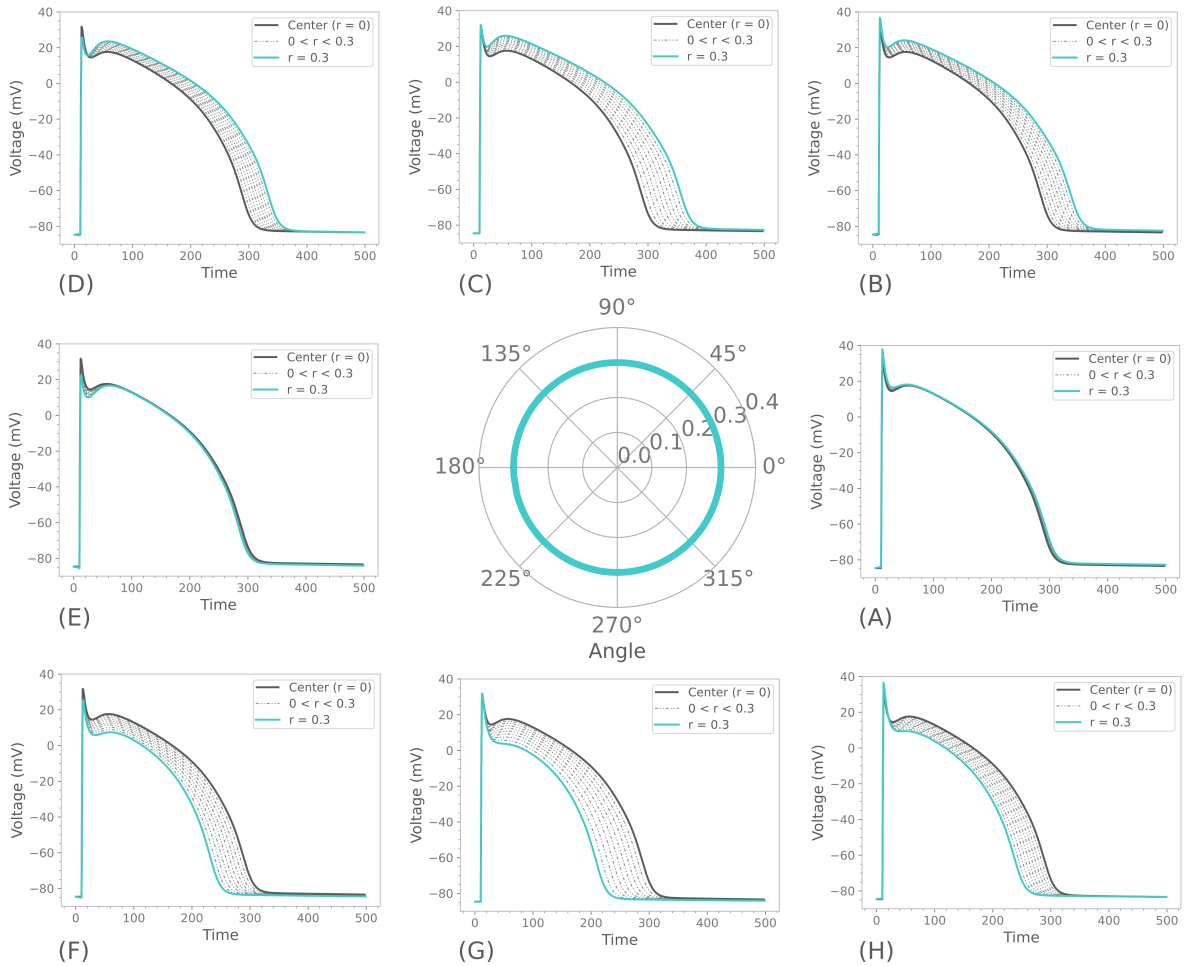


Figure 12: Illustration of how the voltage curves change as a function of direction and radius away from the center $p = (1, 1)$. The middle panel is a polar representation of the direction and radial distance away from $p = (1, 1)$. Each panel (A)-(H) shows 10 curves equally spaced on the interval [0, 0.3]. In particular, the center curve $r = 0$ is a dark solid line, and the curve at $r = 0.3$ is a solid light-blue line. Note that panel (A) corresponds to direction 0°, (B) to direction 45° etc.

mature ventricular cells. This scheme has implications in decreasing dependency on highly specialized laboratory expertise, which can contribute to reducing the cost and time required to develop and launch a new drug to market. The relevance of our method can be seen in relation to the utility of this pipeline.

Other recent studies have also contributed to advancing the pipeline proposed by [18]. In particular, [20] introduces a new computational scheme for inverting the AP model based on a continuation-based optimization algorithm [52], which searches for the optimal parameter of an AP trace by moving iteratively from a known initial parameter. Meanwhile, [53] shows in the context of hiPSC-CMs, that by combining the extracellular potential with the membrane potential and the calcium concentration, the identifiability of the fast sodium (Na^+) current, the main contributor to the rapid upstroke of the AP, can be improved. They also show that other significant currents can be identified in this way.

