UNIVERSITY
OF OSLO

Scaling kernel-based learning for big data

Andreas Oslandsbotn

Supervisors

Alexander Cloninger
Nickolas Forsch
Željko Kereta
Aslak Tveito

Department of Informatics

The Faculty of Mathematics and Natural Sciences

Under linden

Sitte på ein stubbe under linden
I den leikande varme augustvinden
Lye til biene sitt summekor
Der dei strevar og samlar inn vinterfor
Kjenne angen av roser og kaprifol
Og varmande strålar frå seinsommarsol
Nyte selskap med erla og raudstrupen vår
Som tok ferien sin her i hagen i år
Han svinsar og sprett ifrå grein til grein
So syng han ei strofe so klar og so rein
Og erla ho trippar og leitar seg føde
Ho er ferdig for i år med barnestell og møde
Så no skal me nyte vår otium me tre
Her i hagen, under linden, i ro og i fred

Haldis Brenne (1920 – 2011)

Abstract

In the face of datasets with ever increasing sizes, development of scalable learning algorithms with high predictive power is at the forefront of machine learning research. The design of learning schemes using a positive semi-definite function at the core (a kernel function), is motivated by a solid theoretical foundation. These schemes have shown great success on moderately sized datasets. However, kernel-based methods suffers from scalability issues, due to their inherent large memory requirements and computational costs. In this thesis we address this issue by developing kernel-based learning schemes that are compatible with efficient computational models such as streaming, paralellization and distribution. We support the proposed algorithms with theoretical and numerical results.

Samandrag

I møte med store datamengder, er utvikling av skalerbare læringsalgoritmar med gode generaliseringseigenskapar eit viktig forskingsområde i maskinlæring. Konstruksjon av læringsmetoder som utnyttar en positiv semi-definit funksjon (ein kernelfunksjon), er ein strategi som har vist stor suksess ved moderate datamengder og er støtta av eit solid teoretisk grunnlag. Desverre er kernelbaserte metodar avgrensa frå bruk i møte med store datamengder, på grunn av betydelege minne og kalkulasjonsbehov. I denne doktorgraden løyser vi dette problemet ved å utvikle kernelbaserte læringsmetoder som er kompatible med effektive modeller slik som parallellisering, distribuerte system og strømming av data.

Contents

List of Figures

List of Tables

Preface

This thesis is submitted for the degree of *Philosophiae Doctor* at the University of Oslo. The research presented in this thesis was conducted at Simula Research Laboratory from September 2019 until September 2021 and then at the Department of Mathematics, University of California San Diego, between October 2021 and March 2023, before returning to Simula Research Laboratory from April 2023 until June 2023.

The thesis has been conducted under the supervision of Professor Alexander Cloninger at the Department of Mathematics, University of California San Diego, Research Engineer Nickolas Forsch at the Department of Computational Physiology, Simula Research Laboratory, and Željko Kereta, Research Fellow at the Department of Computer Science, University College London. In the supervisor team has also been Professor Aslak Tveito at the Department of Informatics, University of Oslo.

This document comprises an introduction and four original manuscripts; two first-author papers and two co-first-author papers. The introduction provides an overview of the field and clarifies the contribution of the individual manuscripts.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Alexander Cloninger for following me on this journey; your mathematical intuition and openness to discussions have been greatly appreciated. Furthermore, I would like to thank my supervisor Nickolas Forsch for all his support in progressing this thesis and for the valuable insights he has provided on cardiac physiology. Finally, I would like to thank my supervisor Željko Kereta for the discussions, the support, and the eye for mathematical detail.

I would also like to thank the other researchers I have met during my work on this thesis. In particular, I would like to thank Prof. Yoav Freund for his creative ideas and Robi Bhattacharjee, Zhengchao Wan, Sawyer Robertson, and Stefano Vigogna for their collaboration and discussions.

My gratitude also goes to the other PhD students I have met at Simula. A particular warm thanks to the PhD students at the Suurph program and all the great experiences we have shared. I will also thank Kimberly McCabe for her commitment to me and the other suurphers, helping out with everything around the PhD.

Finally, I would like to thank Maria and my family, who have been there to support me during this long endeavor.

Introduction

Recent advances in computational power, memory capabilities, sensor technology, and the internet have allowed the collection and storage of increasingly large data sets. The size of these data sets is not only due to a massive growth in the number of collected samples but also due to a substantial increase in the number of collected attributes (observable quantities) associated with each sample.

In machine learning and data analysis, we are interested in identifying patterns in data sets to gain insights that can be used in real-world applications. The patterns of interest found in data can be highly non-linear. Therefore, these patterns are often infeasible to model with existing elementary functions containing few tunable parameters, such as linear functions, polynomials, Gaussians, etc. Furthermore, with multiple attributes, the data can be difficult to visualize, preventing insights that could otherwise guide the choice of the model function. Consequently, parametric modeling with elementary functions is limited in the face of non-linear patterns.

The limitations of parametric learning have motivated the development of non-parametric learning schemes that do not rely on strong modeling assumptions and pre-defined knowledge about the data. Notable examples are kernel-based methods, decision trees, and neural networks.

In this thesis, we focus on *kernel-based* methods, meaning all methods that map non-linear input data to a high-dimensional feature space using the so-called kernel function. In doing so, non-linear data dependencies are approximately linearized, allowing the usage of fast and efficient linear models. Kernel-based methods came to prominence at the end of the 1990s [105] with the introduction of the support vector machine (SVM) [32] for non-linear classification, kernel principal component analysis (kernel PCA) [102, 103] for non-linear PCA and kernel ridge regression (KRR) [99, 104] for non-linear regression.

Another common strategy to capture non-linear patterns, also relying on the kernel function, are spectral embedding methods based on nearest neighbor graphs such as ISOMAP [115], locally linear embedding (LLE) [93], Laplacian eigenmaps (LE) [14, 16], and diffusion maps [31].

The popularity of kernel-based methods stems from their solid and studied theoretical foundation [14, 31, 92, 101, 105]. However, kernel-based learning methods generally have large memory and computational requirements due to their reliance on a kernel matrix that scales with the number of training samples.

Consequently, in real-world applications, these methods have largely fallen out of favor in the machine learning community with the rise of alternative methods such as multi-layer neural networks.

The motivating principle of this thesis is that reducing computational costs and memory requirements alone, does not unlock the full potential for scalability of learning methods. Rather, the use of modern computational models, such as parallelization, distribution, and learning from streaming data, is also a critical ingredient. In this thesis, we develop techniques and algorithms for the purpose of scaling up kernel-based learning schemes for big data applications. Our strategy is two-fold. On one hand, by utilizing ideas and concepts such as tailored subsampling, boosting, multi-resolution, sparsity, and iterative solvers, we develop learning schemes with low memory requirements and low computational cost. At the same time, the learning methods developed in this work are designed to be compatible with modern computational models, such as those mentioned above, that are essential for large scale applications. In summary, we aim to develop learning schemes that satisfy the following aspects:

1. *Single-pass* - learning schemes that can learn from seeing each sample only once before discarding it.

2. *Distributed* - learning schemes that can be divided into independent modules that can learn independently or with minimal communication.

3. *Minimize in-memory data* - learning schemes with independent modules that only require access to a fraction of the data.

We consider a setting where data is embedded in a high-dimensional ambient space, the Euclidean space $\mathbb{R}^D$. Furthermore, we assume that the data comes from a distribution supported on, or concentrated around, a lower-dimensional set $\mathcal{X} \in \mathbb{R}^D$. We assume that $\mathcal{X}$ is a point cloud, a notion we define in more detail later. In particular, we consider point clouds where the intrinsic structure can be highly non-linear and vary in dimensionality.

We develop scalable regression algorithms that can learn highly non-linear functions and develop techniques that reduce the impact of the *curse of dimensionality* [122]. Furthermore, we develop scalable algorithms that can learn the intrinsic structure of data for the purpose of dimensionality reduction and quantifying similarity between samples in a non-linear space.

The algorithms we develop are verified numerically and supported by theoretical analysis.

Outline In Section 1 we present an overview of relevant concepts related to non-parametric learning in a supervised setting. We then discuss the curse of dimensionality and associated challenges in Section 2. In Section 3 we introduce the concept of point clouds and discuss challenges and opportunities related to uncovering their intrinsic structure. Section 4 then discusses methods for uncovering the structure in point clouds for purposes such as dimensionality reduction. Section 5 reviews several strategies for scaling learning algorithms to big data applications. In Section 6, we present the contributions of the original manuscripts underlying the research performed in this thesis. Finally, Section 7 concludes with a summary of the findings in the thesis and discusses further work.

Notation We denote matrices with upper case and vectors with lower case. For a matrix $A \in \mathbb{R}^{n \times n}$ we let A_{ij} be the i, j-th entries. For a vector $a \in \mathbb{R}^n$, we let a_i denote its i-th entry. For $\Gamma, \Gamma' \subseteq \{1, \dots n\}$ we let $A(\Gamma, \Gamma')$ mean the sub-matrix constructed from the indices contained in Γ, Γ'. We let $A^\top$ denote the transpose of A and $A^\dagger$ the Moore-Penrose pseudoinverse. For a vector $v \in \mathbb{R}^D$ we let $\|v\|_p$ denote the p-norm. For a function $f : \mathcal{X} \to \mathbb{R}$ and a set of samples $\{x_i\}_{i=1}^n$ we take $f([x_n])$ to mean $f([x_n]) = (f(x_1), f(x_2), \dots, f(x_n))^\top$.

For a distribution ρ we mean by $X \sim \rho$ that X is a random variable distributed according to ρ. We let $\mathbb{E}_{X \sim \rho}[X]$ be the expectancy of X. For random variables $(X, Y) \sim \rho$ we let $\mathbb{E}_{(X,Y) \sim \rho}[Y | X = x]$ be the expectancy of Y conditioned on X. By $\mathcal{N}(\mu, \sigma)$, we refer to the normal distribution with mean μ and covariance σ. We mean by $\text{Uni}(\Omega)$ the uniform distribution over a set $\Omega \subset \mathbb{R}^D$.

We let $L^2(\mathcal{X}, \rho_\mathcal{X})$ be the space of square integrable functions with norm $\|f\|_\rho^2 = \int_\mathcal{X} |f(x)|^2 d\rho_\mathcal{X}$. For a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ we let $\mathcal{H}_k$ denote the reproducing kernel Hilbert space induced by k. The inner product in $\mathcal{H}_k$ is defined as $\langle f, g \rangle_k = \sum_{ij} \alpha_i, \beta_j k(x_i, x_j)$, for $g, f \in \mathcal{H}_k$. The associated norm is $\|\cdot\|^2 = \langle \cdot, \cdot \rangle_k$. We denote by (M, d) a metric space with distance metric d.

1 Supervised nonlinear function learning

Consider the problem of learning the relationship between some response $y \in \mathbb{R}$ and an input $x \in \mathcal{X} \subseteq \mathbb{R}^D$ for the purpose of predicting the response y given a new input sample $x \in \mathcal{X}$. This problem is often addressed in a supervised setting and is encountered in numerous applications. For example in bio-medicine, knowing the relationship between ionic membrane currents x and the action potential in cardiac cells y allows predicting the effects of specific treatments during drug development [57, 118].

We can formulate this problem in the framework of statistical learning theory [34, 121]. In this setting, the input-response pair (x, y) is interpreted as the realization of a random variable (X, Y) sampled from a probability distribution $\rho = \rho(y|x)\rho_\mathcal{X}$. The probability distribution ρ is assumed to be unknown and can only be accessed through a finite set of training samples $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$.

The relationship between input and response is described by the conditional distribution $\rho(y|x)$. However, finding an exact description of this distribution is often infeasible. In regression, the goal is instead to learn a target function $f : \mathcal{X} \to \mathbb{R}$, using the available training samples $\mathcal{D}_n$, that gives a suitable approximation of $\rho(y|x)$.

To learn the target function, it is necessary to define a loss that quantifies estimation quality and guides the learning process. The L_2 error is a popular choice because it is mathematically easy to work with and often gives optimization problems that are computationally cheaper to evaluate than, say, other L_p loss functions [49]. Function learning can then be formulated as the minimization of the expected L_2 risk

$$\mathcal{E}(f) = \mathbb{E}_{(X,Y)\sim\rho}[(f(X) - Y)^2], \tag{1.1}$$

for $f \in L^2(\mathcal{X}, \rho_{\mathcal{X}})$, where $L^2(\mathcal{X}, \rho_{\mathcal{X}})$ is the space of square integrable functions. For the remainder of the text, we take $\mathbb{E}$ to mean $\mathbb{E}_{(X,Y)\sim\rho}$ unless otherwise stated.

Remark 1 (The regression function) *The minimizer of the expected L_2 risk can be shown to be the conditional mean $f_\rho(x) = \mathbb{E}_{Y\sim\rho(y|x)}[Y|X = x]$, called the regression function. However, since ρ is known only through a finite sample, the regression function f_ρ cannot be calculated explicitly. Instead, it can be estimated using the training samples in $\mathcal{D}_n$.*

The classical way to estimate f_ρ is by assuming it can be modeled with an explicit function class of elementary functions, such as polynomials or Gaussians. This approach is known as parametric regression, with linear and polynomial regression as typical examples. The problem with parametric regression is that prior knowledge of the shape and characteristics of f_ρ, such as linearity vs. non-linearity, differentiability, continuity, etc., is rarely known. This is especially problematic in multi-feature settings where visualization of the data is impractical or infeasible, which prevents further insights. Since choosing the wrong model can cause significant errors, parametric learning is limited to simpler learning tasks.

1.1 Non-parametric learning in a hypothesis space

Non-parametric regression, in contrast to parametric regression, does not require pre-defined knowledge of the target function; see Györfi et al. [49]. Nevertheless, learning in non-parametric settings still requires a model that, in some way, can be tuned to the data. The simplest approach is to use estimators that rely on local averages, such as kNN and kernel smoothers. A well-established alternative is to introduce a hypothesis space $\mathcal{H} \subseteq L^2(\mathcal{X}, \rho_{\mathcal{X}})$, in which estimators of f_ρ are pursued. It is important to note that the regression function f_ρ need not be contained in $\mathcal{H}$.

The hypothesis space $\mathcal{H}$ provides the necessary structure to define a model that can be fitted to the data. Examples are the spaces of piece-wise polynomials, splines, or reproducing kernel Hilbert spaces (RKHS). The parameters p defining these hypothesis spaces, such as the degree of the polynomial or the bandwidth of the kernel inducing the RKHS, are normally referred to as hyper-parameters and can be tuned to the data to improve the model. In Section 1.2, we discuss the selection of these parameters in more detail.

When searching for an optimal estimator $\widehat{f}_{n,\lambda}$ in a hypothesis space $\mathcal{H}$, the minimizer of the expected L_2 risk $\mathbb{E}[(f(X) - Y)^2]$ is normally estimated using a penalized version of the empirical risk minimizer

$$\widehat{f}_{n,\lambda} = \operatorname*{argmin}_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 + J_\lambda(f). \tag{1.2}$$

Here $J_\lambda(f)$ is a penalty on the complexity of f, and λ is a hyper-parameter that governs the magnitude of the penalty. We can think of λ as restricting the available hypothesis space to less complex functions, as illustrated in Figure 1.1.

To solve the minimization problem in Eq. (1.2), it is necessary to make explicit choices on the hypothesis space $\mathcal{H}$. In Section 1.4, we discuss a specific choice relevant to the work in this thesis, namely that of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$.

1.2 Choosing the hyper-parameters

When estimating a function from finite samples, it is important to consider the bias-variance trade-off [51] and the danger of overfitting to the training samples. The empirical mean of the L_2-loss, namely

$$\mathcal{E}_n(f) = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2,$$

only penalizes the discrepancy from the available data $\mathcal{D}_n$. Consequently, the minimization of $\mathcal{E}_n(f)$ prefers an estimator that interpolates the training samples, but offers no guarantees to predict the response from new samples. To avoid overfitting, hyper-parameters p are introduced to restrict the complexity of the estimator $\widehat{f}_{n,p}$. These hyper-parameters can, for example, be a penalty on the complexity of the estimator, such as λ in the optimization problem defined in Eq. (1.2). Alternatively, they can be parameters restricting the size of the hypothesis space directly, such as the bandwidth of a Gaussian kernel inducing an RKHS. In Györfi et al. [49], the learning goal is stated as finding the hyper-parameters p that minimize the expected L_2 risk

$$\min_p \mathbb{E}_{(X,Y)\sim\rho}[\widehat{f}_{n,p}(X) - Y|\mathcal{D}_n]$$

conditioned on the training data on which $\widehat{f}_{n,p}$ has been trained.

To select the hyper-parameters, it is necessary to evaluate the estimator $\widehat{f}_{n,p}$ on new samples. Since the only available samples are $\mathcal{D}_n$, this is typically done by splitting $\mathcal{D}_n$ into a training set, a validation set, and a test set. The validation set is then used to tune the hyper-parameters, and the test set to evaluate the final generalization performance. Tuning the hyper-parameters based on a single validation set is vulnerable to biased estimation [49]. Techniques such as leave-one-out [25] and k-fold cross-validation [6] are often used to mitigate this issue. The central idea is to split the training set into equalized batches. Then in each iteration, one batch of data is held out during training to evaluate the estimator's prediction capabilities. The mean of the loss over all batches is the final score of a given hyper-parameter. This procedure is repeated for several possible hyper-parameter choices, and the one with the best score is selected.

A major limitation of cross-validation techniques is the high computational cost of repeating the training process over several batches. Furthermore, cross-validation requires the data to be available in memory, but in a streaming setting, this criterion is not satisfied. In the first paper of this thesis, we address this issue in the context of finding a minimizer in an RKHS $\mathcal{H}_k$ induced by a kernel function k_σ. The bandwidth σ of the kernel is a hyper-parameter that must be determined, with implications for the generalization properties of the estimator.

Our strategy is a trade-off between optimizing the hyper-parameter on the one hand and computational cost and compatibility with streaming on the other.

1.3 A note on min-max rates and excess risk

Although we do not analyze optimal estimation rates in this thesis, for completeness and later reference, we introduce a short discussion on min-max rates and excess risk studied in learning theory. The section is based primarily on the discussions in Cucker and Smale [34] and Györfi et al. [49].

When estimating the regression function f_ρ that minimizes the expected L_2 risk using a finite sample $\mathcal{D}_n$ it is of interest to know to what degree the estimator $\widehat{f}_{n,p}$ approximates f_ρ as $n \to \infty$. The study of these rates is done using the min-max approach; see Györfi et al. [49], which seeks lower bounds for the fastest convergence of

$$\inf_{\widehat{f}_n} \sup_{\rho \in \mathcal{C}} \mathbb{E}_{(X,Y)\sim\rho}[(\widehat{f}_n(X) - f_\rho(X))^2]. \tag{1.3}$$

Here the supremum is taken over some class of distributions $\mathcal{C}$ of the random variables (X, Y), and the infimum is taken over all measurable estimators $\widehat{f}_n$ defined on the data $\mathcal{D}_n$ [49]. In other words, one finds a lower bound on the largest excess risk in $L^2(\mathcal{X}, \rho_\mathcal{X})$ over some class of distributions $\mathcal{C}$.