Several alternative pipelines using hiPSC-CMs measurements to predict drug-effect responses in adult human ventricular cardiomyocytes have also been developed. Using statistical modeling and multivariable regression, [54] developed a cross-cell type regression model for predicting responses in human adult ventricular cardiomyocytes from measurements in hiPSC-CMs. Meanwhile, [55] combined *in silico* simulation trails with optical measurments of AP and Ca^{2+} traces in hiPSC-CMs to estimate drug effects. Similarly, [56] estimated drug effects by combining *in silico* simulation trails with Ca^{2+} trace measurements from hiPSC-CMs and also from electrocardiogram measurements from animal preparations.

The use of machine learning techniques for drug effect estimation has also been used in several studies. [57] developed a classifier for estimating drug torsadogenicity using statistical analysis and machine learning techniques. Extracting several key features from the AP and intercellular Ca^{2+} traces of clinical data, a support vector classifier was trained to distinguish between arrhythmogenic and non-

arrhythmogenic drugs. [58] developed a deep-learning classification network capable of learning from time-series data, such as cardiac AP traces. The network was used to differentiate between drugged and drug-free adult ventricular myocytes from measurements of AP traces in hiPSC-CMs and to estimate how electrophysiological perturbations influence the maturation of hiPSC-CMs. The network resembles our proposed scheme as it requires no assumptions about system parameters and can, in principle, be used for any time series data. A notable difference is that the network predicts drug effects in adult ventricular myocytes directly from AP traces measured on hiPSC-CMs, without using a maturation step such as that developed in [18].

4.5 Limitations and future work

We have demonstrated the parameter estimation capability of StreaMRAK on the 1977 Beeler-Reuter AP model of mammalian ventricular cardiomyocytes [37]. Although the BR model captures currents that are of interest for studying the electrophysiological effects of drugs, such as the fast inward sodium (Na^+) current and a slow inward current, primarily carried by calcium (Ca^{2+}) ions, there exist several other AP models which incorporate other more specific currents. Notable examples include: the Ten Tusscher ventricular model which incorporates several other transmembrane currents, transmural differences in currents, and more complex calcium dynamics [9]; the Grandi model which includes an excitation-contraction component [10]; and the O'Hara (ORd) model which more accurately models the human-specific undiseased ventricular action potential and causes of arrhythmic behavior [11]. These models capture more detailed electrophysiology and give a better representation of the cardiac cell, albeit less interpretable. Because of this, future work should focus on extending the analysis in this paper also to these models. Of particular importance with these more complex models is the presence of more realistic potassium (K^+) current dynamics, which is a key current when assessing anti- and pro-arrhythmic drugs [59].

A notable drawback with the proposed kernel regression schemes of StreaMRAK and FALKON is their considerable upfront training time compared to the loss-minimization schemes that do not require any training in advance. However, as shown in this study, this is balanced by the reduction in sample points required to achieve high accuracy, which reduces the computational burden of solving the system of ODEs in the AP model and reduces the memory requirement of storing the samples. Nevertheless, it is worth mentioning that in situations where an extensive upfront training time is undesirable, and high reliability and low memory are of less importance, then the loss-minimization schemes can be more suitable options.

Furthermore, our experiments demonstrate StreaMRAK on an AP model for adult human cardiomyocytes. The next step would be to utilize this model in the drug-effect-identification pipeline developed in [18, 20] and to extend the demonstration to AP models of immature human induced pluripotent stem cells (hiPSC-CMs), such as [39, 60].

Moreover, validation of the methodology against experimental drug response data should be performed. Initially, in terms of drug response experiments on hiPSC-CMs in microphysiological systems, since experimental control data for hiPSC-CMs are more readily attainable. However, in future work, we also aim to experimentally validate the StreaMRAK inversion for AP models of adult human cardiomyocytes. Drug response data for adult human cardiomyocytes is difficult to attain because both low-throughput patch clamp techniques and higher-throughput, multi-cell platforms require highly specialized laboratory expertise and high initial expense for the instrumentation [2, 3]. Because of this, validation of the inversion of AP models of adult human cardiomyocytes will mainly be done using live cell fluorescence microscopy [4, 5]. This can be done by feeding the inverted parameters back into the AP model and then comparing the resulting AP traces with the measured ones.

5 Acknowledgments

AO is funded by Simula Research Laboratory. AC was partially funded by NSF DMS 1819222 and 2012266, and a gift from Intel research. NF is funded by Simula Research Laboratory.

References

[1] Ping Liang, Feng Lan, Andrew S Lee, Tingyu Gong, Veronica Sanchez-Freire, Yongming Wang, Sebastian Diecke, Karim Sallam, Joshua W Knowles, Paul J Wang, et al. Drug screening using a library of human induced pluripotent stem cell–derived cardiomyocytes reveals disease-specific patterns of cardiotoxicity. *Circulation*, 127(16):1677–1691, 2013.