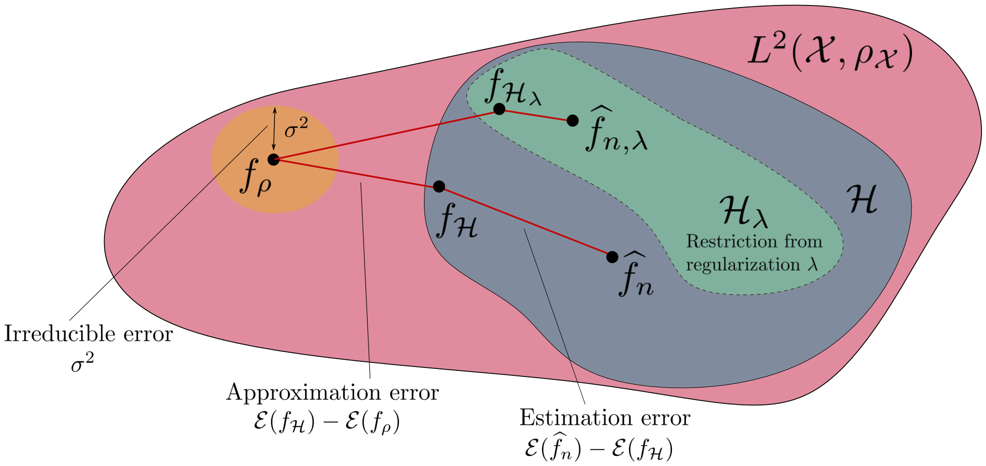

$L^2(\mathcal{X}, \rho_\mathcal{X})$
$f_{\mathcal{H}_\lambda}$
$\widehat{f}_{n,\lambda}$
σ^2
f_ρ
$f_\mathcal{H}$
$\mathcal{H}_\lambda$
Restriction from regularization λ
$\mathcal{H}$
$\widehat{f}_n$
Irreducible error σ^2
Approximation error $\mathcal{E}(f_\mathcal{H}) - \mathcal{E}(f_\rho)$
Estimation error $\mathcal{E}(\widehat{f}_n) - \mathcal{E}(f_\mathcal{H})$

Figure 1.1: Illustration of the hypothesis space $\mathcal{H} \subseteq L^2(\mathcal{X}, \rho_\mathcal{X})$, and the corresponding approximation error $\mathcal{E}(f_\mathcal{H}) - \mathcal{E}(f_\rho)$ and estimation error $\mathcal{E}(\widehat{f}_n) - \mathcal{E}(f_\mathcal{H})$. It is assumed that the noise is additive such that $Y = f_\rho(X) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma)$ and f_ρ is the regression function. $\mathcal{E}(f_\rho) = \sigma^2$ is an irreducible error that can not be improved. The regularization penalizes the complexity of the functions such that hypothesis space is restricted to a subset $\mathcal{H}_\lambda \subseteq \mathcal{H}$. Here $f_{\mathcal{H}_\lambda} = \operatorname{argmin}_{f \in \mathcal{H}_\lambda} \mathcal{E}(f)$. The figure illustrates how the approximation error in $\mathcal{H}_\lambda$ is larger than in $\mathcal{H}$, while the estimation error is smaller. The goal is to find λ such that the overall error $\mathcal{E}(f)$ in Eq. (1.5) is minimized.

In general, the regression function f_ρ is not contained in the hypothesis space $\mathcal{H}$, and the best we can do is $f_\mathcal{H} = \inf_{f \in \mathcal{H}} \mathcal{E}(f)$. This has the consequence of introducing an error $\mathcal{E}(f_\mathcal{H}) - \mathcal{E}(f_\rho)$, known as the approximation error, that can not be reduced by the estimator. We illustrate this situation in Figure 1.1.

Since the approximation error does not depend on the training samples, the min-max rates for an estimator $\widehat{f}_{n,p} \in \mathcal{H}$, are usually studied in terms of the excess risk in $\mathcal{H}$ rather than the excess risk in $L^2(\mathcal{X}, \rho_{\mathcal{X}})$ from Eq. (1.3); See e.g. [24, 34, 95]. We define the excess risk in $\mathcal{H}$ as

$$\mathcal{R}_{\mathcal{H}}(\widehat{f}_{n,p}) = \mathbb{E}[(\widehat{f}_{n,p}(X) - Y)^2] - \inf_{f \in \mathcal{H}} \mathbb{E}[(f(X) - Y)^2]. \tag{1.4}$$

Namely, the error introduced by our finite data estimator $\widehat{f}_{n,p}$ in excess of the error for the best estimator in the hypothesis space. In the literature, it is also common to refer to the excess risk in $\mathcal{H}$ as the estimation or sample error.

We can relate the approximation error and the estimation error to $\mathcal{E}(f)$ through the decomposition

$$\mathcal{E}(f) = \underbrace{\mathcal{E}(\widehat{f}_{n,\lambda}) - \mathcal{E}(f_{\mathcal{H}})}_{\text{Estimation error}} + \underbrace{\mathcal{E}(f_{\mathcal{H}}) - \mathcal{E}(f_\rho)}_{\text{Approximation error}} + \underbrace{\mathcal{E}(f_\rho)}_{\text{Irreducible error}} \tag{1.5}$$

Here, $\mathcal{E}(f_\rho)$ is normally referred to as the irreducible error. Under the assumption that the noise is additive $Y = f_\rho(X) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma)$, we have that $\mathcal{E}(f_\rho) = \sigma^2$. The decomposition is illustrated in Figure 1.1.

We note that the approximation error is normally thought of as a model bias introduced by the choice of hypothesis space. Meanwhile, the estimation error is a variance term arising due to finite training samples. For a fixed sample size n, reducing the size of $\mathcal{H}$ typically increases the approximation error but reduces the estimation error and vice versa. When tuning the hyper-parameters to avoid overfitting, as discussed in section 1.2, it is the trade-off between these two errors we consider. Figure 1.1 illustrates this when the size of the hypothesis space is controlled by the regularization parameter λ. Note that the size of $\mathcal{H}$ can also be controlled by other hyper-parameters, such as the bandwidth of the Gaussian kernel.

1.4 Kernel methods

Kernel methods are a category of non-linear learning algorithms that rely on a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ to create an implicit non-linear embedding of a point cloud $\mathcal{X}$ into a function space equipped with an inner product. The resulting function space is called a reproducing kernel Hilbert space (RKHS), denoted $\mathcal{H}_k$, and allows the use of linear learning schemes.

Definition 2 (Kernel function) *A kernel function is a symmetric function* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *that satisfies the following property*

$$\sum_{i,j} c_i c_j k(x_i, x_j) \geq 0 \quad \forall x_i, x_j \in \mathcal{X}, \quad c_i, c_j \in \mathbb{R}.$$

Given a dataset $\mathcal{D}_n = \{x_i\}_{i=1}^n$ *the kernel function induces a positive semi-definite (PSD) matrix* $K \in \mathbb{R}^{n \times n}$ *with entries* $K_{ij} = k(x_i, x_j)$*, called the kernel matrix.*

The RKHS induced by the kernel function k is defined as

$$\mathcal{H}_k = \Big\{ f : \mathcal{X} \to \mathbb{R} | f(\cdot) = \sum_{i=1}^{\infty} \beta_i k(\cdot, x_i), \beta_i \in \mathbb{R}, x_i \in \mathcal{X}, ||f||_k < \infty \Big\},$$

with inner product $\langle f, g \rangle_k = \sum_{ij} \alpha_i \beta_i k(x_i, x_j)$ for $f(\cdot) = \sum_i \alpha_i k(\cdot, x_i) \in \mathcal{H}_k$ and $g(\cdot) = \sum_j \beta_j k(\cdot, x_j) \in \mathcal{H}_k$. The corresponding norm is denoted $\|\cdot\|^2 = \langle \cdot, \cdot \rangle_k$. Furthermore, the associated feature functions $\phi_i : \mathcal{X} \to \mathbb{R}$, $\phi_i(\cdot) \in \mathcal{H}_k$ can be evaluated implicitly via the "kernel trick" [2, 19]

$$\phi_i(x) = k(x, x_i) = \langle \phi_x, \phi_{x_i} \rangle_k. \tag{1.6}$$

The RKHS is a popular hypothesis space, as it can represent large classes of functions. In fact, for special types of kernels and certain conditions on $\mathcal{X}$, the corresponding $\mathcal{H}_k$ is universal, meaning it contains all bounded continuous functions on $\mathcal{X}$ [79]. Consequently, working in an RKHS allows for representing highly non-linear functions. Moreover, these functions can be efficiently evaluated in the original domain $\mathcal{X}$ using the kernel function k.

Learning in RKHS The theory underlying function approximation in RKHS is well established in machine learning [54] and learning theory [92]. Because of this, kernel methods are generally considered to be theoretically better understood than non-linear learning schemes, such as multi-layer neural networks and decision trees.

Optimizing Eq. (1.2) over the expansion coefficients $\{\beta_i\}_{i=1}^{\infty}$ is generally infeasible as $\mathcal{H}_k$ is normally infinitely dimensional. Therefore, learning in the RKHS is normally done by seeking a minimizer of Eq. (1.2) on the form

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i), \tag{1.7}$$

provided the optimization problem allows this formulation. Namely, a finite linear expansion in the kernel functions evaluated on the training points $\mathcal{D}_n = \{x_i\}_{i=1}^{n}$. Seeking an estimator on this form corresponds to reducing the hypothesis space to a finite-dimensional subspace $\mathcal{H}_n \subset \mathcal{H}_k$ defined as

$$\mathcal{H}_n = \{ f \in \mathcal{H}_k : f(\cdot) = \sum_{x_i \in \mathcal{D}_n} \alpha_i k(\cdot, x_i), \alpha_i \in \mathbb{R} \}. \tag{1.8}$$

In many settings, it can be shown that the optimal estimator of Eq. (1.2) is contained in $\mathcal{H}_n$. For example, Schölkopf et al. [101] shows that this is the case for a certain class of loss functions L, when $J_{\lambda,n}(f) = g(\|f\|_k)$ and $g : [0, \infty] \to \mathbb{R}$ is a monotonically strictly non-increasing function. Here $\|\cdot\|_k$ is the norm in $\mathcal{H}_k$. This result follows from the representer theorem [101], first introduced by Kimeldorf and Wahba [59, 60], and later extended to more general loss functions L, and regularization terms g by Schölkopf et al. [101].

Learning an estimator in an RKHS is, therefore, reduced to optimizing over a finite set of coefficients $\{\alpha_i\}_{i=1}^{n}$. Consequently, the computational complexity does not depend on the dimension of the feature space but rather on the number of training samples. This is advantageous when the dimension of the feature space is significantly larger than n, which is the case for many RKHS settings [101, 126]. However, in many situations, this approach results in poor scaling with the number of samples n.

Computational considerations Although kernel methods have provable theoretical advantages, they suffer from scalability issues due to high memory requirements and computational costs. Kernel methods rely on constructing an $n \times n$ kernel matrix incurring an $\mathcal{O}(n^2 D)$ cost for evaluating the kernel, where D is the dimension of the ambient space where the point cloud is embedded. Furthermore, solving for the coefficients typically reduces to solving a linear system that involves inversion of the kernel matrix, with a cost of $\mathcal{O}(n^3)$. In addition to this comes the $\mathcal{O}(n^2)$ cost of storing the kernel matrix and additional $\mathcal{O}(nD)$ operations required to evaluate the function at a sample point x.

Kernel ridge regression In the following, we consider a well-known kernel method for supervised learning, namely kernel ridge regression (KRR) [99, 104]. This learning scheme considers the minimization problem in Eq. (1.2) with a squared loss combined with the penalization term $J_\lambda(f) = \lambda \left\| f \right\|_k^2$,

$$\widehat{f}_{n,\lambda} = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \left\| f \right\|_k^2 . \tag{1.9}$$

This has the benefit of giving rise to a convex optimization problem in $\mathcal{H}_k$. It can be shown that the minimizer of (1.9) is of the form

$$\widehat{f}_{n,\lambda}(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$$

where the coefficients are given by the linear system

$$(K + \lambda n I)\alpha = y, \tag{1.10}$$

where $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $y = (y_1, \ldots, y_n)^\top$.

The statistical accuracy of the KRR estimator $\widehat{f}_{n,\lambda}$ is optimal in a min-max sense as measured by the excess risk in $\mathcal{H}_k$, with $R_{\mathcal{H}_k}(\widehat{f}_{n,\lambda}) = \mathcal{O}(n^{-1/2})$ when $\lambda = n^{-1/2}$ [95]. However, solving the linear system in Eq. (1.10) requires constructing the kernel matrix K and storing it in memory, with significant costs as discussed above. Furthermore, direct inversion of $K + \lambda n I$ has a cost of $\mathcal{O}(n^3 + n^2 c_{k,D})$ operations where $c_{k,D}$ is the cost of evaluating the kernel k in the input space $\mathcal{X} \subseteq \mathbb{R}^D$.

In the first paper of this thesis, we develop a novel algorithm that improves the scalability of KRR. Several studies in the literature have been dedicated to this purpose, and we cover some of these methods in more detail in Section 5.

1.5 Boosting

Boosting is a framework for building a composite learner from a set of base learners, with significant generalization improvements over the base learners from which it is derived. First introduced by Freund and Schapire in [43, 44, 100], the boosting framework has shown great success in producing efficient learning algorithms. Following the discussion in Friedman [45, 46] we offer an overview of the boosting framework with a particular focus on gradient boosting with L_2 loss.

Consider the learning setting outlined at the beginning of Section 1. The objective is to find a function f that minimizes the expected risk $\mathbb{E}[L(f(X), Y)]$ for some loss function L, given a dataset $\mathcal{D}_n$. Boosting is similar to ensemble learning algorithms that seek to minimize $\mathbb{E}[L(f(X), Y)]$ using an estimator of the form

$$\widehat{f}_{n,\eta,\gamma}(x) = \sum_{l=1}^{T} \eta_l h(x, \gamma_l),$$

where $h(x, \gamma_l)$ is a set of base learners parameterized by γ_l. The optimal estimator is found by fitting the parameters $\{(\eta_l, \gamma_l)\}_{l=1}^{L}$ to the training data.

What sets boosting apart from other ensemble learning algorithms is the way the parameters are found. With a finite dataset $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^{n}$, boosting proposes the following iterative optimization scheme

$$(\eta^{(l)}, \gamma^{(l)}) = \underset{\eta,\gamma}{\operatorname{argmin}} \sum_{i=1}^{n} L(\widehat{f}^{(l-1)}(x_i) + \eta h(x_i, \gamma), y_i) \tag{1.11}$$

where $h(x, \gamma) \in \mathcal{H}$ are base learners chosen from some hypothesis space $\mathcal{H}$. After each step, the master model $\widehat{f}^{(l)}(x)$ is updated according to $\widehat{f}^{(l)}(x) = \widehat{f}^{(l-1)}(x) + \eta^{(l)} h(x, \gamma^{(l)})$.

The advantage of this iterative approach is that each new base learner added to $\widehat{f}^{(l)}$, is exposed to the generalization error of the previous model. Consequently, new base learners can improve the performance of the estimator where improvement is most needed [40]. Furthermore, boosting is shown to work well for weak learners (simple functions that are easy to fit). The advantage is that less computational resources are required.

Gradient boosting An efficient way to approximate the optimization in Eq. (1.11), is gradient boosting proposed in Friedman [45]. The underlying idea is that the increment $\eta h(x, \gamma)$ at iteration l is a step in the hypothesis space $\mathcal{H}$. Clearly, the increment that minimizes the loss L in Eq. (1.11) the most, is in the direction best aligned with the negative gradient of L, evaluated at the current position in $\mathcal{H}$

$$g_l(x) = \left. \frac{\partial L(f(x), y)}{\partial f(x)} \right|_{f(x) = \widehat{f}^{(l-1)}(x)}$$

With finite data $\mathcal{D}_n$, the gradient direction can be estimated as $g_l = \Big(g_l(x_1), \dots, g_l(x_n)\Big)^{\top}$, where $x_i \in \mathcal{D}_n$. Fitting the base learner $h(x, \gamma)$ to g_l by solving an optimization over γ gives the step direction in $\mathcal{H}$. To find the step length η_l, a new procedure can be run over values of η.

Remark 3 *We note that for the L_2 loss, an explicit expression can be found and shown to be $g_l(x_i) = y_i - \widehat{f}^{(l-1)}(x_i)$. This means that the best base learner at step l is the learner that gives the best fit to the residual after the previous step.*

In the first paper in this thesis, we develop an algorithm that combines KRR, discussed in Section 1.4, with gradient boosting. The algorithm utilizes the fact that the boosting framework works well for weak learners, which allows the kernel at each step to be selected without too much effort dedicated to finding the optimal bandwidth. This allows the algorithm to avoid the expensive hyper-parameter tuning discussed in Section 1.2. The hyper-parameter selection strategy together with the iterative nature of boosting, allows the algorithms to work efficiently with streaming data.

1.6 Alternative methods

Learning in the non-parametric setting is not restricted to kernel methods. Other notable examples are smoothing and multi-variate splines [49], regression trees [69] and neural networks [80]. A comparison is given in Table 1.1.

Neural networks are particularly interesting due to their widespread use and great success in practical applications. In many ways, neural networks can be considered a counterpart of kernel methods; whereas learning from finite data with kernel methods is well understood in learning theory, neural networks have less theoretical support. On the other hand, kernel methods are, in their standard form, prevented from large-scale applications due to their large memory requirements and computational expenses. At the same time, deep neural networks [113] are the current go-to method for big data applications.

Model	Regression trees	Kernel methods	Neural networks
Theoretical foundation	Medium	High	Low
Interpretability	Medium	Medium	Low
Scalability	High	Low	High
Predictive power	Medium-High	High	High

Table 1.1: Comparison of different regression models. Comparison partly based on Table 10.1 in Hastie [51]. We note that although regression trees in their standard form are considered to have poor predictive power [51], boosted regression trees have proven to be very successful, and can also be combined with parallelization [119].

2 The curse of dimensionality

A fundamental challenge when learning a regression function f from a set of known training samples $\mathcal{D}_n$, is that the number of samples n needed to achieve a certain accuracy grows exponentially with the dimension of $\mathcal{X}$. This problem is often referred to as the *curse of dimensionality* [17] and occurs in several big data applications such as medicine [18], neuroscience [5] and time series [122].

We can understand the cause and implications of the curse of dimensionality from two perspectives. The first perspective is geometrical and relates to how the volume of space increases with the dimension. The other perspective comes from statistical learning theory and relates to how optimal estimation rates degrade with increasing dimensionality.

Geometric perspective Consider n samples distributed uniformly in a D-dimensional unit ball. It can be shown that the median distance from the origin of this ball to the closest data point has the following dependency on n and D [51]

$$d(n, D) = (1 - (1/2)^{1/n})^{1/D}.$$

In particular, $\lim_{D \to \infty} d(n, D) = 1$. Consequently, the distance between samples increases exponentially with the dimension (i.e. the density of samples decreases).

The implication when learning a function from a training set $\mathcal{D}_n$ is that high dimensions force extrapolation over large distances; unless we compensate with exponentially more samples. We should therefore expect function estimation to suffer in this regime.

Estimation rates perspective The geometrical perspective is useful for gaining intuition on why the curse of dimensionality occurs. However, to fully appreciate the consequences, it is useful to consider the min-max estimation rates discussed in Section 1.3, as these provide lower bounds on how well a regression function can be estimated from a training set $\mathcal{D}_n$ given a specific distribution class; see Györfi et al. [49] and Novak and Triebel [82] for more details.

For our purposes, it suffices to restate a well-known result that provides a lower bound for most regression problems in $\mathbb{R}^D$. Following Györfi et al. [49] we define the distribution $\mathcal{C}^{q,C}$ in Definition 4.

Definition 4 *Let $\mathcal{C}^{q,C}$ be the class of distributions of the random variables (X, Y), where $X \sim Uni([0,1]^D)$, $Y = f_\rho(X) + \eta$, $f_\rho \in \mathcal{F}^{q,C}$ and the noise $\eta \sim \mathcal{N}(0,1)$ is independent of X. Here $\mathcal{F}^{(q,C)}$ denote the class of all (q, C)-smooth functions $f : \mathbb{R}^D \to \mathbb{R}$, such that for $\alpha \in \mathbb{N}_0^D$ and $q = k + \beta$ we have $|\partial_\alpha f(x) - \partial_\alpha f(z)| \leq C|x - z|^\beta$, for $x, z \in \mathbb{R}^D$, $C \geq 0$, $k \in \mathbb{N}_0$, $|\alpha| \leq k$ and $0 < \beta < 1$; see Györfi et al. [49] for more details.*

It can be shown that the min-max rate for the distribution class $\mathcal{C}^{q,C}$ is

$$\inf_{f_n} \sup_{(X,Y) \in \mathcal{D}^\alpha} \mathbb{E}[(\widehat{f}(X) - f(X))^2] = \mathcal{O}(n^{-\frac{2\alpha}{2\alpha + D}}) \tag{1.12}$$

In other words, the number of samples necessary to achieve a mean square error accuracy of ε grows exponentially with D as $\varepsilon \to \infty$.

Avoiding the curse of dimensionality As we have seen, learning in high dimensions is computationally infeasible due to the large number of samples required. If data is sampled from a distribution supported in a high dimensional space $\mathbb{R}^D$ there is not much we can do. However, data is often supported on lower-dimensional subsets $\mathcal{X}$. Consequently, if this structure can be captured, significant improvements can be made. For example, when data lies on a d-dimensional linear subspace or a d dimensional smooth manifold, the optimal convergence rate from Eq. (1.12) reduces to $n^{\frac{-2\alpha}{2\alpha + d}}$. For $d \ll D$ this can make a substantial difference.

The benefit of reducing the dimensionality of the representation has motivated a vast literature on non-linear dimensionality reduction techniques (NLDR). We discuss some of these schemes in more detail in Section 4.1 with NLDR methods relevant to the work in this thesis.

3 Point clouds and their intrinsic structure

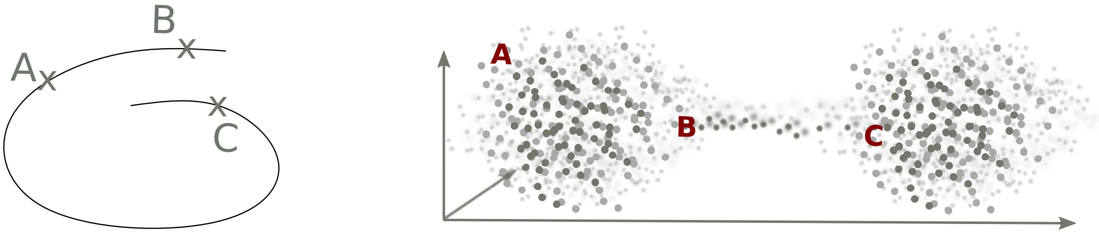
A point cloud is a collection of points sampled from a probability measure $\rho_{\mathcal{X}}$ supported on a lower dimensional set $\mathcal{X} \subset \mathbb{R}^D$, whose intrinsic structure is unknown. In practical applications, data is often modeled as point clouds instead of using more stringent assumptions such as that of a manifold; see Sindhwani et al. [108] and Little et al. [67] with references therein for examples. Furthermore, many well-known NLDR algorithms such as Laplacian eigenmaps [14], diffusion maps [31], and geometric multi-resolution analysis (GMRA) [4] have theoretical guarantees derived under assumptions of a smooth manifold, but are in practice often applied to point clouds with less structural assumptions.

It is common to characterize the intrinsic dimension of point clouds in terms of the covering dimension [84, 123] and the doubling (Assouad) dimension [1, 123].

Definition 5 (The doubling dimension) *(Adapted from Abraham et al. [1]) Let (M, d) be a metric space. The doubling dimension of M is defined as $\textsf{ddim}(M) = \log_2(\kappa)$, where κ is the minimal number of balls of radius $r/2$, required to cover a ball $B_r(x)$ for all $x \in M$ and for all $r > 0$.*

Point clouds can have highly involved and non-linear intrinsic structures, which can impose challenges for algorithms designed to learn patterns in the data. The non-linear structure implies that the Euclidean distance induced by the ambient space $\mathbb{R}^D$ is a poor proxy for measuring distances between samples far apart. Figure 1.2a illustrates this situation. Consider the distance between the points labeled A, B, and C. Using the distance of the ambient space, the distance between A, B is larger than the distance between B, C. However, a more natural distance would be along the swiss-roll, in which case A, B is closer.

Furthermore, the density of the point cloud, encoded in $\rho_{\mathcal{X}}$ might vary across $\mathcal{X}$, and for applications such as clustering, capturing these variations in density is another aspect of importance. For example, consider the distance between A, B, and B, C on the dumbbell distribution in Figure 1.2b. When A, B belongs to the same cluster, it might be more natural that their distance should be smaller than that between B, C.



(a) Distance on swiss-roll

(b) Distances on dumbbell distribution

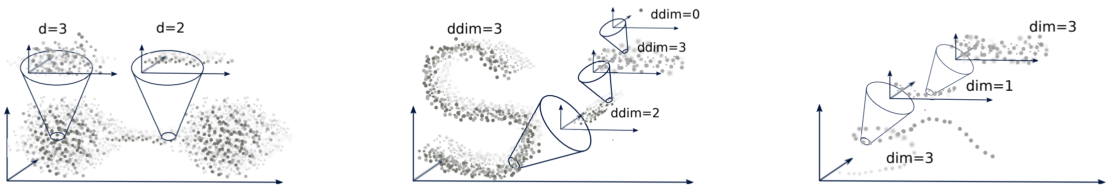
Figure 1.2: Illustration of how the intrinsic structure of point clouds affects what constitutes a natural distance.

Moreover, an important aspect to consider is that the intrinsic dimension of point clouds can vary [33, 67]. Perhaps the most intuitive way that the dimension can vary is between different regions of $\mathcal{X}$. Figure 1.3a illustrates this situation for a dumbbell-shaped point cloud where the spheres are 3-dimensional, and the connecting bridge is a 2-dimensional plane.

A somewhat less intuitive case is when the intrinsic dimension, as measured by the doubling dimension, varies with the resolution at which we consider the problem. Figure 1.3b illustrates this. With a large enough radius r when measuring the doubling dimension, the point cloud will have the dimension of the ambient space. However, with an appropriate resolution (radius), we have a more natural intrinsic dimension, namely a set that is locally approximately 2-dimensional.

The impact of resolution on the perceived dimension has implications for algorithms relying on the ambient distance metric in local neighborhoods. Examples are the kernel radius used to construct the nearest neighborhood graph in spectral graph embeddings [14, 30, 93, 115] and the radius of radial kernels used in kernel methods [104]. In other words, the radius used in these methods has implications for the structure that they see.

Another aspect associated with the resolution is the noise; when the radius is of the same order of magnitude as the noise level, the intrinsic dimension becomes that of the ambient space, as illustrated in Figure 1.3c. In general, there can be regions or scales where the dimension is very close to the ambient dimension. In this case, learning becomes practically infeasible, as discussed in Section 2. In the first paper in this thesis, we propose a strategy that can identify such regions and effectively give up when the dimension is too large, focusing instead on learning in regions where the dimension is lower.



(a) Intrinsic dimension changes with location.

(b) Intrinsic dimension changes with the resolution.

(c) Noise has high intrinsic dimension.

Figure 1.3: Three examples of variation in the intrinsic dimension. The coloring of the point clouds illustrates depth. (a) shows how the intrinsic dimension can change with the location. (b) shows how the intrinsic dimension can change with the resolution scale. (c) shows how the intrinsic dimension of noise is high-dimensional.

4 Unsupervised learning of intrinsic structure

As motivated in Section 3, learning the intrinsic structure of point clouds is of interest for reducing the impact of the curse of dimensionality and defining natural notions of similarity between samples. In the following, we consider some notable approaches relevant to the work in this thesis. In particular, this section will focus on graph-based methods and kernel PCA.

4.1 Graph methods

In graph theory [127], a graph (X_n, E) is a mathematical structure that models relationships between objects called nodes (or vertices) $X_n = \{x_1, \dots, x_n\}$ by assigning edges $E = \{(x_i, x_j) : x_i, x_j \in X_n,\ i \neq j\}$ between pairs of nodes that are connected. The most rudimentary graph has only these two properties. However, other properties are often added to model more sophisticated relationships. For example, edges are often associated with weights W_{ij} to distinguish between different degrees of similarity. The edge weights are typically represented as a matrix $W \in \mathbb{R}^{n \times n}$, called the *weight matrix*, resulting in a weighted graph (X_n, W); see Figure 1.4. Other properties to take into consideration can be the direction of edges, connectedness etc. In this thesis, we will mainly be concerned with undirected weighted graphs with symmetric edges. For discussion on other graph structures, we refer to the literature on graph theory [127].

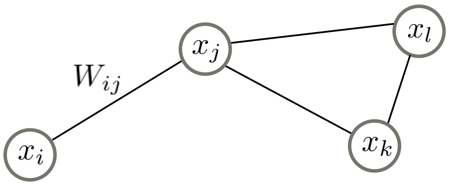


Figure 1.4: Illustration of a simple weighted graph with nodes $X_n = \{x_i, x_j, x_k, x_l\}$.

The structure of graphs makes them well-suited for modeling relationships between entities in complex systems. Examples of use range from social networks [75], biological systems [86] to computer science in general [91]. By modeling these systems as graphs, they can be analyzed efficiently and benefit from the vast literature in graph theory. An important direction in this regard is spectral graph theory [111], which studies graphs and functions on graphs through eigenvectors and eigenvalues of graph matrices. The matrices most typically studied for these purposes are the random walk matrix $A = D^{-1}W$ (also known as the diffusion matrix) and variations of the graph Laplacian $L = W - D$. Here $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix called the degree matrix where the diagonal entry $D_{ii} = \sum_j W_{ij}$ is the degree of node x_i.

In many situations, data is provided as a point cloud $\mathcal{X}$ embedded in some metric space without an explicit graph structure. Before these data sets can be analyzed using techniques from graph theory and spectral graph theory, it is necessary to first represent the point cloud as a graph.

Representing point clouds as graphs Consider a point cloud $\mathcal{X} \subset \mathbb{R}^D$ embedded in some ambient space $\mathbb{R}^D$. We can represent $\mathcal{X}$ as a graph by constructing a local neighborhood graph with edge weights $W_{ij} = k(x_i, x_j)$, for some function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Typically, k is a positive semi-definite function (a kernel function) concentrating in a local neighborhood around each point. Popular choices are the Gaussian kernel $k(x_i, x_j) = \exp(-d(x_i, x_j)^2/\sigma^2)$ and the radial kernel $k_r(x_i, x_j) = \mathbb{1}(d(x_i, x_j) \leq r)$ where $r > 0$ is some fixed radius and $d(\cdot, \cdot)$ is the metric of the ambient space.

Definition 6 (Local neighbourhood graph) *Let (M, d) be a compact metric space and let $\mathcal{X} \subseteq M$. Let $X_n = \{x_1, \ldots, x_n\} \sim \rho_{\mathcal{X}}$ be a set of points sampled from a distribution $\rho_{\mathcal{X}}$ supported on or concentrated around $\mathcal{X}$. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function. The local neighborhood graph is the weighted graph defined by (X_n, W) where X_n are the graph nodes and $W_{ij} = k(x_i, x_j)$ are the edge weights.*

The intuition for this construction is that the neighborhood graph is a discrete approximation of the point cloud under the assumption that the distance induced by the ambient space is meaningful in small neighborhoods around each point. This is further supported by results from manifold learning [55], which is concerned with reconstructing the underlying low-dimensional structure of point clouds that lie on or close to a lower-dimensional manifold. For example, consider a point cloud $\mathcal{X} \subseteq \mathbb{R}^D$ supported on a lower-dimensional manifold. It can be shown, under certain assumptions, that the eigenvectors of the associated graph Laplacian approximate the Laplace-Beltrami operator on the manifold [13, 52]. Similarly, it can be shown that the graph geodesic, the shortest path between two nodes in a graph, converges to the geodesic on the manifold [115].

In the following, we discuss some of the ways graph methods can capture the intrinsic structure of point clouds.

Random walks and Diffusion maps Perhaps the simplest method used to capture the structure of graphs is that of the graph random walk [70]. The random walk on a graph is a process that, at each time step t transitions between graph nodes with a certain probability. In its simplest form, if at time t we are at node i, then the random walk transitions to node x_j with probability

$$p_{ij} = W_{ij}/D_{ii}$$

where W_{ij} is the edge weight between nodes x_i, x_j and D_{ii} is the degree of node x_i. The sequence of steps induced by the random walk is a Markov chain, and many properties of graph random walks can be derived from the study of Markov chains.

The random walk matrix $A = D^{-1}W$ encodes the random process, such that the i, j-th entry of A^t is the probability that we reach node j after t steps, starting at i. Let $p_t \in \mathbb{R}^n$ be a distribution on the nodes at step t. The probability of being at any node $x \in X_n = \{x_i, \ldots, x_n\}$ in the graph can then be encoded by the equation $p_t = A^t p_0$, where p_0 is the initial distribution. The stationary solution of this process is $p_{t'+1} = p_{t'} = p$, meaning that $p = Ap$. The stationary solution p is, therefore, an eigenvector of A with eigenvalue 1. It is easy to verify that p is the largest eigenvalue solution [70].

Stationary solutions of the random walk matrix have been utilized in many applications where the eigenvector entries assign a score to each node. Page rank [21, 85] uses a stationary solution of a modified Markov process, following a specific initial distribution p_0, to rank web pages. Meanwhile, label propagation [130] uses a trap or "ground" (i.e., a node with zero escape probability) to generate a stationary distribution with a decay towards nodes with a different label. For example, with two classes A, and B, the nodes with the label A act as a ground for random walks from the nodes labeled B and vice versa.

In the fourth paper in this thesis, we utilize the stationary solution of a particular transition matrix on a nearest neighbor graph to uncover the structure in a point cloud. The transition matrix can be interpreted as a modified random walk matrix, subject to the effect of a universal ground, that imposes a termination probability to each step in the walk. This imposes a gradual decay in the probability of walking away from the source. The decay depends on the intrinsic graph structure and the "strength" of the ground.

The diffusion maps (DM) algorithm [30] is closely related to the random walk matrix. Diffusion maps provide an embedding of a point cloud into a k-dimensional Euclidean space, that allows distance between points to be calculated using the Euclidean distance in the embedding space. Diffusion maps construct a nearest neighbor graph on the point cloud in question and define a random walk matrix on this graph. It then creates an embedding from the first k eigenvectors of the t-th power of the random walk matrix. The spectral decay determines the number of eigenvectors used in the embedding. In the next section, we discuss diffusion maps in relation to spectral embeddings.

Spectral embeddings Using eigenvectors of graph matrices, spectral embeddings aim to find a lower dimensional representation of point clouds that preserve the intrinsic structure. These methods are often referred to as manifold learning methods as they derive their justification from this perspective. However, they are often applied successfully to point clouds that do not satisfy strict manifold assumptions [55].

The main application of these techniques is non-linear dimensionality reduction (NLDR) [63, 120] where they are used to overcome the curse of dimensionality, as discussed in Section 2. However, spectral embeddings have also been used successfully for several other purposes such as spectral clustering [112, 124] and semi-supervised learning [15].

Notable methods in this category are ISOMAP [115], locally linear embedding (LLE) [93], Laplacian eigenmaps (LE) [14, 16], and diffusion maps [31]. These methods all rely on a similar pipeline which involves constructing a local neighborhood graph using a PSD kernel and then defining an $n \times n$ matrix on this graph. The last step is the spectral embedding step, where the lower dimensional embedding is found by calculating the eigenvectors of the $n \times n$ matrix [55]. As an example, following Belkin and Niyogi [14] the d-dimensional LE embedding of the graph nodes X_n can be defined as

$$x_i \mapsto (v_1(x_i), \ldots, v_d(x_i))^\top \tag{1.13}$$

where $v_l : X_n \to \mathbb{R}$ is the eigenfunction corresponding to the i-th smallest eigenvalue of L. Note that the eigenfunction of $\lambda_0 = 0$ is not included as it is constant for all nodes.

The main difference between the methods lies in the matrix used for the spectral embedding. For example, ISOMAP constructs an $n \times n$ distance matrix on the graph by considering the shortest path between the nodes and then finds a lower dimensional embedding through multi-dimensional scaling, which involves calculating the eigenvalues of this distance matrix. Meanwhile, LE relies on a spectral decomposition of the graph Laplacian, which, similarly to the ER distance, incorporates all possible paths between points [50]. Due to the instability of the shortest path distance w.r.t noise, the LE is generally more stable [14]. Furthermore, assuming that the point cloud is sampled from a manifold, the LE can be shown to converge to the corresponding Laplace-Beltrami operator [13, 52].

The major problem with spectral methods is the computational cost of the eigenfunction calculations. Furthermore, eigenfunction calculations are typically difficult to parallelize and often exhibit **global** behavior, see Definition 7, which means that all data points must be available. Another drawback of these methods is that they are incompatible with streaming, as calculating the embeddings for new data requires the entire procedure to be re-run [55].

Definition 7 (Global and local functions) *Let $f : X_n \to \mathbb{R}$ be a function defined on the nodes in the graph (X_n, W). We say f is* ***global*** *if $|f(x_i)| > \eta$, $|f(x_j)| > \eta$ for $d(x_i, x_j) > r$ for some large η. We say f is* ***local*** *if for any nodes where $d(x_i, x_j) > r$ we have that $|f(x_i)| < \zeta$ or $|f(x_j)| < \zeta$ for some sufficiently large r and sufficiently small ζ.*

In the fourth paper of this thesis, we construct an embedding using localized functions, which are cheaper to compute than global functions and can be computed independently, which allows parallelization and distribution. This scheme is also compatible with streaming data, as the localized functions can easily be extended to new samples.