[2] Bruce G. Kornreich. The patch clamp technique: Principles and technical considerations. *Journal of Veterinary Cardiology*, 9(1):25–37, 2007.

[3] Chris Chambers, Ian Witton, Cathryn Adams, Luke Marrington, and Juha Kammonen. High-throughput screening of $Na_V1.7$ modulators using a giga-seal automated patch clamp instrument. *Assay and drug development technologies*, 14(2):93–108, 2016.

[4] Todd J Herron, Peter Lee, and José Jalife. Optical imaging of voltage and calcium in cardiac cells & tissues. *Circulation research*, 110(4):609–623, 2012.

[5] Anurag Mathur, Peter Loskill, Kaifeng Shao, Nathaniel Huebsch, SoonGweon Hong, Sivan G. Marcus, Natalie Marks, Mohammad Mandegar, Bruce R. Conklin, Luke P. Lee, and Kevin E. Healy. Human iPSC-based cardiac microphysiological system for drug screening applications. *Scientific Reports*, 5(1), March 2015.

[6] Gernot Schram, Marc Pourrier, Peter Melnyk, and Stanley Nattel. Differential distribution of cardiac ion channel expression as a basis for regional specialization in electrical function. *Circulation research*, 90(9):939–950, 2002.

[7] Jeanne M Nerbonne and Robert S Kass. Molecular physiology of cardiac repolarization. *Physiological reviews*, 85(4):1205–1253, 2005.

[8] Yoram Rudy and Jonathan R. Silva. Computational biology in the study of cardiac ion channels and cell electrophysiology. *Quarterly Reviews of Biophysics*, 39(1):57–116, 2006.

[9] Kirsten HWJ Ten Tusscher and Alexander V Panfilov. Alternans and spiral breakup in a human ventricular tissue model. *American Journal of Physiology-Heart and Circulatory Physiology*, 291(3):H1088–H1100, 2006.

[10] Eleonora Grandi, Francesco S. Pasqualini, and Donald M. Bers. A novel computational model of the human ventricular action potential and ca transient. *Journal of Molecular and Cellular Cardiology*, 48(1):112–121, 2010.

[11] Thomas O'Hara, László Virág, András Varró, and Yoram Rudy. Simulation of the undiseased human cardiac ventricular action potential: Model formulation and experimental validation. *PLOS Computational Biology*, 7(5):1–29, 05 2011.

[12] Y. Rudy. From genes and molecules to organs and organisms: Heart. In *Comprehensive Biophysics*, pages 268–327. Elsevier, 2012.

[13] Piero Colli Franzone, Luca Franco Pavarino, and Simone Scacchi. *Mathematical cardiac electrophysiology*, volume 13. Springer, 2014.

[14] Zhilin Qu, Gang Hu, Alan Garfinkel, and James N. Weiss. Nonlinear and stochastic dynamics in the heart. *Physics Reports*, 543(2):61–162, 2014.

[15] Aslak Tveito, Karoline H. Jæger, Miroslav Kuchta, Kent-Andre Mardal, and Marie E. Rognes. A cell-based framework for numerical modeling of electrical conduction in cardiac tissue. *Frontiers in Physics*, 5, 2017.

[16] Alfio Quarteroni, Toni Lassila, Simone Rossi, and Ricardo Ruiz-Baier. Integrated heart—coupling multiscale and multiphysics models for the simulation of the cardiac function. *Computer Methods in Applied Mechanics and Engineering*, 314:345–407, 2017.

[17] Andrew G Edwards and William E Louch. Species-dependent mechanisms of cardiac arrhythmia: a cellular focus. *Clinical Medicine Insights: Cardiology*, 11:1179–5468, 2017.

[18] Aslak Tveito, Karoline Horgmo Jæger, Nathaniel Huebsch, Berenice Charrez, Andrew G Edwards, Samuel Wall, and Kevin E Healy. Inversion and computational maturation of drug response using human stem cell derived cardiomyocytes in microphysiological systems. *Scientific reports*, 8(1):1–14, 2018.

[19] Karoline Horgmo Jæger, Samuel Wall, and Aslak Tveito. Detecting undetectables: Can conductances of action potential models be changed without appreciable change in the transmembrane potential? *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(7):73–102, 2019.

[20] Karoline Horgmo Jæger, Verena Charwat, Bérénice Charrez, Henrik Finsberg, Mary M. Maleckar, Samuel Wall, Kevin E. Healy, and Aslak Tveito. Improved computational identification of drug response using optical measurements of human stem cell derived cardiomyocytes in microphysiological systems. *Frontiers in Pharmacology*, 10, 2020.

[21] Denis Noble, Alan Garny, and Penelope J Noble. How the Hodgkin–Huxley equations inspired the cardiac physiome project. *The Journal of physiology*, 590(11):2613–2628, 2012.