Quantifying similarity between samples A fundamental problem when learning from point clouds is defining a natural notion of distance between samples. As we have seen, the metric of the ambient space is, in most cases, only suitable locally. However, this allows using a graph as a discrete approximation of the point cloud in question and then defining more suitable distances on this graph. In the following, we discuss two important distance metrics often encountered in the literature.

Perhaps the most natural distance is the graph geodesic [22], which corresponds to finding the shortest path between two nodes in a graph. The geodesic can be found using efficient algorithms such as Dijkstra's algorithm and can be shown to converge to the geodesic on the manifold as the number of samples grows to infinity [115]. However, the shortest path distance is unstable, especially when the data is not on a manifold, and small amounts of noise can dramatically affect the result [12].

An alternative to the graph geodesic is the effective resistance distance (ER) [61]. The ER between two graph nodes $x_i, x_j \in X_n$ is normally defined as

$$R_n(x_i, x_j) = (e_i - e_j)^\top L^\dagger (e_i - e_j), \tag{1.14}$$

where $L^\dagger$ is the pseudoinverse of the graph Laplacian and $e_i \in \mathbb{R}^n$ is the basis vector for node x_i, with 1 at the i-th entry and zero otherwise.

The ER distance differs from the graph geodesic by considering all possible paths between two points instead of only the shortest path. This makes ER more stable to noise and enables capturing cluster structures; namely, tightly connected regions of the graph have a smaller ER distance than loosely connected regions. The problem with ER is that it has been shown to converge to a trivial limit in the large graph limit [71, 125], which means it does not scale very well to big data applications. In the third paper in this thesis, we look into ways to overcome this issue in order to extend ER to large graphs.

4.2 Graphs as resistor networks

A resistor network (or resistor graph) is a conceptual framework that allows working with graphs using analogies to electrical circuits. A graph (X_n, W) can be thought of as an electrical network where the nodes are connected by resistors $R_{ij} = 1/W_{ij}$. It follows that a function on the graph nodes $v : X_n \to \mathbb{R}$ can be interpreted as a voltage $v(x_i)$, which induces a current through Ohm's law such that

$$v(x_i) - v(x_j) = R_{i,j} J_{i,j}, \quad \text{or alternatively} \quad J_{i,j} = W_{ij}(v(x_i) - v(x_j)). \tag{1.15}$$

Combining this with Kirchoff's current law, which states that the sum of currents entering a node i must be zero, one can define the *energy* of the voltage

$$E(v) := \sum_{x_i, x_j \in X_n} W_{i,j}(v(x_i) - v(x_j))^2 = v^T L v. \tag{1.16}$$

The voltage function that minimizes the energy can be used to uncover information about the graph. However, in an unconstrained system, the minimizer is trivial as zero energy can be obtained for any voltage function for which all entries are equal, namely $v(x_i) = v$ for all $x_i \in X_n$. Consequently, it is necessary to introduce constraints on the system. This is normally done by imposing conditions on the voltage or the current flow in the system. Several graph embeddings can be shown to correspond to minimizers of the energy under different constraints. As examples, we consider Laplacian eigenmaps and effective resistance.

Laplacian eigenmaps Following Belkin and Niyogi [14], the coordinate functions $\{v_l\}_{l=1}^d$ of the d-dimensional LE embedding in Eq. (1.13) can be shown to be the solutions of the energy minimization problem

$$\begin{aligned} \min_{v_l : X_n \to \mathbb{R}} \quad & \sum_{x_i, x_j = 1}^{n} W_{ij}(v_l(x_i) - v_l(x_j))^2 \\ \text{subject to} \quad & v_l \perp v_{l'} \quad \text{and} \quad v_l \perp 1. \end{aligned} \tag{1.17}$$

for $l = 1, \dots, d$. The constraint $v_l \perp \mathbf{1}$ is introduced to avoid the trivial solution while $v_l \perp v_{l'}$ ensures that the voltage solutions are orthogonal. Here $\mathbf{1} \in \mathbb{R}^n$ is the vector of all ones. Meanwhile, it is worth noting that there are no constraints imposing any locality of the voltage functions. This explains why the eigenfunctions of LE can often be global, as discussed in Section 4.1. Furthermore, the orthogonality condition prevents calculating the eigenfunctions independently. Consequently, this limits the use of distribution and parallelization to solve the minimization problems.

In the fourth paper in this thesis, we address this by introducing constraints that force the voltage function to be local, in the sense of Definition 7. Furthermore, we do not require individual voltage functions to be orthogonal, which allows them to be solved for independently.

Effective resistance Similarly to LE, effective resistance can also be formulated as an energy minimization problem. Following Jørgensen et al. [58] the ER between two graph nodes $x_i, x_j \in X_n$ can be defined as in Proposition 8.

Proposition 8 *The effective resistance between nodes x_i, x_j corresponds to $R_n(x_i, x_j) = 1/J_{tot}$, where*

$$J_{tot} := \sum_{j \in \{1,\dots n\}} W_{ij}(v^*(x_i) - v^*(x_j))$$

and v^* is the function that minimizes the Dirichlet energy

$$\begin{aligned} \min_{v: X_n \to \mathbb{R}} \quad & \sum_{x_i, x_j \in X_n} W_{i,j}(v(x_i) - v(x_j))^2, \\ \textit{subject to} \quad & v(x_i) = 1, \quad v(x_j) = 0 \end{aligned}$$

Proof See Theorem 4.2, in Jørgensen and Pearse [58]. ■

In Proposition 8, the solution is constrained using Dirichlet conditions on the voltage. However, several equivalent formulations exist, as shown in Jørgensen and Pearse [58]. Solving a minimization problem with constraints on the current, it is easy to show that the solution corresponds to the ER defined in Eq. (1.14).

4.3 Kernel methods revisited

Kernel methods and learning in RKHSs are not restricted to regression and supervised learning. Since $\mathcal{H}_k$ is a linear space equipped with an inner product, the distance between features $\phi_i \in \mathcal{H}_k$ is straightforward to compute. At the same time, from Eq. (1.6) it follows that the kernel function can be thought of as a non-linear similarity measure on the original set $\mathcal{X}$. Consequently, the kernel embedding allows samples $x_i, x_j \in \mathcal{X}$ to be compared implicitly through the inner product in $\mathcal{H}_k$, without knowing a natural notion of distance in the original space $\mathcal{X}$. In the following, we discuss a method that utilizes these observations to extend traditional PCA to non-linear structures.

Kernel principal component analysis A notable kernel method for unsupervised learning is kernel principal component analysis (kernel PCA) [102, 103]. Kernel PCA generalizes standard PCA to point clouds with non-linear intrinsic structures. The intuition is that the linear structure of $\mathcal{H}_k$ allows standard PCA to be applied.

After centering the data in the feature space, it can be shown that the p-th principal component is given as

$$v_p(x) = \frac{1}{\sqrt{\lambda_p}} \sum_{i=1}^{n} \alpha_{p,i} k(x, x_i),$$

where (λ_p, α_p) is the p-th eigenvalue-eigenvector pair of the kernel matrix $K_{ji} = k(x_i, x_j)$, and

$$K\alpha = m\lambda\alpha.$$

We see that kernel PCA, similarly to KRR, is reduced to solving a linear system, and the solution is expressed as a linear expansion in the kernel evaluated at the training samples. We note that, as stated by Schölkopf et al. [101], kernel PCA can also be thought of as a minimization problem on the form (1.2).

It is clear that kernel PCA suffers from the same scalability issues as KRR. Solving the linear system requires calculating the eigenvalues of the kernel matrix, which in general, requires $\mathcal{O}(n^3)$ operations. Furthermore, kernel PCA requires $\mathcal{O}(n^2)$ memory to store the kernel matrix and needs $\mathcal{O}(nD)$ operations to project new samples x onto the principal components.

We mention that kernel PCA gives an elegant connection between kernel methods and the spectral embedding methods discussed in Section 4.1. In particular, it is shown by Ham et al. [50] that ISOMAP, LLE, and the Laplacian eigenmaps can all be interpreted as kernel PCA for a particular choice of kernel. For example, Laplacian eigenmaps can be considered as performing kernel PCA with the pseudoinverse of the graph Laplacian, which is closely connected to commute times and the ER distance on graphs.

4.4 Alternative methods

The task of learning intrinsic structures in point clouds has been addressed in several other works in the literature, beyond kernel methods and graph-based methods. A notable approach is the geometric multi-resolution analysis (GMRA) developed in a series of works [4, 64, 65, 72]. Another notable example is local tangent space alignment [131].

5 Scaling kernel-based learning for big data

For most real-world applications, it is necessary to use algorithms that can handle large amounts of data in a resource-efficient manner. Furthermore, the scale of modern data sets has motivated the development of powerful computational models such as streaming, parallelization, and distributed systems. To fully utilize the potential of these computational models, it is essential to develop algorithms that can operate and work with data in the way the computational models demand.

As we have seen in Section 1.4 and 4.1, graph-based methods such as spectral embeddings and kernel methods such as KRR and kernel PCA are, in their basic form, prevented from large-scale applications due to their considerable memory requirements and computational costs. In particular, we have seen that these methods rely on the construction of large $n \times n$ kernel matrices and on expensive inversions and spectral decompositions of these matrices. Furthermore, these methods are not optimized for powerful computational models such as streaming, parallelization, and distribution.

In the following, we start by reviewing the requirements on data handling imposed by computational models such as streaming, parallelization, and distribution. We then review several techniques rooted in randomized numerical linear algebra [76] used to reduce the size of kernel matrices for the purpose of improving time and memory usage. We also review iterative approaches for efficiently solving eigenvalue problems and matrix inversion problems.

5.1 Computational models

Modern data sets are often prevented from loading into memory in their entirety due to their sheer size or because they are provided only through a continuous stream of examples. Furthermore, computational models such as the streaming model of computation, parallelization, and distribution enforce their own requirements on how data can be managed. To fully utilize the potential of parallelization and distribution and to enable learning from data streams, it is necessary to be aware of the requirements of these computational models when designing learning algorithms.

In the following, we will give a brief overview of these computational models and their requirements.

Streaming model of computation A streaming model of computation [81] is necessary for data prevented from being made available in memory in its entirety. This could be because the data is too large to be kept in memory and, therefore, must be loaded incrementally or in batches. Alternatively, it could be because the data is recorded and made available continuously. The development of learning algorithms to handle streaming data is of great interest in machine learning, as shown by recent reviews by Gomes et al. [47] and Bahri et al. [10], due to the rapid increase of data exhibiting such requirements in big data applications.

Streaming algorithms read data as a single sample or a mini-batch at a time and incorporate it into the learning model. After processing a sample, the algorithm discards it to limit the data kept *in-memory*. Because of the large data size, possibly infinite, the computational complexity of operations performed in-memory can not scale with the data size. In general, this is handled by storing only a sketch of the data in-memory. The size of the sketch is usually significantly smaller or even independent of the size of the data itself. Updates to the learning models either happen incrementally with each new sample, or batch-incrementally with a batch of new samples [10].

In the first paper of this thesis, we develop a learning scheme for KRR that is designed for the streaming setting.

Parallelization and distributed computations Parallelization and distributed computations are closely linked computational concepts but differ in some vital aspects [11]. The fundamental difference between the two is that parallel computing utilizes several processors typically sharing memory on the same computer. Meanwhile, distribution refers to computations performed at independent computers often provided through cloud services that do not have access to the same memory. Parallelization can be used to greatly increase the utilization of computer resources, while distribution allows the use of clusters to significantly scale the available resources beyond one computer.

In the fourth paper in this thesis, we develop an embedding scheme based on localized functions that can be calculated independently and do not require access to all data simultaneously, therefore allowing parallelization and distribution.

5.2 Matrix approximation techniques

When dealing with large positive semi-definite matrices, a common strategy for reducing the computational expense is to find a low-rank approximation that can replace the original matrix. This is part of a more general question, closely studied in numerical linear algebra [76], on how to find a good spanning subset of rows or columns for a given matrix. The motivation is that many matrices have singular values that decay fast. Therefore, in principle, it should be possible to approximate such matrices by a subset of basis vectors. The best rank-one approximation is clearly the leading singular vector, but as the main goal is to speed up computations, using the singular vectors is often not an option.

In numerical linear algebra, there are several methods designed for low-rank approximations of matrices. A discussion on some of these techniques can be found in Mahoney et al. [74] and in Bach [9] with particular emphasis on kernel methods. However, we will concentrate on two specific approximation techniques, namely Nyström sub-sampling [128] and random features [90], which have been particularly successful in the context of kernel methods. We mention that several other matrix approximation strategies exist. An example is block kernel approximation, which utilizes the clustering structure of kernel matrices to approximate the matrix. Another notable example is memory-efficient kernel approximation, which utilizes low-rank structures in the matrix in addition to the cluster structure to enhance the approximation further [107].

Nyström sub-sampling The main idea of the Nyström sub-sampling is to create an approximation of a PSD matrix $A \in \mathbb{R}^{n\times n}$ using a subset $\widetilde{\Gamma} \subset \Gamma = \{1,\dots n\}$ of the matrix column indices. The Nyström approximation was proposed simultaneously by Williams and Seeger [128] and Smola [109], the main difference being the column selection strategy. In its fundamental form, the Nyström approximation can be written as

$$\widetilde{A}_{nn} = A_{nm}A_{mm}^{\dagger}A_{mn},$$

where $A_{nm} = A(\Gamma,\widetilde{\Gamma}) \in \mathbb{R}^{n\times m}$, and similarly $A_{mm} = A(\widetilde{\Gamma},\widetilde{\Gamma}) \in \mathbb{R}^{m\times m}$, are constructed on the selected subset of columns. We note that the error in the approximation is elegantly connected to the Schur complement as $A_{nn}/A_{mm} = A_{nn} - A_{nm}A_{mm}^{\dagger}A_{mn}$.

The use of the Nyström approximation is useful in many applications involving spectral decompositions and inversion of A. A typical application is where the inversion of $A + \lambda I$ is necessary. For example, KRR requires solving a linear system of the form $(K + \lambda I)\alpha = y$, where $K \in \mathbb{R}^{n\times n}$ is the kernel matrix and $y \in \mathbb{R}^n$ is the labeled training samples. Using the Nyström approximation and the Woodbury formula we get

$$\alpha = \frac{1}{\lambda}\Big(y - K_{nm}(\lambda K_{mm} + K_{mn}^{\top})K_{nm})K_{mn}y\Big),$$

which can be solved with $\mathcal{O}(m^2 n)$ computations and $\mathcal{O}(nm)$ memory [128]. Other more advanced solution techniques have also been used, the most notable in terms of KRR is FALKON developed by Rudi et al. [95], which combines sub-sampling with an iterative solver and a sub-sampled preconditioner.

In the context of kernel methods, each column corresponds to a feature vector $\phi_i = k(\Gamma, x_i)$. The column sub-sampling can therefore be thought of as selecting a smaller hypothesis space $\mathcal{H}_m \subseteq \mathcal{H}_n \subseteq \mathcal{H}_k$, spanned by the features associated with the selected columns. This is similar to the restriction introduced by $\mathcal{H}_n$ in Eq. (1.8). Furthermore, since each column is associated with a specific sample in the original domain $\mathcal{X}$, the column sub-sampling can be thought of as selecting a subset of the training samples $\widetilde{\Gamma} = \{\widetilde{x}_i\}_{i=1}^m \subset \mathcal{D}_n$, often referred to as Nyström centers. The estimator is expressed as a linear expansion in the kernels centered at these Nyström centers

$$\widehat{f} = \sum_{\widetilde{x}_i \in \widetilde{\Gamma}} \widetilde{\alpha}_i k(\cdot, \widetilde{x}_i) \in \mathcal{H}_m.$$

Algorithms based on the Nyström approach differ in the way they select the subset of columns. In the following, we review some popular selection schemes.

Randomized sub-sampling: The approach normally associated with Nyström approximation is random sub-sampling, where sub-samples are selected uniformly at random without replacement. This is the original strategy proposed by Williams and Seeger [128]. Despite its simple nature, it has proven to provide good approximations and is especially attractive as it requires no extra computations and is easy to analyze theoretically [9, 94, 95]. We note that random sub-sampling is often referred to as Nyström sub-sampling, although the Nyström approximation is not restricted to the random sub-sampling choice.

A problem with random sub-sampling is that important columns can be missed. For example in regions with few available samples, random sub-sampling might miss out on sampling columns from these regions entirely. Meanwhile, regions with a high density of samples can have too much influence.

Leverage scores sub-sampling: Mahoney and Drineas [73] introduced the concept of leverage scores as a way to ensure that important columns are sampled, whereby important we mean columns that have a proportionally large effect on the low-rank fit. The fundamental idea is to create a probability distribution that reflects the importance of columns and then sample the columns according to this distribution. Alaoui and Mahoney [3] extended this concept to kernel ridge regression, introducing leverage scores tailored to this setting. In this formulation, each training sample $x_i \in \mathcal{D}_n$ is associated with a leverage score

$$l(x_i) = (K(K + tnI)^{-1})_{ii} \quad \text{and probability} \quad p_i = l(x_i)/\sum_{i=1}^{n} l(x_i).$$

The columns are sampled with probability $p = (p_1, \ldots, p_n)$. The challenge with leverage scores is that they are expensive to compute, therefore approximations are typically used instead [29, 38, 94–96]. An efficient algorithm for approximating leverage scores can be found in Rudi et al. [96].

Non-probabilistic sub-sampling: The sub-sampling schemes reviewed above rely on a probabilistic approach to sub-sampling. However, sub-sampling strategies for low-rank matrix approximation are not limited to this setting. In particular, Smola [109] proposed a greedy strategy, relying on the pivoted Cholesky method, which iteratively searches for optimal columns. Meanwhile, Fine and Scheinberg [41] proposed a method based on incomplete Cholesky factorization. These non-probabilistic techniques give better approximations than their probabilistic counterparts but have in general larger computational expenses and are harder to analyze [9, 76].

Random features The idea behind random feature approximation of kernel matrices is rooted in the *empirical approximation method* found in approximation theory and randomized linear algebra [36, 37, 76]. The main idea is as follows, assume that we have a low-rank random matrix $Z \sim \rho_Z$ sampled from some distribution ρ_Z such that its expectation equals the matrix we want to approximate, namely $A = \mathbb{E}[Z]$. It follows that the empirical mean is a good estimator of A

$$\widetilde{Z}_m = \frac{1}{m}\sum_{i=1}^{m} Z_i,$$

where each Z_i is sampled i.i.d from ρ_Z.

In Rahimi and Recht [90], this approximation strategy was introduced for kernel matrices along with the concept of random features. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function, and let $K \in \mathbb{R}^{n\times n}$ with $K_{ij} = k(x_i, x_j)$ be the associated kernel matrix constructed on the data set $\mathcal{D}_n = \{x_i\}_{i=1}^n$. The random feature approach is concerned with finding a bounded function $z(x, w) : \mathcal{X} \times \mathcal{W} \to \{z \in \mathbb{C}, \|z\| \leq \infty\}$, sampled according to some distribution ρ_W defined on $\mathcal{W}$, such that

$$k(x_i, x_j) = \int z(x_i, w)z(x_j, w)^* d\rho_{\mathcal{W}} = \mathbb{E}[z(x_i, w)z(x_j, w)^*].$$

If $z(x, w)$ is known, the kernel matrix can be approximated by forming the random matrix $Z(w) = z(w)z(w)^*$, whose expectation is $K = \mathbb{E}[Z]$. Here $z(w) = \Big(z(x_1, w), \quad \ldots, \quad z(x_1, w)\Big)$ is called a *random feature*. The existence of a function $z(x, w)$ that satisfies this property is the case for several PSD kernels. For example, for the Gaussian kernel, we have $z(x, w) = \exp(i\langle x, w\rangle)$ where $w \sim \mathcal{N}(0, \sigma^{-2})$ [76]. The cost of this approximation is $\mathcal{O}(nmd)$ where d is the dimension of $\mathcal{X}$.

Computational remarks When using the approximation techniques described above, in the context of KRR, it is of interest to characterize the number of sub-samples m necessary to maintain the statistical accuracy of standard KRR.

Consider a hypothesis space $\mathcal{H}$ and the optimal statistical accuracy measured in terms of the excess risk in $\mathcal{H}$, namely $\mathcal{R}_{\mathcal{H}}(f)$ as defined in Eq. (1.4). The standard KRR estimator $\widehat{f}_n$ from Eq. (1.9), achieves the rate $\mathcal{R}_{\mathcal{H}}(\widehat{f}_n) = \mathcal{O}(n^{-1/2})$ when $\lambda = n^{-1/2}$, which is optimal statistical accuracy in a min-max sense according to Rudi et al. [95]. Table 1.2 (adapted from Rudi et al. [95]) compares this to alternative implementations of KRR for parameter choices that give the same optimal statistical accuracy.

KRR solver	Training time	Kernel evaluations	Memory	Test time
Standard KRR	n^3	n^2	n^2	n
Random features [90]	n^2	$n\sqrt{n}$	n	$\sqrt{n}$
Nyström [109, 128]	n^2	$n\sqrt{n}$	n	$\sqrt{n}$
Nyström iterative [24]	n^2	$n\sqrt{n}$	n	$\sqrt{n}$
FALKON [95]	$n\sqrt{n}$	$n\sqrt{n}$	n	$\sqrt{n}$

Table 1.2: Table adapted from Table 1 in Rudi et al. [95]. Computational and memory requirements to achieve optimal learning rates for KRR

We note that further computational improvements to KRR can be achieved by introducing iterative methods, such as gradient descent with early stopping, to approximate the solution of $(K + \lambda nI)\alpha = y$ from Eq. (1.10) [24]. FALKON proposed in Rudi et al. [95], takes this a step further by introducing a preconditioner also subject to Nyström-based sub-sampling, which further reduces computational and memory requirements while maintaining the same optimal statistical accuracy.

Utilizing GPU acceleration and parallelization, Meanti et al. [78] demonstrate that FALKON can be applied efficiently to large-scale datasets with billions of points. However, despite its success in application to large-scale data sets, FALKON still requires the selection of an optimal bandwidth and does not incorporate an efficient way to select this bandwidth in a streaming framework, where cross-validation techniques are inapplicable. The same can be said about Nyström sub-sampling techniques based on probabilistic schemes such as random sub-sampling and leverage scores, which require access to the available data to determine the optimal number of sub-samples m, which again might depend on the bandwidth. FALKON does not address the issue of selecting these samples in a streaming environment.

In the first paper in this thesis, we develop a novel algorithm utilizing a modified version of FALKON as the core solver. The algorithm introduces a novel sub-sampling and bandwidth selection scheme to extend the KRR approach to streaming data. Another notable effort to extend KRR to the streaming setting is the multi-kernel online learning scheme proposed by Shen et al. [106]. This algorithm utilizes a random feature-based approach for matrix approximation and combines this with an iterative gradient descent-based update of the expansion weights.

5.3 Sparse eigensolvers

Finding the eigenvalues and eigenvectors of large matrices is an expensive operation. For $n \times n$ matrices, the cost is $\mathcal{O}(n^3)$. However, when additional information is known about the matrix, this cost can be significantly reduced. For example, utilizing sparsity in a matrix can reduce both memory and computational requirements. In this section, we discuss how sparsity can reduce expenses involved with finding the eigenvector of the leading eigenvalue, in terms of the power method.

Sparse matrices An $m \times n$ matrix is said to be sparse if it has $\mathcal{O}(\min(m, n))$ non-zero elements [116]. Meanwhile, a matrix that has very few non-zero elements is referred to as dense. In the modern era, sparse representation of matrices is available in most programming languages, and the advantage of working with these representations is as follows:

- *Memory*: Large matrices can be stored in a compressed form where only the non-zero elements with their associated indices are stored.
- *Computational*: Time is saved if only operations with non-zero elements are performed.

An important application of sparse representations is the power method and finding the largest eigenvector of sparse matrices.

Power method The power method [76], also known as power iteration, is an iterative method in the family of Krylov subspace methods, for finding the leading eigenvalue λ_{max} or eigenvector of a positive semi-definite matrix. Let $A \in \mathbb{R}^{n\times n}$ be a PSD matrix. Starting with an initial guess $v_0 \in \mathbb{R}^n$ the power method generates a sequence of eigenvector estimates

$$v_t = \frac{Av_{t-1}}{\|Av_{t-1}\|_2}, \quad \text{with associated eigenvalue estimates} \quad \eta_t = v_t^\top A v_t,$$

for $t \geq 1$. We note that this is similar to the procedure of finding the stationary solution of the random walk matrix, discussed in Section 4.1.

For each iteration t, the power method calculates the matrix-vector product Av_t, if the matrices are sparse, the computational cost of these operations can be significantly reduced. Further improvements can be made if the vector v_t is also sparse [8]. However, despite this, there is still the need to construct and index a large $\mathbb{R}^{n\times n}$ sparse matrix.

In the fourth paper in this thesis, we suggest a scheme that ensures that the vector v_t is spatially localized in the underlying graph, see Definition 7. Consequently, it is possible to work with only a sub-matrix of A. For large sample sizes n, this can have significant improvements as one only needs to construct the matrix on a subset of the data.

6 Manuscript contributions

In this section, we present four original manuscripts developed in this thesis. The first two manuscripts are concerned with regression in a supervised setting. The last two manuscripts are concerned with uncovering the intrinsic structure of point clouds in an unsupervised setting. The first paper presents a novel KRR solver StreaMRAK. The second paper demonstrates StreaMRAK as a tool for predicting ionic membrane currents from cardiac action potential traces. The third paper proposes a new definition of effective resistance to alleviate the convergence issues encountered by the standard definition. The fourth paper demonstrates a new embedding strategy for point clouds.

6.1 Paper I: StreaMRAK

Kernel ridge regression allows the learning of highly non-linear functions. The success of this method has been demonstrated in many applications and is supported by a well-established theoretical foundation [54, 92, 101, 104]. However, KRR, like other kernel-based learning algorithms, suffers from large memory requirements and high computational costs. These costs arise because learning with KRR involves solving a linear system $(K + \lambda nI)\alpha = y$ for the coefficients $\alpha \in \mathbb{R}^n$, where the kernel matrix $K \in \mathbb{R}^{n\times n}$ grows with the number of samples.

Efforts to overcome computational expenses and large memory requirements have focused on reducing the size of the kernel matrix with sub-sampling techniques such as Nyström approximations and random features. Furthermore, using scalable iterative methods to solve the linear system $(K + \lambda nI)\alpha = y$ have been shown to significantly cut down the computational costs. These efforts have led to many capable KRR solvers, as summarized in Table 1.2. A prominent example is FALKON, developed by Rudi et al. [95], which has been demonstrated to work efficiently with massive datasets [78].

Manuscript contribution We develop a novel kernel-based learning algorithm called StreaMRAK, for the purpose of extending KRR to the streaming computational model. The contributions of this algorithm can be summarized as

1. Efficient use of samples.
2. Efficient selection of hyper-parameters.
3. Compatibility with streaming.
4. A novel way to mitigate the curse of dimensionality.

The motivation for the algorithm is that in a streaming computational model, we expect large amounts of data to arrive sequentially or in batches. However, computational resources and memory are limited. In light of this, StreaMRAK has been designed to make efficient use of samples, only storing samples as long as they are needed and then discarding them.

Furthermore, the use of cross-validation to optimize the kernel bandwidth is cumbersome in a streaming setting. Therefore, StreaMRAK implements a multi-resolution approach to learning that consists of two parts, a novel sub-sampling scheme combined with an efficient bandwidth selection strategy. Together, these methods adapt the bandwidth and sub-sample density to the resolution level in a data-driven manner. The benefit is that expensive optimization over the bandwidth hyper-parameter is avoided, although at the cost of not finding the best possible bandwidth at each level. The sub-sampling part of the multi-resolution approach is formulated as a pyramid, starting at a low-resolution level l, with few sub-samples from the data, it gradually increases the resolution and number of sub-samples for growing l.

The multi-resolution scheme also includes a boosting formulation of KRR, where the estimator at level l is defined as

$$\widehat{f}_{n,\lambda}^{(l)}(x) = \widehat{f}_{n,\lambda}^{(l-1)}(x) + \widehat{s}_{n,\lambda}^{(l)}(x).$$

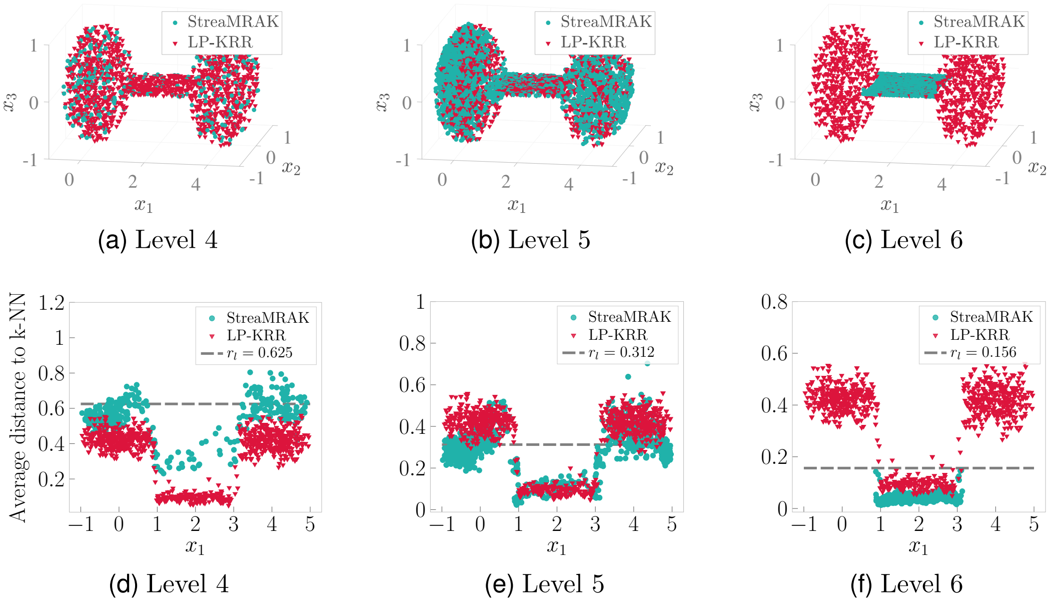
and $\widehat{s}_{n,\lambda}^{(l)}$ is the estimator obtained after regression on the residual $d^{(l)}([x_n]) = y - \widehat{f}_{n,\lambda}^{(l-1)}([x_n])$, where $d^{(0)} = y$. FALKON is employed as a base solver to solve the KRR at each level. Here we take $f([x_n])$ to mean $f([x_n]) = (f(x_1), f(x_2), \dots, f(x_n))^\top$.

This procedure corresponds to gradient boosting with L_2 loss from Section 1.5, where the step length is $\beta^{(1)} = 1$ at each step. Specific to StreaMRAK is that the samples $[x_n]$ used to calculate the residual at each level, are sampled independently from the samples used to train the model at the previous level.

We note that the multi-resolution scheme developed in StreaMRAK is inspired by a specific multi-resolution scheme used in image analysis, known as the Laplacian pyramid (LP) [23]. Moreover, during a further literature review, after finalizing the paper, we were able to establish a connection between the LP and a particular version of gradient boosting. In fact, Shao et al. [66] proposed a boosted version of KRR similar to StreaMRAK. However, we note that the boosted KRR developed in Shao et al. [66] does not correspond to a multi-resolution scheme. This is because the bandwidth is kept fixed at each level and the sub-sampling density is not adapted to the bandwidth. Regardless, the boosting perspective provides a useful foundation for interpreting the performance of StreaMRAK, and is, therefore, the perspective we have chosen to take in this discussion.

Boosting achieves two things. First, it is known to generate a composite estimator with better generalization properties than its base learners, even when these are weak learners. This justifies the adaptive bandwidth-selection approach discussed earlier. Since weak learners are acceptable, kernels can be defined with a bandwidth that is adequate without too much effort dedicated to finding the optimal one. Secondly, since StreaMRAK uses new samples at each level to evaluate the residuals $d^{(l)}([x_n])$, the estimator at the next level sees the generalization error of the previous and can compensate for it.

Finally, learning in high dimensions is known to be infeasible due to the large number of samples required; see Section 2. This problem is especially problematic in a multi-resolution scheme since high resolution, i.e. small kernel bandwidth, requires a high density of samples. Furthermore, as discussed in Section 3, the



(a) Level 4

(b) Level 5

(c) Level 6

(d) Level 4

(e) Level 5

(f) Level 6

Figure 1.5: Demonstration of how StreaMRAK adapts the sub-sampling density to the kernel bandwidth. The upper row shows the distribution of sub-samples on the dumbbell. The lower row shows the average distance to the 7-nn samples. The sub-samples selected by StreaMRAK are marked in blue, while the samples selected by random Nyström sub-sampling are marked in red. The grey dotted line in the lower row is the bandwidth. (Here LP-KRR refers to sub-sampling using random Nyström sub-sampling)

intrinsic dimension of point clouds can change based on location and resolution. In particular, if the noise level is reached, the intrinsic dimension becomes that of the ambient space. Because of this, StreaMRAK implements a scheme to identify regions of high dimensionality. The strategy is then to dedicate fewer resources to these regions and focus instead on lower-dimensional regions where learning is feasible.

To better understand the adaptive bandwidth- and sub-sampling strategy, we refer to Figure 1.5, which illustrates the method on a dumbbell-shaped point cloud. In the dumbbell, the spheres are 5-dimensional, and the connecting plane is 2-dimensional. In the figure, the upper row shows how the sub-samples are distributed on the dumbbell for levels $l = \{4, 5, 6\}$. The blue marks refer to sub-sampling with StreaMRAK, while the red marks (labeled LP-KRR) refer to random Nyström sub-sampling. The lower row in the figure shows, for levels $l = \{4, 5, 6\}$, the average distance between each sub-sample and its 7-nearest neighbors in the set of sub-samples. These distances are compared with the kernel bandwidth r_l at the corresponding level, shown by the grey dotted line. At each level, the kernel bandwidth is reduced by $r_l = 2^{-l} r_0$ from some initial bandwidth $r_0 > 0$. It is desirable that the average distance between sub-samples is comparable in magnitude with the kernel bandwidth. It is apparent that the sub-sampling used in StreaMRAK, adapts better to the bandwidth. In addition, from Figure 1.5f, we can see how StreaMRAK gives up in high dimensions by no longer selecting samples from the spherical regions when the bandwidth becomes too small.

6.2 Paper II: Improving inversion of model parameters from action potential recordings with kernel methods

An important aspect of developing anti-arrhythmic cardiac drugs is the measurement of ionic membrane currents $p = (p_1, \ldots, p_d)$ in cardiomyocytes. These ionic currents are responsible for the electrical properties and dynamics of cardiomyocytes, which in turn are essential to the contractions generated by these cells; see Remark 9. Furthermore, most anti-arrhythmic drug agents interact with ionic channels in the cellular and sub-cellular membranes to modulate ionic currents. Consequently, the measurement of ionic membrane currents can be used to guide the development of drugs that target these channels and give valuable insights into heart disease and electrical properties of the heart.

Remark 9 *"Cardiomyocytes are the cells responsible for generating contractile force in the intact heart."[129]*

Direct measurements of ionic membrane currents require expensive equipment and specialized practitioners [26, 62]. Meanwhile, the dynamics of these currents are responsible for generating the cardiac transmembrane potential v, known as the action potential (AP). We denote this relationship as $v = f(p)$. The AP can be measured at significantly lower cost and expertise, using techniques such as live cell fluorescence microscopy and microelectrode arrays [28, 53, 77]. Furthermore, several mathematical models $\widetilde{f}$ are developed to approximate the function f [39, 42, 48, 83, 87, 89, 97, 98, 114, 117]. Because of this, AP measurements, together with AP models, are a promising gateway to efficiently quantifying ionic membrane currents.

Problem 10 *Given an experimentally measured AP trace $w_i = (w_{i1}, \ldots, w_{iT}) \in \mathcal{V}_T \subset \mathbb{R}^T$ where T is the number of recorded time steps. Characterize the corresponding ionic membrane currents $p = (p_1, \ldots, p_d) \in \mathcal{P} \subset \mathbb{R}^d$ with the help of an AP model $\widetilde{f} : \mathcal{P} \to \mathcal{V}_T$.*

Problem 10 is an inverse problem. Namely, given a set of observations we want to find the parameters that caused them. This problem is, therefore, often referred to as the problem of AP trace inversion. What makes Problem 10 challenging is that the relationship between ionic membrane currents and the cardiac action potential is highly non-linear and stochastic in nature [88]. Furthermore, the AP is determined by substantial amounts of distinct ionic currents, many of which are of interest to identify in clinical applications. The AP models, designed to capture these dynamics, inevitably consist of complex systems of equations, typically large systems of ODEs that are expensive to compute; see e.g. Qu et al. [88] for a review on AP models and their construction.

Moreover, in many AP models, several ionic currents that are of interest to identify suffer from sensitivity and identifiability issues [56]. Meaning that their effect on the AP is hard to detect or the effects from different currents cancel each other out in certain regions of the parameter domain. Consequently, when designing algorithms for AP trace inversion, these are challenges that need to be taken into account.

In the literature on AP trace inversion, Problem 10 is normally addressed by defining a loss function $L(\phi^{AP}(w), \phi^{AP}(v))$ over a set of AP features $\phi^{AP}(v) = (\phi_1^{AP}(v), \ldots, \phi_m^{AP}(v))$, $\phi_i : \mathcal{V}_T \to \mathbb{R}$ constructed on the AP traces. The strategy is to search in parameter space $\mathcal{P}$ for a parameter vector p whose corresponding AP trace $v = \widetilde{f}(p)$ minimize the loss function. To find the minimizer, the common strategy is to use gradient-free iterative optimization schemes such as Nealder-Mead and Particle swarm [27, 57, 68]. However, the challenge with these iterative optimization schemes is that they must solve the AP model at each iteration. Since existing AP models are large systems of ODEs that are expensive to solve, this makes iterative optimization schemes slow in the face of large datasets.