[22] Dominic G Whittaker, Michael Clerx, Chon Lok Lei, David J Christini, and Gary R Mirams. Calibration of ionic and cellular cardiac electrophysiology models. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 12(4):e1482, 2020.

[23] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[24] Andreas Oslandsbotn, Željko Kereta, Valeriya Naumova, Yoav Freund, and Alexander Cloninger. Streamrak a streaming multi-resolution adaptive kernel algorithm. *Applied Mathematics and Computation*, 426:127112, 2022.

[25] J. Kivinen, A.J. Smola, and R.C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004.

[26] Bernhard Schölkopf and Alexander J Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 1 edition, 2002.

[27] Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(95):2651–2667, 2006.

[28] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, volume 28, pages 775–783, 2015.

[29] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

[30] Haim Avron, Kenneth L. Clarkson, and David P. Woodruff. Faster kernel ridge regression using sketching and preconditioning. *J. Matrix. Anal. Appl.*, 38(4):1116–1138, 2017.

[31] Evgeny Burnaev and Ivan Nazarov. Conformalized kernel ridge regression. In *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 45–52, 2016.

[32] Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. Falkon: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[33] Benjamin Yee Shing Li, Lam Fat Yeung, and King Tim Ko. Indefinite kernel ridge regression and its application on QSAR modelling. *Neurocomputing*, 158:127–133, 2015.

[34] P. Mohapatra, S. Chakravarty, and P.K. Dash. Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system. *Swarm and Evolutionary Computation*, 28:144–160, 2016.

[35] Peter Exterkate, Patrick J.F. Groenen, Christiaan Heij, and Dick van Dijk. Nonlinear forecasting with many predictors using kernel ridge regression. *Int. J. Forecas*, 32(3):736–753, 2016.

[36] Michiel Stock, Tapio Pahikkala, Antti Airola, Bernard De Baets, and Willem Waegeman. A comparative study of pairwise learning methods based on kernel ridge regression. *Neural Comput.*, 30(8):2245–2283, 2018.

[37] Go W Beeler and H Reuter. Reconstruction of the action potential of ventricular myocardial fibres. *The Journal of physiology*, 268(1):177–210, 1977.

[38] David G Strauss, Gary Gintant, Zhihua Li, Wendy Wu, Ksenia Blinova, Jose Vicente, J Rick Turner, and Philip T Sager. Comprehensive in vitro proarrhythmia assay (CiPA) update from a cardiac safety research consortium / health and environmental sciences institute / FDA meeting. *Therapeutic Innovation & Regulatory Science*, 53(4):519–525, 2019.

[39] Michelangelo Paci, Jari Hyttinen, Katriina Aalto-Setälä, and Stefano Severi. Computational models of ventricular-and atrial-like human induced pluripotent stem cell derived cardiomyocytes. *Annals of biomedical engineering*, 41(11):2334–2348, 2013.

[40] R. Halfar. Dynamical properties of beeler–reuter cardiac cell model with respect to stimulation parameters. *International Journal of Computer Mathematics*, 97(1-2):498–507, 2020.

[41] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[42] Mikhail Belkin and Partha Niyogi. Convergence of laplacian eigenmaps. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

[43] Henrik Finsberg. Computational physiology, 2022. Available online at: `https://github.com/ComputationalPhysiology/ap_features/tree/main/demo/beeler_reuter`, last accessed on 01.26.2022.

[44] Andreas Oslandsbotn. Python implementation of Streamrak and Falkon with drug effect estimation performance experiments. *GitHub repository AndOslandsbotn*, 2023. `https://github.com/AndOslandsbotn/Identifying-cardiac-drug-effects-using-StreaMRAK`.

[45] Rodney H Falk and Richard I Fogel. Flecainide. *Journal of cardiovascular electrophysiology*, 5(11):964–981, 1994.

[46] F Scamps, A Undrovinas, and G Vassort. Inhibition of ICa in single frog cardiac cells by quinidine, flecainide, ethmozin, and ethacizin. *American Journal of Physiology-Cell Physiology*, 256(3):C549–C559, 1989.

[47] David J Triggle. L-type calcium channels. *Current pharmaceutical design*, 12(4):443–457, 2006.

[48] Andrew A Grace and A John Camm. Quinidine. *New England Journal of Medicine*, 338(1):35–45, 1998.

[49] XIAN-GANG Zong, Martin Dugas, and P Honerjäger. Relation between veratridine reaction dynamics and macroscopic Na current in single cardiac cells. *The Journal of general physiology*, 99(5):683–697, 1992.

[50] Matthias Schramm and Robertson Towart. Modulation of calcium channel function by drugs. *Life Sciences*, 37(20):1843–1860, 1985.

[51] Songrit Maneewongvatana and David M Mount. Analysis of approximate nearest neighbor searching with clustered point sets. *arXiv preprint arXiv:cs/9901013v1*, 1999.

[52] Eugene L Allgower and Kurt Georg. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media, 2012.