In Tveito et al. [118], this scalability issue is addressed by first sampling a large quantity of ionic current parameters $\{p_i\}_{i=1}^n$ from $\mathcal{P}$, either uniformly or from a grid. The system of ODEs is then solved on these parameters to generate a dataset $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$ consisting of AP traces and the corresponding current parameters. We refer to this dataset as a "pre-computed" dataset. For a given measured AP trace w, one can then search for the closest AP trace within $\mathcal{D}_n$, namely $v_{opt} = \operatorname{argmin}_{v \in \mathcal{D}_n} L(v, w)$, where L is some loss function. If n is small, v_{opt} can be found by brute force, computing the distance between all sample pairs. However, for sufficiently large n, this is not computationally viable. In Tveito et al. [118], an iterative scheme, searching in bounding boxes defined in $\mathcal{P}$, was used instead. Thereby reducing the number of samples to compare.

The advantage of using a pre-computed dataset is that it moves the computational expense to a pre-computation step, making the algorithm significantly faster in the prediction phase; where one wish to find the ionic membrane currents corresponding to AP traces w measured in the lab. However, this comes at the cost of introducing large memory requirements in storing the pre-computed data, as well as requiring advanced methods for reading and accessing the data.

Manuscript contribution In this paper, we propose solving Problem 10 by learning an estimator $\widehat{f}_n$ of the inverse map $\widetilde{f}^{-1}$ using a pre-computed dataset $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$. The benefit of learning a model instead of using iterative optimization is that once the model is trained, prediction can be performed without the ODE system or the extensive pre-computed dataset.

Furthermore, we propose to use a kernel function $k : \mathcal{V}_T \times \mathcal{V}_T \to \mathbb{R}$ to implicitly map the AP traces into a high dimensional feature space, namely an RKHS $\mathcal{H}_k$. The features $\{\phi_i(v)\}_{i=1}^\infty$ of the RKHS give a much richer representation of the AP traces than the AP features $\{\phi_i^{AP}(v)\}_{i=1}^m$. Here $\phi_i = k(\cdot, v_i)$. Moreover, efficient comparison of AP traces is made possible by the kernel trick, $k(v_i, v_j) = \langle \phi_i, \phi_j \rangle_k$, which circumvents the need to calculate the features explicitly.

To find the best estimator in the RKHS we use kernel regularized ridge regression as this gives rise to a convex optimization problem in $\mathcal{H}_k$, thereby avoiding the issue of local minima. In the manuscript, we compare the KRR solvers StreaMRAK and FALKON, where StreaMRAK is the algorithm developed in the first paper in this thesis; see Section 6.1.

For a measured AP trace w, the performance of StreaMRAK and FALKON is compared to finding the best fit in the dataset, namely $v_{opt} = \text{argmin}_{v_i \in \mathcal{D}_n} L(v_i, w)$. For this purpose, the L_2 loss directly in $\mathcal{V}_T \subset \mathbb{R}^T$ and the L_2 loss in the AP-feature space are used. Since we are interested in comparing accuracy, it is natural to compare with v_{opt} as this is the solution that is searched for iteratively in Tveito et al. [118].

The contribution of this manuscript is to demonstrate that kernel methods are a viable modeling strategy for the problem of estimating ionic current parameters from AP trace measurements. The manuscript demonstrates that the kernel methods StreaMRAK and FALKON have significantly higher accuracy and reliability than the optimization scheme used in Tveito et al. [118]. This is important as high accuracy and reliability in predictions are essential in drug development, where errors can have severe consequences. In particular, StreaMRAK is shown to outperform FALKON both in terms of accuracy and reliability across different regions of the parameter domain.

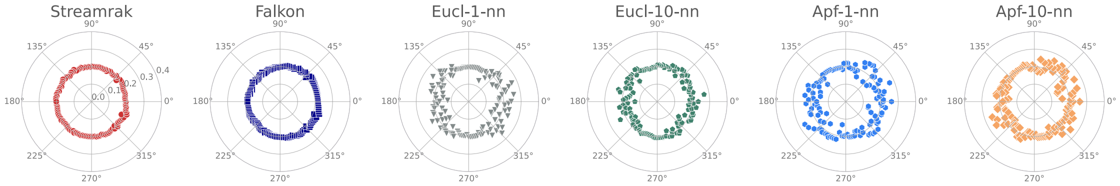


Figure 1.6: Predictions of parameters from AP traces corresponding to parameters on the circle $\mathcal{C} = \{p \in \mathcal{P} : \|p - p_0\|_2 = 0.2\}$, where $p_0 = (1, 1)$. For StreaMRAK and FALKON the predicted parameters are very close to the circle $\mathcal{C}$.

Figure 1.6 compares StreaMRAK and FALKON with four alternative parameter prediction schemes. Here Eucl-1-nn refers to $v_{opt} = \text{argmin}_{v_i \in \mathcal{D}_n} L(v_i, w)$ with the L_2 loss in $\mathcal{V}_T$ and Eucl-10-nn refers to the average over the 10 nearest neighbours as measured by this loss. Similarly, Apf-1-nn refers to v_{opt} with the L_2 loss in AP-feature space, and Apf-10-nn refers to the average over the 10 nearest neighbours. The algorithms are given AP traces corresponding to parameters sampled from the circle $\mathcal{C} = \{p \in \mathcal{P} : \|p - p_0\|_2 = 0.2\}$, where $p_0 = (1, 1)$. The goal is to predict the parameters that generated the given AP traces. From the figure, it is clear that the prediction accuracy of StreaMRAK and FALKON is higher and also more consistent in every direction in the parameter domain than the alternative schemes.

6.3 Paper III: Effective resistance in metric spaces

Effective resistance (ER) is a distance metric on graphs. An important application of this distance metric is to uncover the intrinsic structure of point clouds. However, ER suffers from a major limitation. Namely, as the graph size increases, the ER between nodes in a graph converges to a trivial limit. The latest demonstration of this problem is due to Von-Luxburg et al. [71, 125] following several other works on this issue [7, 20, 70]. This problem is commonly referred to as the Von-Luxburg limit, which we define in Proposition 11.

Proposition 11 (Von-Luxburg limit [71]) *Let $G_n = (X_n, W)$ be a graph, with nodes $X_n = \{x_1, \dots, x_n\}$, edge weights W_{ij} and let $D_i = \sum_{j=1}^n W_{ij}$ be the degree of node x_i. Let $R_n(x_i, x_j)$ denote the effective resistance between node $x_i, x_j \in X_n$ defined in Proposition 8. It then follows that*

$$\lim_{n\to\infty} R_n(x_i, x_j) \propto 1/D_i + 1/D_j$$

The consequence of Proposition 11 is that in the asymptotic limit, the distance between two graph nodes $x_i, x_j \in X_n$ is only determined by their respective degrees D_i, D_j. Consequently, the ER is effectively meaningless as a distance metric. Furthermore, Von-Luxburg et al. [125] show that this problem occurs already for relatively small graphs with $n \approx 1000$ nodes.

Manuscript contribution The contribution of this manuscript is to introduce the concept of region-based ER and to demonstrate that this definition does not suffer from the trivial limit described in Proposition 11. Let $G_n = (X_n, W)$ be a graph with nodes $X_n = \{x_1, \dots, x_n\}$ and let $X_s, X_g \subset X_n$ be two non-empty disjoint subsets. We define the region-based ER as $R_n(X_s, X_g) = 1/J_{tot}$ where

$$J_{tot} = \sum_{x_i \in X_s} \sum_{x_j \in X_n} W_{ij}(v_n^*(x_i) - v_n^*(x_j))$$

is the total current between X_s and X_g induced by the energy-minimizing voltage v_n^*. This voltage is defined as the solution to the energy minimization problem

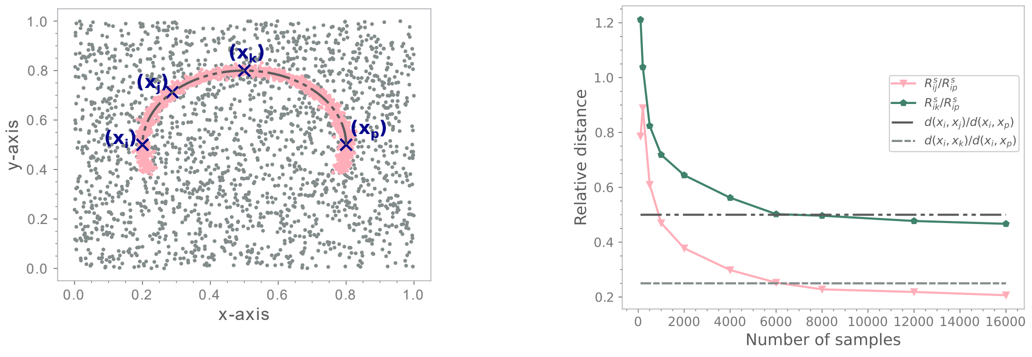
$$\begin{aligned} \min_{v : X_n \to \mathbb{R}} \quad & \sum_{x_i, x_j \in X_n} W_{i,j}(v(x_i) - v(x_j))^2 \\ \text{Subject to} \quad & v(x_i) = 1 \, \forall x_i \in X_s, \quad v(x_i) = 0, \, \forall x_i \in X_g. \end{aligned}$$

The region-based ER can be contrasted with the classical definition of ER from Proposition 8.

The region-based ER is based on the definition of ER between sets from Song et al. [110]. In the manuscript, we extend this to the setting where X_n are sampled from a distribution μ defined over some metric space (M, d). We let $X_s = \{x \in X_n : x \in M_s\}$ and $X_g = \{x \in X_n : x \in M_g\}$ where $M_s, M_g \subset M$ are disjoint measurable subsets of M.

Using a kernel function $k : M \times M \to \mathbb{R}$, combined with appropriate scaling of the edge weights, a local neighborhood graph G_n is constructed as described in Definition 6. Under certain technical conditions on M, k, and μ, the region-based ER defined on G_n is shown to converge in probability to a limit object $R_\mu(M_s, M_g)$ as $n \to \infty$. The contributions of the manuscript can be summarized as:

1. *Existence*: We prove the existence and uniqueness of $R_\mu(M_s, M_g)$
2. *Convergence*: We prove that $R_n(X_s, X_g)$ converges to $R_\mu(M_s, M_g)$ in probability as $n \to \infty$



(a)

(b)

Figure 1.7: Convergence of region-based ER to a meaningful limit. (a) The pink half-moon is a high-density region over a low-density background. (b) the horizontal lines corresponds to Γ_{ijp} and Γ_{ikp}. Pink line shows R^s_{ij}/R^s_{ip} while green line shows R^s_{ik}/R^s_{ip}.

3. *Distance metric*: We prove that region-based ER is a distance metric

4. *Meaningful*: We demonstrate numerically that region-based ER converges to a meaningful limit

For a numerical example, consider the distance along the arch of the half-moon illustrated in Figure 1.7a. Let $d(\cdot, \cdot)$ denote the true distance along the arch and define the ratios

$$\Gamma_{ijp} := d(x_i, x_j)/d(x_i, x_p) = 0.25 \quad \text{and} \quad \Gamma_{ikp} := d(x_i, x_k)/d(x_i, x_p) = 0.5 \tag{1.18}$$

Let $R_{ij} := R_n(X_i, X_j)$. As seen from Figure 1.7, when n increases, the ratios of the region-based ER, namely R_{ij}/R_{ip} and R_{ik}/R_{ip}, converges to values close to the ratios Γ_{ijp} and Γ_{ikp} respectively. Note that since the ER incorporates all possible paths between the two nodes, we do not expect the region-based ER to converge exactly to Γ_{ijp} and Γ_{ikp}. However, since the density of the half-moon is significantly higher than that of the background, we expect the limits to be close. On the other hand, with the limit in Proposition 11, the ratios are 1. This is because the density on the half-moon is uniform, which means the respective degrees of the nodes are the same.

6.4 Paper IV: Structure from voltage

Non-linear dimensionality reduction (NLDR) is the discipline of finding lower-dimensional representations of non-linear data to reduce the impact of the curse of dimensionality. As datasets are rapidly growing in size, developing scalable NLDR algorithms is becoming increasingly important.

A powerful strategy to achieve scalability is to utilize powerful computational models such as parallelization, distribution, and streaming. However, existing NLDR techniques based on eigenfunction calculations, such as Laplacian eigenmaps [14], are generally incompatible with these computational models. From the discussion in Section 4.2, we know that part of the problem with LE is that it does not provide guarantees for the functions to be localized. Whereby localized, we mean in the sense defined in Definition 7. Furthermore, demanding the eigenfunctions to be orthogonal means they can not be computed independently. Moreover, once the eigenfunctions are calculated, they can not be extended to new samples without repeating the process.

Manuscript contributions In this manuscript, we propose a novel embedding scheme based on localized voltage functions that can be calculated independently, thereby allowing them to be computed using parallelization and distributed schemes. The voltage functions we define can also be easily extended to new samples, which makes them compatible with a streaming model of computation. We refer to these voltage functions as *grounded metric voltage* functions (GMVs) denoted v_{n,s_i}. The proposed embedding is

$$x_i \mapsto (v_{n,s_1}(x_i), \dots v_{n,s_m}(x_i))^\top. \tag{1.19}$$

Consider a setting where $X_n = \{x_1, \dots, x_n\}$ is sampled from a distribution μ over some metric space (M, d). Let $k : M \times M \to [0, 1]$ be a kernel function and let (X_n, W) be a graph with edge weights $W_{ij} = k(x_i, x_j)/n^2$. Furthermore, let r_s be some radius and $g : \mathbb{R} \to [0, r]$ with $r > 0$ be a monotonic strictly decreasing function. Define $X_s = \{x \in X_n : x \in M_s\}$, where $M_s = \{x \in M : g(d(x, x_s)) \leq r_s\}$. Here $x_s \in M$ is what we call a source center. For a given source x_s, we define the associated GMV function $v_{n,s} : X_n \to [0, 1]$ as the solution to the energy minimization problem

$$\min_{v:X\to[0,1]} \sum_{x_i,x_j\in X_n} W_{ij}(v(x_i) - v(x_j))^2 + \sum_{x_i\in X_n} \rho v^2(x_i)$$
$$\text{Subject to} \quad v(x_i) = 1 \quad \text{for all} \quad x_i \in X_s.$$

The term $\sum_{x_i} \rho v^2(x_i)$ incorporates the effect of an universal ground $x_g \notin X_n$, with voltage $v(x_g) = 0$, that connects to all nodes in X_n with edge weight $\rho = \rho_g/n$ for $\rho_g > 0$. This can easily be seen by considering $X_n \cup \{x_g\}$ and adding an extra row and column to W_{ij} with the weight ρ. The term $\sum_{x_i\in X_n\cup\{x_g\}} \rho(v(x_i) - v(x_g))^2 = \sum_{x_i\in X_n\cup\{x_g\}} \rho v^2(x_i)$ can then be extracted from the sum. Since the voltage of the ground is anyway $v(x_g) = 0$, we drop the sum over x_g and ignore the ground in constructing W_{ij}.

We note that in a random walk perspective, the ground can be interpreted as a trap node with zero escape probability.

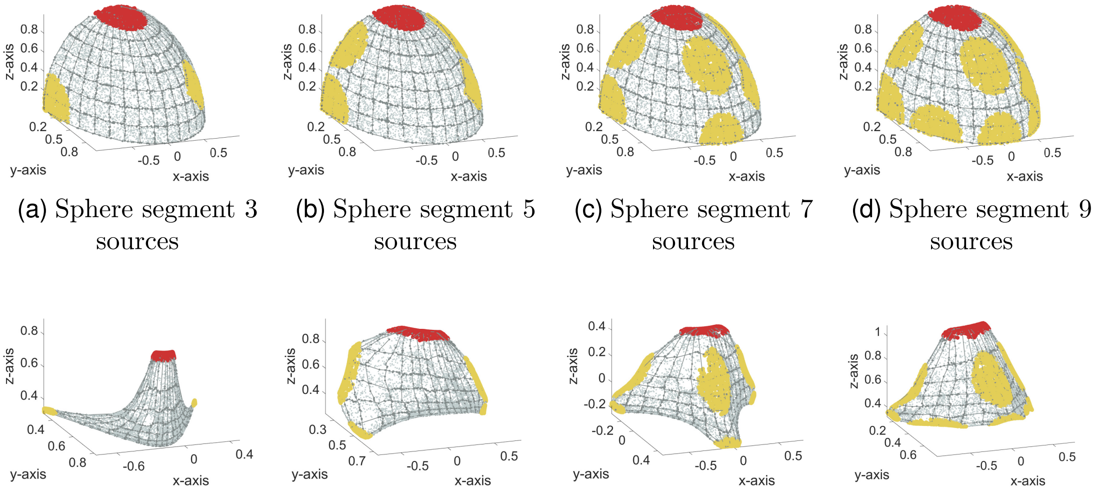
The idea of the source and ground constraints is that together, they create a voltage function localized around x_s. The contributions of the manuscript can be summarized as follows:

1. *Existence*: We prove the existence and uniqueness of a limit object v_s^*.

2. *Convergence*: We prove that $v_{n,s}$ converges to v_s^* in probability as $n \to \infty$.

3. *Locality*: We provide bounds on the shape of v_s^* on the unit sphere S^{d-1}, proving that $v_s^*(x_i)$ decays exponentially with increasing $d_S(x_i, x_s)$, with decay governed by the magnitude of ρ. Here d_S is the geodesic on S^{d-1}.

4. *Embedding*: We show analytically and numerically how the GMV can provide an embedding of the unit sphere.



(a) Sphere segment 3 sources (b) Sphere segment 5 sources (c) Sphere segment 7 sources (d) Sphere segment 9 sources

(e) Embedding 3 sources (f) Embedding 5 sources (g) Embedding 7 sources (h) Embedding 9 sources

Figure 1.8: Embedding of the two first quadrants of the unit sphere. The yellow and red circles are source regions. The upper row shows the two first quadrants of the unit sphere, and the lower row is the embedding $x_i \mapsto (v_{n,s_1}(x_i), \dots, v_{n,s_m}(x_i))$ for $m \in \{3, 5, 7, 9\}$.

It can be shown that v_{n,s_i} satisfies

$$v_{n,s_i} = \widetilde{D}^{-1}\widetilde{W}^{(s_i)} v_{n,s_i}$$

where $\widetilde{W}^{(s_i)}$ is referred to as the grounded weight matrix. This means that v_{n,s_i} can be found by power iteration where $\widetilde{D}^{-1}\widetilde{W}^{(s_i)}$ is applied iteratively until convergence. Due to the locality of v_{n,s_i}, it follows that only a sub-matrix is necessary, which greatly reduces the computational expense of the iterative procedure.

In Figure 1.8, an embedding $x_i \mapsto (v_{n,s_1}(x_i), \dots, v_{n,s_m}(x_i))$ of the 2-dimensional unit sphere embedded in $\mathbb{R}^D$ is demonstrated for $m \in \{3, 5, 7, 9\}$. Here $m \ll D$. For visualization, multidimensional scaling [35] is used to project the representation into $\mathbb{R}^3$.