[53] Karoline Horgmo Jæger, Verena Charwat, Samuel Wall, Kevin E. Healy, and Aslak Tveito. Identifying drug response by combining measurements of the membrane potential, the cytosolic calcium concentration, and the extracellular potential in microphysiological systems. *Frontiers in Pharmacology*, 11:569–489, 2021.

[54] Jingqi QX Gong and Eric A Sobie. Population-based mechanistic modeling allows for quantitative predictions of drug responses across cell types. *NPJ systems biology and applications*, 4(1):11, 2018.

[55] Michelangelo Paci, Elisa Passini, Aleksandra Klimas, Stefano Severi, Jari Hyttinen, Blanca Rodriguez, and Emilia Entcheva. All-optical electrophysiology refines populations of in silico human iPSC-CMs for drug evaluation. *Biophysical Journal*, 118(10):2596–2611, 2020.

[56] Elisa Passini, Oliver J Britton, Hua Rong Lu, Jutta Rohrbacher, An N Hermans, David J Gallacher, Robert JH Greig, Alfonso Bueno-Orovio, and Blanca Rodriguez. Human in silico drug trials demonstrate higher accuracy than animal models in predicting clinical pro-arrhythmic cardiotoxicity. *Frontiers in physiology*, pages 668,, 2017.

[57] M Cummins Lancaster and EA Sobie. Improved prediction of drug-induced torsades de pointes through simulations of dynamics and machine learning algorithms. *Clinical Pharmacology & Therapeutics*, 100(4):371–379, 2016.

[58] Parya Aghasafari, Pei-Chi Yang, Divya C Kernik, Kazuho Sakamoto, Yasunari Kanda, Junko Kurokawa, Igor Vorobyov, and Colleen E Clancy. A deep learning algorithm to translate and classify cardiac electrophysiology. *eLife*, 10:68335, jul 2021.

[59] Thomas Colatsky, Bernard Fermini, Gary Gintant, Jennifer B. Pierson, Philip Sager, Yuko Sekino, David G. Strauss, and Norman Stockbridge. The comprehensive in vitro proarrhythmia assay (CiPA) initiative — update on progress. *Journal of Pharmacological and Toxicological Methods*, 81:15–20, 2016.

[60] Michelangelo Paci, Elisa Passini, Stefano Severi, Jari Hyttinen, and Blanca Rodriguez. Phenotypic variability in LQT3 human induced pluripotent stem cell-derived cardiomyocytes and their response to antiarrhythmic pharmacologic therapy: An in silico approach. *Heart Rhythm*, 14(11):1704–1712, 2017.

[61] Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

[62] Amit Singer and Ronald R. Coifman. Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25(2):226–239, 2008.

[63] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171 – 1220, 2008.

[64] Lorenzo Rosasco, Mikhail Belkin, and Ernesto De Vito. On learning with integral operators. *Journal of Machine Learning Research*, 11(30):905–934, 2010.

[65] MA Aiserman, Emmanuil M Braverman, and Lev I Rozonoer. Theoretical foundations of the potential function method in pattern recognition. *Avtomat. i Telemeh.*, 25(6):917–936, 1964.

[66] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, page 144–152, 1992.

[67] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *Int. Conf. Comput. Learn. Theory*, pages 416–426, 2001.

6 Appendix

We recapture some minor results and definitions and finish with a closer look at kernel methods and a more extensive outline of StreaMRAK and FALKON.

6.1 The inverse of multivariate vector functions

The inverse function theorem 6.1 states that a vector-valued function over multiple variables $f : X \subseteq \mathbb{R}^n \to \mathbb{R}^n$ has a local inverse at $\mathbf{x}_0 \in X$ if the jacobian $J_f(\mathbf{x}_0)$ is invertible. In other words, if the determinant $\det J_f(\mathbf{x}_0) \neq 0$.

Theorem 6.1 (The inverse function theorem). *(See for instance [61]) Let $X \in \mathbb{R}^n$ be open, and let $f : X \to \mathbb{R}^n$ be a continuously differentiable function $f \in C^1(X, \mathbb{R}^n)$. Let $\mathbf{x}_0 \in X$ and let $J(\mathbf{x}_0)$ be the Jacobian at $\mathbf{x}_0$. If $J_f(\mathbf{x}_0)$ is invertible (i.e. $\det J_f(\mathbf{x}_0) \neq 0$), then there exists an open neighborhood $\mathcal{N}(\mathbf{x}_0)$ such that the inverse f^{-1} of $f : \mathcal{N}(\mathbf{x}_0) \to f(\mathcal{N}(\mathbf{x}_0))$ exists and $J_{f^{-1}}(f(\mathbf{x}_0)) = (J_f(\mathbf{x}_0))^{-1}$.*

6.2 Definitions and minor results

Definition 6.2 (Jacobian). *Consider a function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}^m$ where all partial derivatives exists for $\mathbf{x} \in X$. Then the Jacobian matrix of f is defined as*

$$J_f(\mathbf{x}) = \begin{bmatrix} \partial_1 f_1(\mathbf{x}) & \dots & \partial_n f_m(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \partial_m f_1(\mathbf{x}) & \dots & \partial_n f_m(\mathbf{x}) \end{bmatrix}$$

Definition 6.3 (Positive semi-definite matrix). *Let $\mathbf{A} \subset \mathbb{R}^{n \times n}$ be a symmetric matrix. If $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$ then we say that $\mathbf{A}$ is a positive semi-definite matrix.*

Definition 6.4 (Positive definite function). *Let $f : \mathbb{R} \to \mathbb{C}$ be a complex-valued function and $\mathbf{A}_{ij} = f(\mathbf{x}_i - \mathbf{x}_j)$ be the symmetric matrix induced by f. If $\mathbf{A}$ is a positive semi-definite matrix as follows from Def. 6.3, then we say f is a positive definite function.*

Definition 6.5 (Epsilon cover). *Consider the metric space $(\mathcal{X}, d)$ and let $\Gamma_\varepsilon \subset \mathcal{X}$ be a subset of samples from $\mathcal{X}$. Furthermore, let $\epsilon, \delta > 0$ be two constants. If for every sample $\mathbf{x}' \in \Gamma_\varepsilon$ we have $\varepsilon < d(\mathbf{x}', \mathbf{x}) < \varepsilon + \delta$ for all $\mathbf{x} \in \Gamma_\varepsilon$ such that $\mathbf{x} \neq \mathbf{x}'$. Then we call Γ_ε an epsilon cover of $\mathcal{X}$.*

6.3 A note on invertability

From the inverse function theorem 6.1, we know that a multi-variable function such as F_T is locally invertible over a region where the jacobian $J_F(p)$ from Def 6.2 (the matrix containing all partial derivatives of F_T) is invertible, i.e., $\det J_F(p) \neq 0$. This means, in essence, that we can construct a model of F_T^{-1} from some domain $\mathcal{V}$ to the parameter space $\mathcal{P}$ provided the forward map is unique; all voltage curves $v \in \mathcal{V}_T$ correspond to unique parameters $p \in \mathcal{P}$. We note that if a domain $\mathcal{V}_T$ contains voltage curves $v_i \neq v_j$ for which the corresponding parameters are equal $p_i = p_j$, it is necessary first to split the domain into sub-domains over which the forward map is unique. Separate models can then be learned for each of these sub-domains.

6.4 Local distances in the image of a non-linear map

Consider the spaces X, Y and let $f : X \to Y$ be a non-linear map between them. Let $x_0 \in X$ be a point in X and $y_0 = f(x_0)$. We then have that the ball $B_X(x_0, \delta)$ in X, defined as

$$B(x_0, \delta) = \{x \in X : \|x - x_0\| \leq \delta\},$$

maps to the ellipse

$$\{y \in Y : \left\|J_{f^{-1}}(y_0)(y - y_0)\right\| \leq \delta\},$$

in the output space Y, where $J_{f^{-1}}(y_0)$ is the Jacobian (derivative) of f^{-1} at y_0, see section 6.1 in [62] for a proof.

6.5 Kernel methods

The use of kernel methods for supervised learning has a strong theoretical foundation [26, 63, 64]. Kernel methods rely on the use of a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, with $\mathcal{X} \subset \mathbb{R}^d$, to construct an infinite-dimensional vector space of functions $\mathcal{H}_k = \overline{\text{span}}\{k(\mathbf{x}, \mathbf{x}') : \mathbf{x}' \in \mathcal{X}\}$, called a reproducing kernel Hilbert space (RKHS). See Def 6.4 for the definition of a positive definite function. As $\mathcal{H}_k$ is a vector space, any function $h \in \mathcal{H}_k$ can be expressed as a linear combination of the basis vectors $k_{\mathbf{x}}(\mathbf{x}') := k(\mathbf{x}, \mathbf{x}')$. Furthermore, the advantage of creating this function space comes to light through the "kernel trick" [65, 66], which states that each basis vector corresponds to the inner product between two non-linear feature vectors $\phi(\mathbf{x}), \phi(\mathbf{x}') \in \mathcal{H}_k$, namely $k_{\mathbf{x}'}(\mathbf{x}) = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$. As the feature vectors can be highly non-linear functions in $\mathbf{x}$, a linear model in $\mathcal{H}_k$ corresponds to a highly complex function in the input space $\mathcal{X}$, which explains the great expressive power of this space.