7 Summary and outlook

Learning algorithms based on a kernel function are theoretically well understood in statistical learning theory and machine learning, which makes them attractive learning algorithms in terms of reliability and interpretability. However, in their basic form, they suffer from large memory requirements and computational costs, preventing their utility in real-world applications where datasets are huge. Therefore, to allow the extension of these algorithms to big data, the development of scalable kernel-based learning schemes is of interest.

In this thesis, we made several contributions that improve the scalability of kernel-based learning. The novel KRR solver StreaMRAK developed in paper I and further demonstrated in paper II, has shown promise as an efficient KRR solver with improvements over the existing KRR solver FALKON.

Future work should focus on studying the generalization properties of StreaMRAK. In particular, in the asymptotic limit, FALKON has estimation rates that are optimal in a min-max sense [95]. As such StreaMRAK can not achieve better in this limit. However, our numerical experiments suggest that StreaMRAK can reach high predictive accuracy with significantly fewer samples than FALKON. The characterization of the generalization properties of StreaMRAK is therefore of great interest to gain insights on how the boosting and adaptive sub-sampling impact the learning.

In the second paper of this thesis, StreaMRAK was demonstrated as a reliable algorithm for estimating ionic membrane currents from cardiac AP traces. This is an important problem within cardiac anti-arrhythmic research and cardiac drug development. Therefore, a further demonstration of StreaMRAK in this field is of great interest. In particular, the important aspects of AP trace inversion are scalability, reliability, and the ability to handle parameters with identifiability and sensitivity issues. The study in paper II focuses on demonstrating StreaMRAK in terms of its reliability and its ability to detect low-sensitivity parameters. The study is performed in a controlled setting with a limited number of parameters. Future work should focus on identifiability issues and, finally, on applying StreaMRAK to modern AP models with hundreds of parameters.

The third paper in this thesis has demonstrated how the region-based ER avoids the convergence issues of standard ER. Future work should focus on characterizing the dependency between source radius, kernel bandwidth, the weight-to-ground, and graph size. The consequences of decreasing the source radius are of particular interest.

The theoretical and numerical results from paper IV lay the foundations for an embedding scheme utilizing grounded metric voltage functions. For future work, there are in particular two directions of interest. We are currently working towards extending the embedding scheme to more general manifolds and real-world data. Furthermore, the computational and algorithmic aspects of the embedding scheme require further development. In particular, our results show that the GMVs are localized and can be computed independently in an iterative manner, indicating compatibility with distribution and streaming. Current efforts are therefore dedicated to algorithmic developments that can utilize these properties.

Bibliography

[1] Ittai Abraham, Yair Bartal and Ofer Neimany. 'Advances in metric embedding theory'. In: *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. 2006, pp. 271–286. DOI: 10.1145/1132516.1132557.

[2] MA Aiserman, Emmanuil M Braverman and Lev I Rozonoer. 'Theoretical foundations of the potential function method in pattern recognition'. In: *Avtomat. i Telemeh.* 25.6 (1964), pp. 917–936.

[3] Ahmed Alaoui and Michael W Mahoney. 'Fast randomized kernel ridge regression with statistical guarantees'. In: *Advances in neural information processing systems* 28 (2015). URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/f3f27a324736617f20abbf2ffd806f6d-Paper.pdf.

[4] William K Allard, Guangliang Chen and Mauro Maggioni. 'Multi-scale geometric methods for data sets II: Geometric multi-resolution analysis'. In: *Applied and computational harmonic analysis* 32.3 (2012), pp. 435–462. DOI: 10.1016/j.acha.2011.08.001.

[5] Elena A Allen, Erik B Erhardt and Vince D Calhoun. 'Data visualization in the neurosciences: overcoming the curse of dimensionality'. In: *Neuron* 74.4 (2012), pp. 603–608. DOI: 10.1016/j.neuron.2012.05.001.

[6] Sylvain Arlot and Alain Celisse. 'A survey of cross-validation procedures for model selection'. In: *Statistics Surveys* 4 (2010), pp. 40–79. DOI: 10.1214/09-SS054.

[7] Chen Avin and Gunes Ercal. 'On the cover time and mixing time of random geometric graphs'. In: *Theoretical Computer Science* 380.1-2 (2007), pp. 2–22. DOI: 10.1016/j.tcs.2007.02.065.

[8] Ariful Azad and Aydin Buluç. 'A Work-Efficient Parallel Sparse Matrix-Sparse Vector Multiplication Algorithm'. In: *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 2017, pp. 688–697. DOI: 10.1109/IPDPS.2017.76.

[9] Francis Bach. 'Sharp analysis of low-rank kernel matrix approximations'. In: *Proceedings of the 26th Annual Conference on Learning Theory*. Vol. 30. Proceedings of Machine Learning Research. Princeton, NJ, USA, 2013, pp. 185–209. URL: https://proceedings.mlr.press/v30/Bach13.html.

[10] Maroua Bahri et al. 'Data stream analysis: Foundations, major tasks and tools'. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11.3 (2021), e1405. DOI: 10.1002/widm.1405.

[11] Ron Bekkerman, Mikhail Bilenko and John Langford. *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011.

[12] Mikhail Belkin. 'Problems of learning on manifolds'. PhD thesis. The University of Chicago, 2004.

[13] Mikhail Belkin and Partha Niyogi. 'Convergence of Laplacian Eigenmaps'. In: *Advances in Neural Information Processing Systems*. Vol. 19. 2006. URL: https://proceedings.neurips.cc/paper_files/paper/2006/file/5848ad959570f87753a60ce8be1567f3-Paper.pdf.

[14] Mikhail Belkin and Partha Niyogi. 'Laplacian Eigenmaps for Dimensionality Reduction and Data Representation'. In: *Neural Computation* 15.6 (2003), pp. 1373–1396. DOI: 10.1162/089976603321780317.

[15] Mikhail Belkin and Partha Niyogi. 'Semi-supervised learning on manifolds'. In: *Machine Learning Journal* 1 (2002).

[16] Mikhail Belkin and Partha Niyogi. 'Towards a theoretical foundation for Laplacian-based manifold methods'. In: *Journal of Computer and System Sciences* 74.8 (2008), pp. 1289–1308. DOI: 10.1016/j.jcss.2007.08.006.

[17] R. Bellman and R. Kalaba. 'On adaptive control processes'. In: *IRE Transactions on Automatic Control* 4.2 (1959), pp. 1–9. DOI: 10.1109/TAC.1959.1104847.

[18] Visar Berisha et al. 'Digital medicine and the curse of dimensionality'. In: *NPJ digital medicine* 4.1 (2021), p. 153. URL: https://doi.org/10.1038/s41746-021-00521-5.

[19] 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*. 1992, pp. 144–152. DOI: 10.1145/130385.130401.

[20] Stephen P Boyd et al. 'Mixing Times for Random Walks on Geometric Random Graphs.' In: *ALENEX/ANALCO*. 2005, pp. 240–249.

[21] Sergey Brin and Lawrence Page. 'The anatomy of a large-scale hypertextual Web search engine'. In: *Computer Networks and ISDN Systems* 30.1 (1998), pp. 107–117. DOI: 10.1016/S0169-7552(98)00110-X.

[22] Fred Buckley and Frank Harary. *Distance in graphs*. Vol. 2. Addison-Wesley Redwood City, 1990.

[23] Peter J. Burt and Edward H. Adelson. 'The Laplacian Pyramid as a Compact Image Code'. In: *Readings in Computer Vision*. Morgan Kaufmann, 1987, pp. 671–679. DOI: 10.1016/B978-0-08-051581-6.50065-9.

[24] Raffaello Camoriano et al. 'NYTRO: When Subsampling Meets Early Stopping'. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Vol. 51. Proceedings of Machine Learning Research. 2016, pp. 1403–1411. URL: https://proceedings.mlr.press/v51/camoriano16.html.

[25] Gavin C. Cawley and Nicola L.C. Talbot. 'Fast exact leave-one-out cross-validation of sparse least-squares support vector machines'. In: *Neural Networks* 17.10 (2004), pp. 1467–1475. DOI: 10.1016/j.neunet.2004.07.002.

[26] Chris Chambers et al. 'High-throughput screening of Na$_V$1. 7 modulators using a giga-seal automated patch clamp instrument'. In: *Assay and drug development technologies* 14.2 (2016), pp. 93–108. DOI: 10.1089/adt.2016.700.

[27] Fulong Chen et al. 'Identification of the Parameters of the Beeler–Reuter Ionic Equation With a Partially Perturbed Particle Swarm Optimization'. In: *IEEE Transactions on Biomedical Engineering* 59.12 (2012), pp. 3412–3421. DOI: 10.1109/TBME.2012.2216265.

[28] Mike Clements and Nick Thomas. 'High-Throughput Multi-Parameter Profiling of Electrophysiological Drug Effects in Human Embryonic Stem Cell Derived Cardiomyocytes Using Multi-Electrode Arrays'. In: *Toxicological Sciences* 140.2 (2014), pp. 445–461. DOI: 10.1093/toxsci/kfu084.

[29] Michael B Cohen et al. 'Uniform sampling for matrix approximation'. In: *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*. 2015, pp. 181–190. DOI: 10.1145/2688073.2688113.

[30] Ronald R Coifman and Stéphane Lafon. 'Diffusion maps'. In: *Applied and computational harmonic analysis* 21.1 (2006), pp. 5–30. DOI: 10.1016/j.acha.2006.04.006.

[31] Ronald R Coifman et al. 'Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps'. In: *Proceedings of the national academy of sciences* 102.21 (2005), pp. 7426–7431. DOI: 10.1073/pnas.0500334102.

[32] Corinna Cortes and Vladimir Vapnik. 'Support-vector networks'. In: *Machine learning* 20 (1995), pp. 273–297. DOI: 10.1007/BF00994018.

[33] Jose A Costa and Alfred O Hero. 'Learning intrinsic dimension and intrinsic entropy of high-dimensional datasets'. In: *2004 12th European Signal Processing Conference*. IEEE. 2004, pp. 369–372.

[34] Felipe Cucker and Steve Smale. 'On the mathematical foundations of learning'. In: *Bulletin of the American mathematical society* 39.1 (2002), pp. 1–49.

[35] Ivan Dokmanic et al. 'Euclidean Distance Matrices: Essential theory, algorithms, and applications'. In: *IEEE Signal Processing Magazine* 32.6 (2015), pp. 12–30. DOI: 10.1109/MSP.2015.2398954.

[36] Petros Drineas, Ravi Kannan and Michael W Mahoney. ‘Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication’. In: *SIAM Journal on Computing* 36.1 (2006), pp. 132–157. DOI: 10.1137/S0097539704442684.

[37] Petros Drineas, Ravi Kannan and Michael W Mahoney. ‘Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix’. In: *SIAM Journal on computing* 36.1 (2006), pp. 158–183. DOI: 10.1137/S0097539704442696.

[38] Petros Drineas et al. ‘Fast approximation of matrix coherence and statistical leverage’. In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 3475–3506.

[39] Andrew G Edwards and William E Louch. ‘Species-dependent mechanisms of cardiac arrhythmia: a cellular focus’. In: *Clinical Medicine Insights: Cardiology* 11 (2017), pp. 1179–5468. DOI: 10.1177/1179546816686061.

[40] Artur J Ferreira and Mário AT Figueiredo. ‘Boosting algorithms: A review of methods, theory, and applications’. In: *Ensemble machine learning: Methods and applications* (2012), pp. 35–85.

[41] Shai Fine and Katya Scheinberg. ‘Efficient SVM training using low-rank kernel representations’. In: *Journal of Machine Learning Research* 2.Dec (2001), pp. 243–264.

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

[43] Yoav Freund. ‘Boosting a weak learning algorithm by majority’. In: *Information and computation* 121.2 (1995), pp. 256–285. DOI: 10.1006/inco.1995.1136.

[44] Yoav Freund and Robert E Schapire. ‘A decision-theoretic generalization of on-line learning and an application to boosting’. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139. DOI: 10.1006/jcss.1997.1504.

[45] Jerome H. Friedman. ‘Greedy function approximation: A gradient boosting machine.’ In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451.

[46] Jerome H. Friedman. ‘Stochastic gradient boosting’. In: *Computational Statistics & Data Analysis* 38.4 (2002), pp. 367–378. DOI: 10.1016/S0167-9473(01)00065-2.

[47] Heitor Murilo Gomes et al. ‘Machine learning for streaming data: state of the art, challenges, and opportunities’. In: *ACM SIGKDD Explorations Newsletter* 21.2 (2019), pp. 6–22. DOI: 10.1145/3373464.3373470.

[48] Eleonora Grandi, Francesco S. Pasqualini and Donald M. Bers. ‘A novel computational model of the human ventricular action potential and Ca transient’. In: *Journal of Molecular and Cellular Cardiology* 48.1 (2010), pp. 112–121. DOI: 10.1016/j.yjmcc.2009.09.019.

[49] László Györfi et al. *A distribution-free theory of nonparametric regression.* Vol. 1. Springer, 2002.

[50] Jihun Ham et al. 'A kernel view of the dimensionality reduction of manifolds'. In: *Proceedings of the twenty-first international conference on Machine learning.* 2004, p. 47. DOI: 10.1145/1015330.1015417.

[51] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction.* Vol. 2. Springer, 2009.

[52] Matthias Hein, Jean-Yves Audibert and Ulrike von Luxburg. 'Graph Laplacians and their convergence on random neighborhood graphs.' In: *Journal of Machine Learning Research* 8.6 (2007).

[53] Todd J Herron, Peter Lee and José Jalife. 'Optical imaging of voltage and calcium in cardiac cells & tissues'. In: *Circulation research* 110.4 (2012), pp. 609–623. DOI: 10.1161/CIRCRESAHA.111.247494.

[54] Thomas Hofmann, Bernhard Schölkopf and Alexander J. Smola. 'Kernel methods in machine learning'. In: *The Annals of Statistics* 36.3 (2008), pp. 1171–1220. DOI: 10.1214/009053607000000677.

[55] Alan Julian Izenman. 'Introduction to manifold learning'. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4.5 (2012), pp. 439–446. DOI: 10.1002/wics.1222.

[56] Karoline Horgmo Jæger et al. 'Identifying Drug Response by Combining Measurements of the Membrane Potential, the Cytosolic Calcium Concentration, and the Extracellular Potential in Microphysiological Systems'. In: *Frontiers in Pharmacology* 11 (2021), pp. 569–489. DOI: https://doi.org/10.3389/fphar.2020.569489.

[57] Karoline Horgmo Jæger et al. 'Improved Computational Identification of Drug Response Using Optical Measurements of Human Stem Cell Derived Cardiomyocytes in Microphysiological Systems'. In: *Frontiers in Pharmacology* 10 (2020). DOI: 10.3389/fphar.2019.01648.

[58] Palle ET Jorgensen and PJ Pearse Erin. 'Operator theory and analysis of infinite networks'. In: *arXiv preprint arXiv:0806.3881* 3 (2008).

[59] George Kimeldorf and Grace Wahba. 'Some results on Tchebycheffian spline functions'. In: *Journal of mathematical analysis and applications* 33.1 (1971), pp. 82–95.

[60] George S. Kimeldorf and Grace Wahba. 'A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines'. In: *Ann. Math. Stat.* 41.2 (2011), pp. 495–502. DOI: 10.1214/aoms/1177697089.

[61] Douglas J Klein and Milan Randić. 'Resistance distance'. In: *Journal of mathematical chemistry* 12.1 (1993), pp. 81–95. DOI: 10.1007/BF01164627.

[62] Bruce G. Kornreich. 'The patch clamp technique: Principles and technical considerations'. In: *Journal of Veterinary Cardiology* 9.1 (2007), pp. 25–37. DOI: 10.1016/j.jvc.2007.02.001.

[63] John A Lee, Michel Verleysen et al. *Nonlinear dimensionality reduction.* Vol. 1. Springer, 2007.

[64] Wenjing Liao and Mauro Maggioni. 'Adaptive Geometric Multiscale Approximations for Intrinsically Low-dimensional Data.' In: *Journal of Machine Learning Research* 20 (2019), pp. 98–1.

[65] Wenjing Liao, Mauro Maggioni and Stefano Vigogna. 'Learning adaptive multiscale approximations to data and functions near low-dimensional sets'. In: *IEEE Information Theory Workshop.* IEEE. 2016, pp. 226–230. DOI: 10.1109/ITW.2016.7606829.

[66] Shao-Bo Lin, Yunwen Lei and Ding-Xuan Zhou. 'Boosted Kernel Ridge Regression: Optimal Learning Rates and Early Stopping'. In: *Journal of Machine Learning Research* 20.46 (2019), pp. 1–36. URL: http://jmlr.org/papers/v20/18-063.html.

[67] Anna V Little, Mauro Maggioni and Lorenzo Rosasco. 'Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature'. In: *Applied and Computational Harmonic Analysis* 43.3 (2017), pp. 504–567. DOI: 10.1016/j.acha.2015.09.009.

[68] Axel Loewe et al. 'Parameter Estimation of Ion Current Formulations Requires Hybrid Optimization Approach to Be Both Accurate and Reliable'. In: *Frontiers in Bioengineering and Biotechnology* 3 (2016). DOI: 10.3389/fbioe.2015.00209.

[69] Wei-Yin Loh. 'Classification and regression trees'. In: *Wiley interdisciplinary reviews: data mining and knowledge discovery* 1.1 (2011), pp. 14–23. DOI: https://doi.org/10.1002/widm.8.

[70] László Lovász. 'Random walks on graphs: A survey'. In: *Combinatorics* 2 (1993), pp. 1–46.

[71] Ulrike Luxburg, Agnes Radl and Matthias Hein. 'Getting lost in space: Large sample analysis of the resistance distance'. In: *Advances in Neural Information Processing Systems.* Vol. 23. Curran Associates, Inc., 2010. URL: https://proceedings.neurips.cc/paper/2010/file/0d0871f0806eae32d30983b62252da50-Paper.pdf.

[72] Mauro Maggioni, Stanislav Minsker and Nate Strawn. 'Multiscale Dictionary Learning: Non-Asymptotic Bounds and Robustness'. In: *Journal of Machine Learning Research* 17.2 (2016), pp. 1–51. URL: http://jmlr.org/papers/v17/maggioni16a.html.

[73] Michael W Mahoney and Petros Drineas. 'CUR matrix decompositions for improved data analysis'. In: *Proceedings of the National Academy of Sciences* 106.3 (2009), pp. 697–702.

[74] Michael W. Mahoney. 'Randomized Algorithms for Matrices and Data'. In: *Foundations and Trends in Machine Learning* 3.2 (2011), pp. 123–224. DOI: 10.1561/2200000035.

[75] Abdul Majeed and Ibtisam Rauf. 'Graph theory: A comprehensive survey about graph theory applications in computer science and social networks'. In: *Inventions* 5.1 (2020), p. 10. DOI: 10.3390/inventions5010010.

[76] Per-Gunnar Martinsson and Joel A Tropp. 'Randomized numerical linear algebra: Foundations and algorithms'. In: *Acta Numerica* 29 (2020), pp. 403–572. DOI: 10.1017/S0962492920000021.

[77] Anurag Mathur et al. 'Human iPSC-based Cardiac Microphysiological System For Drug Screening Applications'. In: *Scientific Reports* 5.1 (Mar. 2015). DOI: https://doi.org/10.1038/srep08883.