Consider training data $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ drawn from the sample space $\mathcal{X} \times \mathbb{R}$ according to some probability distribution ρ. Supervised learning aims to train a model that gives a good approximation of the function that generated this data. Using an RHKS as our model space, we can then formulate the optimization problem

$$\widehat{h}_n = \underset{h \in \mathcal{H}_n}{\text{argmin}} \sum_{i=1}^n (h(\mathbf{x}_i) - y_i)^2 + \lambda \|h\|_{\mathcal{H}_n}^2, \quad \text{for each} \quad j \in [d], \tag{6.1}$$

where the model space is restricted to the finite-dimensional subspace $\mathcal{H}_n \subset \mathcal{H}_k$ due to the finite size of the training data. The solution to this problem is guaranteed by the Representer theorem [67] to be a linear combination of the basis vectors that make up $\mathcal{H}_n$, namely

$$h(\mathbf{x}) = \sum_{i=1}^n \alpha_i k(\mathbf{x}, \mathbf{x}_i).$$

It follows that the optimization problem in Eq. (6.1) can be solved as the linear system $(\mathbf{K} + \lambda \mathbf{I}_n)\boldsymbol{\alpha} = \mathbf{y}$, where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $y = (y_1, \ldots, y_n)^\top$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top$. However, inverting the $n \times n$ matrix $\mathbf{K}$ is computationally expensive, with a time complexity of $\mathcal{O}(n^2)$. To remedy this, the large-scale kernel method FALKON [32] introduces several improvements to speed up the inversion.

6.5.1 FALKON

In FALKON the computation complexity of KRR is reduced by selecting a random subset of the training data $\widetilde{\Gamma}_m = \{\widetilde{\mathbf{x}}_i\}_{i=1}^m$ for $m \ll n$, which we refer to as landmarks or Nyström sub-samples. The model space is then reduced to the span of the kernels centered on these sub-samples, namely $\widetilde{\mathcal{H}}_m = \overline{\text{span}}\{k(\mathbf{x}, \widetilde{\mathbf{x}}') : \widetilde{\mathbf{x}}' \in \widetilde{\Gamma_m}\}$. Solving the optimization problem in Eq. (6.1) with the reduced model space gives rise to a linear system involving the much smaller $n \times m$ kernel matrix $\widetilde{\mathbf{K}}_{ij} = k(\mathbf{x}_i, \widetilde{\mathbf{x}}_j)$. Further improvements are made by introducing a pre-conditioner and solving the linear system iteratively using conjugate gradient with early stopping. For more details on the algorithm, we refer to [32] and [24].

6.5.2 StreaMRAK

The streaming multi-resolution adaptive kernel algorithm StreaMRAK introduced in [24], significantly improves the model used in FALKON and standard KRR. Using principles from boosted gradient descent, an iterative model is introduced which solves the KRR over several levels. The iterative model is on the form

$$h^{(L)}(\mathbf{x}) = \sum_{l=0}^L s^{(l)}(\mathbf{x}) = h^{(L-1)}(\mathbf{x}) + s^{(L)}(\mathbf{x}), \quad h^{(0)} = s^{(0)}$$

where

$$s^{(l)}(\mathbf{x}) = \sum_{i=1}^n \alpha_i^{(l)} k^{(l)}(\mathbf{x}, \mathbf{x}_i) \quad \textit{with} \quad k^{(l)}(\mathbf{x}, \mathbf{x}_i) = \exp\left(- \frac{\|\mathbf{x} - \mathbf{x}_i\|}{2r_l^2} \right) \tag{6.2}$$

is a correction term to the previous level. The coefficients $\alpha_i^{(l)}$ at each level l are found by using the FALKON solver to regress on the residuals $\mathbf{d}^{(l)} = (d_i^{(l)}, \ldots, d_n^{(l)}) \in \mathbb{R}^n$ from the previous level. The

residuals are defined as

$$d_i^{(l)} = \begin{cases} y_i & l = 1 \\ y_i - h^{(l-1)}(\mathbf{x}_i) & l > 1. \end{cases}$$

where $h^{(l-1)}(\mathbf{x}_i)$ is the prediction of y_i using the trained model from the previous level.

In StreaMRAK the kernel bandwidth r_l is reduced at each new level as $r_l = 2^{-l} r_0$. Furthermore, instead of the random sub-sampling of landmarks used in FALKON, the landmarks are selected from an epsilon cover with epsilon $\varepsilon = r_l$, see Def. 6.5. This way the distance between landmarks is tailored to the kernel bandwidth, which allows the optimal utility of each sub-sample. With these two choices, StreaMRAK implements an adaptive multi-resolution scheme that is more robust at learning complex functions than standard KRR solvers and requires significantly less memory [24].

6.6 Appendix: Supplementary experiments

We include two experiments that did not fit in the main text. In Figure 13, we see the MSE of the predictions over the parameter domain $P = [0.2, 2]^2$ as a function of training data size n, where $n \in [1000, 60000]$. In Table 4 and Table 5 we see the Root mean square error (RMSE), Max absolute error (Max abs. err.) and standard deviation (Std) for the parameter estimations of (g_{Na}, g_s) along the drug directions A, B and C from the experiment presented in Section 3.4.

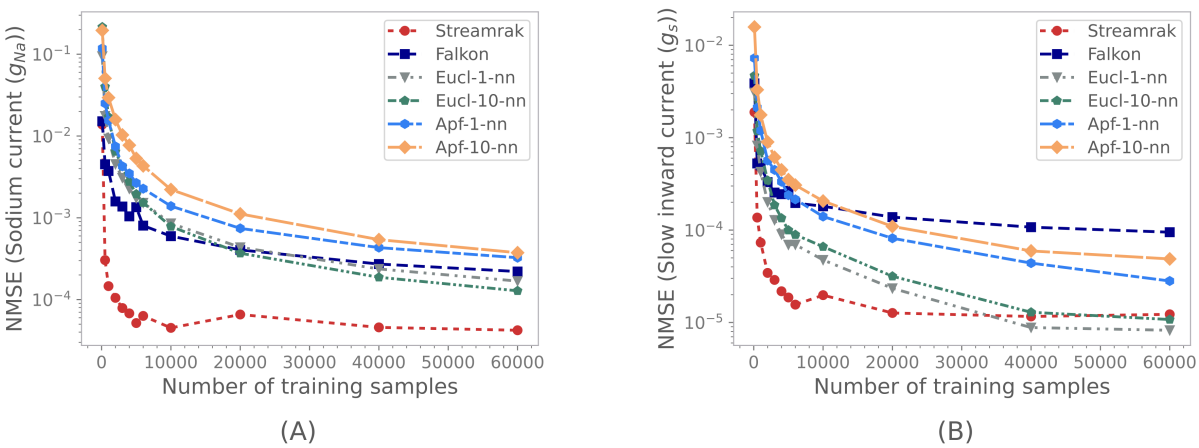


Figure 13: MSE of the predictions over the parameter domain $P = [0.2, 2]^2$ as a function of training data size. Panel (A) shows the MSE of predictions of the g_{Na} parameter (Na^+ current). Panel (B) shows the MSE of predictions of the g_s parameter (Ca^{2+} -like current).

Table 4: Table showing Root mean square error (RMSE), Max absolute error (Max abs. err.) and standard deviation (Std) for the parameter estimations of (g_{Na}, g_s) along the drug directions A and B from the experiment presented in Section 3.4.

	Calcium blocker (A)			**Mixed blocker (B)**		
Algorithms	RMSE	Max abs. err.	Std	RMSE	Max abs. err.	Std
Streamrak	(4.e-3, 3.e-4)	(9.e-3, 8.e-4)	(4.e-3, 3.e-4)	(5.e-3, 5.e-4)	(1.e-2, 1.e-3)	(5.e-3, 5.e-4)
Falkon	(3.e-2, 2.e-3)	(3.e-2, 3.e-3)	(3.e-3, 5.e-4)	(2.e-2, 2.e-3)	(3.e-2, 3.e-3)	(6.e-3, 5.e-4)
Eucl-1-nn	(3.e-2, 2.e-3)	(7.e-2, 4.e-3)	(3.e-2, 2.e-3)	(3.e-2, 4.e-3)	(7.e-2, 7.e-3)	(2.e-2, 4.e-3)
Eucl-10-nn	(3.e-2, 2.e-3)	(6.e-2, 5.e-3)	(3.e-2, 2.e-3)	(3.e-2, 2.e-3)	(5.e-2, 4.e-3)	(3.e-2, 2.e-3)
Apf-1-nn	(3.e-2, 1.e-2)	(6.e-2, 2.e-2)	(3.e-2, 9.e-3)	(4.e-2, 1.e-2)	(7.e-2, 3.e-2)	(4.e-2, 1.e-2)
Apf-10-nn	(3.e-2, 1.e-2)	(6.e-2, 3.e-2)	(3.e-2, 1.e-2)	(5.e-2, 2.e-2)	(7.e-2, 3.e-2)	(5.e-2, 2.e-2)

Table 5: Table showing Root mean square error (RMSE), Max absolute error (Max abs. err.) and standard deviation (Std) for the parameter estimations of (g_{Na}, g_s) along the drug directions C from the experiment presented in Section 3.4.

	Sodium blocker (C)		
Algorithms	RMSE	Max abs. err.	Std
Streamrak	(6.e-3, 5.e-4)	(1.e-2, 1.e-3)	(5.e-3, 5.e-4)
Falkon	(2.e-2, 2.e-3)	(3.e-2, 3.e-3)	(8.e-3, 4.e-4)
Eucl-1-nn	(3.e-2, 3.e-3)	(6.e-2, 4.e-3)	(3.e-2, 2.e-3)
Eucl-10-nn	(4.e-2, 5.e-3)	(6.e-2, 6.e-3)	(2.e-2, 1.e-3)
Apf-1-nn	(5.e-2, 2.e-2)	(1.e-1, 5.e-2)	(5.e-2, 2.e-2)
Apf-10-nn	(3.e-2, 1.e-2)	(5.e-2, 2.e-2)	(3.e-2, 2.e-2)