[78] Giacomo Meanti et al. 'Kernel methods through the roof: handling billions of points efficiently'. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14410–14422. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/a59afb1b7d82ec353921a55c579ee26d-Paper.pdf.

[79] Charles A. Micchelli, Yuesheng Xu and Haizhang Zhang. 'Universal Kernels'. In: *Journal of Machine Learning Research* 7.95 (2006), pp. 2651–2667. URL: http://jmlr.org/papers/v7/micchelli06a.html.

[80] Berndt Müller, Joachim Reinhardt and Michael T Strickland. *Neural networks: an introduction.* Springer Science & Business Media, 1995.

[81] Shanmugavelayutham Muthukrishnan et al. 'Data streams: Algorithms and applications'. In: *Foundations and Trends in Theoretical Computer Science* 1.2 (2005), pp. 117–236. DOI: http://dx.doi.org/10.1561/0400000002.

[82] Erich Novak and Hans Triebel. 'Function Spaces in Lipschitz Domains and Optimal Rates of Convergence for Sampling.' In: *Constructive approximation* 23.3 (2006).

[83] Thomas O'Hara et al. 'Simulation of the Undiseased Human Cardiac Ventricular Action Potential: Model Formulation and Experimental Validation'. In: *PLOS Computational Biology* 7.5 (May 2011), pp. 1–29. DOI: https://doi.org/10.1371/journal.pcbi.1002061.

[84] Phillip A Ostrand. 'Covering dimension in general spaces'. In: *General Topology and its applications* 1.3 (1971), pp. 209–221. DOI: https://doi.org/10.1016/0016-660X(71)90093-6.

[85] Lawrence Page et al. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford infolab, 1999.

[86] Georgios A Pavlopoulos et al. 'Using graph theory to analyze biological networks'. In: *BioData mining* 4 (2011), pp. 1–27. DOI: https://doi.org/10.1186/1756-0381-4-10.

[87] Zhilin Qu et al. 'Nonlinear and stochastic dynamics in the heart'. In: *Physics Reports* 543.2 (2014), pp. 61–162. DOI: https://doi.org/10.1016/j.physrep.2014.05.002.

[88] Zhilin Qu et al. 'Nonlinear and stochastic dynamics in the heart'. In: *Physics Reports* 543.2 (2014), pp. 61–162. DOI: https://doi.org/10.1016/j.physrep.2014.05.002.

[89] Alfio Quarteroni et al. 'Integrated Heart—Coupling multiscale and multiphysics models for the simulation of the cardiac function'. In: *Computer Methods in Applied Mechanics and Engineering* 314 (2017), pp. 345–407. DOI: https://doi.org/10.1016/j.cma.2016.05.031.

[90] Ali Rahimi and Benjamin Recht. 'Random features for large-scale kernel machines'. In: *Advances in neural information processing systems* 20 (2007). URL: https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

[91] Ferozuddin Riaz and Khidir M Ali. 'Applications of graph theory in computer science'. In: *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE. 2011, pp. 142–145. DOI: 10.1109/CICSyN.2011.40.

[92] Lorenzo Rosasco, Mikhail Belkin and Ernesto De Vito. 'On Learning with Integral Operators'. In: *Journal of Machine Learning Research* 11.30 (2010), pp. 905–934. URL: http://jmlr.org/papers/v11/rosasco10a.html.

[93] Sam T. Roweis and Lawrence K. Saul. 'Nonlinear Dimensionality Reduction by Locally Linear Embedding'. In: *Science* 290.5500 (2000), pp. 2323–2326. DOI: 10.1126/science.290.5500.2323.

[94] Alessandro Rudi, Raffaello Camoriano and Lorenzo Rosasco. 'Less is more: Nyström computational regularization'. In: *Advances in Neural Information Processing Systems* 28 (2015). URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/03e0704b5690a2dee1861dc3ad3316c9-Paper.pdf.

[95] Alessandro Rudi, Luigi Carratino and Lorenzo Rosasco. 'Falkon: An optimal large scale kernel method'. In: *Advances in neural information processing systems* 30 (2017). URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/05546b0e38ab9175cd905eebcc6ebb76-Paper.pdf.

[96] Alessandro Rudi et al. 'On fast leverage score sampling and optimal learning'. In: *Advances in Neural Information Processing Systems* 31 (2018). URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/56584778d5a8ab88d6393cc4cd11e090-Paper.pdf.

[97] Y. Rudy. 'From Genes and Molecules to Organs and Organisms: Heart'. In: *Comprehensive Biophysics*. Elsevier, 2012, pp. 268–327. DOI: https://doi.org/10.1016/B978-0-12-374920-8.00924-3.

[98] Yoram Rudy and Jonathan R. Silva. 'Computational biology in the study of cardiac ion channels and cell electrophysiology'. In: *Quarterly Reviews of Biophysics* 39.1 (2006), pp. 57–116. DOI: https://doi.org/10.1017/S0033583506004227.

[99] Craig Saunders, Alexander Gammerman and Volodya Vovk. 'Ridge regression learning algorithm in dual variables'. In: *Proceedings of the 15-th International Conference on Machine Learning*. 1998, pp. 515–521.

[100] Robert E Schapire. 'The strength of weak learnability'. In: *Machine learning* 5 (1990), pp. 197–227. DOI: https://doi.org/10.1007/BF00116037.

[101] Bernhard Schölkopf, Ralf Herbrich and Alex J. Smola. 'A generalized representer theorem'. In: *Int. Conf. Comput. Learn. Theory.* 2001, pp. 416–426. DOI: https://doi.org/10.1007/3-540-44581-1_27.

[102] Bernhard Schölkopf, Alexander Smola and Klaus-Robert Müller. 'Kernel principal component analysis'. In: *International conference on artificial neural networks.* Springer. 1997, pp. 583–588. DOI: https://doi.org/10.1007/BFb0020217.

[103] Bernhard Schölkopf, Alexander Smola and Klaus-Robert Müller. 'Nonlinear component analysis as a kernel eigenvalue problem'. In: *Neural computation* 10.5 (1998), pp. 1299–1319. DOI: 10.1162/089976698300017467.

[104] Bernhard Schölkopf, Alexander J Smola, Francis Bach et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

[105] John Shawe-Taylor, Nello Cristianini et al. *Kernel methods for pattern analysis.* Cambridge university press, 2004.

[106] Yanning Shen, Tianyi Chen and Georgios B Giannakis. 'Random feature-based online multi-kernel learning in environments with unknown dynamics'. In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 773–808.

[107] Si Si, Cho-Jui Hsieh and Inderjit Dhillon. 'Memory efficient kernel approximation'. In: *International Conference on Machine Learning.* PMLR. 2014, pp. 701–709.

[108] Vikas Sindhwani, Partha Niyogi and Mikhail Belkin. 'Beyond the point cloud: from transductive to semi-supervised learning'. In: *Proceedings of the 22nd international conference on Machine learning.* 2005, pp. 824–831. DOI: https://doi.org/10.1145/1102351.1102455.

[109] Alexander J Smola. 'Sparse greedy matrix approximation for machine learning'. In: *Proceedings of the 17th international conference on machine learning, June 29-July 2 2000.* Morgan Kaufmann. 2000.

[110] Yue Song, David J Hill and Tao Liu. 'On extension of effective resistance with application to graph laplacian definiteness and power network stability'. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 66.11 (2019), pp. 4415–4428. DOI: 10.1109/TCSI.2019.2929180.

[111] Daniel Spielman. 'Spectral graph theory'. In: *Combinatorial scientific computing* 18 (2012).

[112] Daniel A Spielman and Shang-Hua Teng. 'Spectral partitioning works: Planar graphs and finite element meshes'. In: *Linear Algebra and its Applications* 421.2-3 (2007), pp. 284–305. DOI: https://doi.org/10.1016/j.laa.2006.07.020.

[113] Vivienne Sze et al. 'Efficient processing of deep neural networks: A tutorial and survey'. In: *Proceedings of the IEEE* 105.12 (2017), pp. 2295–2329. DOI: 10.1109/JPROC.2017.2761740.

[114] Kirsten HWJ Ten Tusscher and Alexander V Panfilov. ‘Alternans and spiral breakup in a human ventricular tissue model’. In: *American Journal of Physiology-Heart and Circulatory Physiology* 291.3 (2006), H1088–H1100. DOI: https://doi.org/10.1177/2168479018795117.

[115] Joshua B. Tenenbaum, Vin de Silva and John C. Langford. ‘A Global Geometric Framework for Nonlinear Dimensionality Reduction’. In: *Science* 290.5500 (2000), pp. 2319–2323. DOI: 10.1126/science.290.5500.2319.

[116] Reginald P Tewarson and Reginald P Tewarson. *Sparse matrices.* Vol. 69. Academic press New York, 1973.

[117] Aslak Tveito et al. ‘A Cell-Based Framework for Numerical Modeling of Electrical Conduction in Cardiac Tissue’. In: *Frontiers in Physics* 5 (2017). DOI: https://doi.org/10.3389/fphy.2017.00048.

[118] Aslak Tveito et al. ‘Inversion and computational maturation of drug response using human stem cell derived cardiomyocytes in microphysiological systems’. In: *Scientific reports* 8.1 (2018), pp. 1–14. DOI: https://doi.org/10.1038/s41598-018-35858-7.

[119] Stephen Tyree et al. ‘Parallel boosted regression trees for web search ranking’. In: *Proceedings of the 20th international conference on World wide web.* 2011, pp. 387–396. DOI: https://doi.org/10.1145/1963405.1963461.

[120] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik et al. ‘Dimensionality reduction: a comparative Review’. In: *Journal of Machine Learning Research* 10.66-71 (2009), p. 13.

[121] Vladimir Vapnik. *The nature of statistical learning theory.* Springer science & business media, 1999.

[122] Michel Verleysen and Damien François. ‘The Curse of Dimensionality in Data Mining and Time Series Prediction’. In: *Computational Intelligence and Bioinspired Systems.* 2005, pp. 758–770. DOI: https://doi.org/10.1007/11494669_93.

[123] Nakul Verma, Samory Kpotufe and Sanjoy Dasgupta. ‘Which spatial partition trees are adaptive to intrinsic dimension?’ In: *arXiv preprint arXiv:1205.2609* (2012).

[124] Ulrike Von Luxburg. ‘A tutorial on spectral clustering’. In: *Statistics and computing* 17 (2007), pp. 395–416. DOI: https://doi.org/10.1007/s11222-007-9033-z.

[125] Ulrike Von Luxburg, Agnes Radl and Matthias Hein. ‘Hitting and commute times in large random neighborhood graphs’. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1751–1798.

[126] Grace Wahba. *Spline models for observational data.* SIAM, 1990.

[127] Douglas Brent West et al. *Introduction to graph theory.* Vol. 2. Prentice hall Upper Saddle River, 2001.

[128] Christopher Williams and Matthias Seeger. ‘Using the Nyström method to speed up kernel machines’. In: *Advances in neural information processing systems* 13 (2000). URL: https://proceedings.neurips.cc/paper_files/paper/2000/file/19de10adbaa1b2ee13f77f679fa1483a-Paper.pdf.

[129] Elizabeth A Woodcock and Scot J Matkovich. ‘Cardiomyocytes structure, function and associated pathologies’. In: *The international journal of biochemistry & cell biology* 37.9 (2005), pp. 1746–1751. DOI: https://doi.org/10.1016/j.biocel.2005.04.011.

[130] Zhu Xiaojin and Ghahramani Zoubin. ‘Learning from labeled and unlabeled data with label propagation’. In: *Technical Report CMU-CALD-02–107*. Carnegie Mellon University, 2002.

[131] Zhenyue Zhang and Hongyuan Zha. ‘Nonlinear Dimension Reduction via Local Tangent Space Alignment’. In: *Intelligent Data Engineering and Automated Learning*. Berlin, Heidelberg, 2003, pp. 477–481. DOI: https://doi.org/10.1007/978-3-540-45080-1_66.

Original manuscripts

Paper I A. Oslandsbotn, Z. Kereta, V. Naumova, Y. Freund and A. Cloninger, ‘StreaMRAK a streaming multi-resolution adaptive kernel algorithm’. Published in *Applied Mathematics and Computation* (2022). (DOI: doi.org/10.1016/j.amc.2022.127112)

Paper II A. Oslandsbotn, A. Cloninger, and N. Forsch, ‘Improving inversion of model parameters from action potential recordings with kernel methods’. Submitted to *Mathematical Biosciences and Engineering* (2023). (BioRxiv: doi.org/10.1101/2023.03.15.532862)

Paper III R. Bhattacharjee, A. Cloninger, Y. Freund, and A. Oslandsbotn ‘Effective resistance in metric spaces’. Submitted to *Journal of Machine Learning Research* (2023). (Co-first-authors: Robi Bhattacharjee and Andreas Oslandsbotn) (arXiv: doi.org/10.48550/arXiv.2306.15649)

Paper IV R. Bhattacharjee, A. Cloninger, Y. Freund, and A. Oslandsbotn ‘Structure from voltage’. In preparation for journal submission. (Co-first-authors: Robi Bhattacharjee and Andreas Oslandsbotn) (arXiv: doi.org/10.48550/arXiv.2203.00063)

Paper I

StreaMRAK a streaming multi-resolution adaptive kernel algorithm

Andreas Oslandsbotn, **Željko Kereta**
Valeria Naumova, **Yoav Freund**
Alexander Cloninger

Published in Applied Mathematics and Computation

(doi.org/10.1016/j.amc.2022.127112)

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

StreaMRAK a streaming multi-resolution adaptive kernel algorithm[☆,☆☆]

Andreas Oslandsbotn[a,b,*], Željko Kereta[c], Valeriya Naumova[d], Yoav Freund[e], Alexander Cloninger[e]

[a] *University of Oslo, Problemveien 7, Oslo 0315, Norway*
[b] *Simula School of Research and Innovation, Martin Linges Vei 25, Fornebu 1364, Norway*
[c] *University College London, Gower St, London WC1E 6BT, England*
[d] *Simula Research Laboratory, Martin Linges vei 25, Fornebu 1364, Norway*
[e] *University of California San Diego, 9500 Gilman Dr, La Jolla CA 92093, United States*

ARTICLE INFO

Article history:
Received 15 August 2021
Revised 22 February 2022
Accepted 19 March 2022
Available online 9 April 2022

2021 MSC:
68Q32
65D15
46E22
68W27

Keywords:
Streaming
Reproducing kernel Hilbert space
Kernel methods
Laplcian pyramid
Adaptive kernel
Sub-sampling

ABSTRACT

Kernel ridge regression (KRR) is a popular scheme for non-linear non-parametric learning. However, existing implementations of KRR require that all the data is stored in the main memory, which severely limits the use of KRR in contexts where data size far exceeds the memory size. Such applications are increasingly common in data mining, bioinformatics, and control. A powerful paradigm for computing on data sets that are too large for memory is the *streaming model of computation*, where we process one data sample at a time, discarding each sample before moving on to the next one. In this paper, we propose StreaMRAK - a streaming version of KRR. StreaMRAK improves on existing KRR schemes by dividing the problem into several levels of resolution, which allows continual refinement to the predictions. The algorithm reduces the memory requirement by continuously and efficiently integrating new samples into the training model. With a novel sub-sampling scheme, StreaMRAK reduces memory and computational complexities by creating a *sketch* of the original data, where the sub-sampling density is adapted to the bandwidth of the kernel and the local dimensionality of the data. We present a showcase study on two synthetic problems and the prediction of the trajectory of a double pendulum. The results show that the proposed algorithm is fast and accurate.

1. Introduction

Machine learning algorithms based on kernel ridge regression (KRR) [1] is an active field of research [2–6], with applications ranging from time series prediction in finance [7], parameter inference in dynamical systems [8], to pairwise learning

[☆] Code is found at: https://github.com/AndOslandsbotn/StreaMRAK.git

[☆☆] Abbreviations: KRR (kernel regularized ridge regression), LP (Laplacian pyramid), CT (Cover-tree), DCT (Damped CT).

[*] Corresponding author.

E-mail addresses: andreas.oslandsbotn@gmail.com (A. Oslandsbotn), z.kereta@ucl.ac.uk (Ž. Kereta), valeriya@simula.no (V. Naumova), yfreund@eng.ucsd.edu (Y. Freund), acloninger@ucsd.edu (A. Cloninger).

https://doi.org/10.1016/j.amc.2022.127112

[9], face recognition [10] and drug estimation and gene analysis in biomedicine [11,12]. This paper develops a streaming variation of KRR using a radial kernel, a new sub-sampling scheme, and a multi-resolution formulation of the learning model.

Many popular data analysis software packages, such as Matlab$^{\text{TM}}$ require loading the entire dataset into memory. While computer memory is growing fast, the size of available data sets is growing much faster, limiting the applicability of *in-memory* methods.[1]

Streaming [13] is a computational model where the input size is much larger than the memory size. Streaming algorithms read one item at a time, update their memory, and discard the item. The computer memory is used to store a *model* or a *sketch* of the overall data distribution, which is orders of magnitude smaller than the data itself. The development of streaming algorithms is experiencing increased popularity in the face of big data applications such as data mining [14] and bioinformatics [15], where data sets are typically too large to be kept *in-memory*. Many big data applications call for non-linear and involved models, and thus, the development of non-parametric and non-linear models is critical for successful learning.

Among the most popular non-parametric learning algorithms are kernel methods, which include well-known learning schemes such as the support vector machine (SVM) and KRR, to name a few. The appeal of kernel methods lies in their strong theoretical foundation [1,16], as well as their ability to map complex problems to a linear space without requiring an explicit mapping. A common class of kernels are radial kernels $k(\mathbf{x}, \tilde{\mathbf{x}}) = \Phi(\|\mathbf{x} - \tilde{\mathbf{x}}\|/r)$ for $\mathbf{x}, \tilde{\mathbf{x}} \in \mathcal{X} \subseteq \mathbb{R}^D$ and bandwidth $r > 0$ [17]. These kernels are universal (with a few exceptions [18]), meaning that they can approximate any bounded continuous function on $\mathcal{X}$ arbitrarily well. However, in high dimensions, kernel methods suffer from the ”curse of dimensionality” and require large amounts of training data to converge. Furthermore, the computational complexity, memory requirement, and the number of parameters to learn grow unbounded with the number of training samples, a drawback known as the ”curse of kernelization” [19]. In the context of streaming, the prospect of unbounded data streams makes this shortcoming even more detrimental.

Although kernel-based learning schemes are typically formulated as convex optimization problems, which do not require tuning hyper-parameters such as learning rate etc., there is still a need to determine the optimal kernel. For the Gaussian kernel, this amounts to selecting the bandwidth. Classically, an optimal kernel is chosen through batch techniques such as leave-one-out and k-fold cross-validation [20–22]. However, these approaches are inefficient as they spend significant time evaluating bad kernel hypotheses and often use multiple runs over the data, which is impossible in a streaming setting.

To meet a need for non-linear non-parametric algorithms for streaming data, we propose the *streaming multi-resolution adaptive kernel algorithm* (StreaMRAK) - a computationally and memory-efficient streaming variation of KRR. StreaMRAK address the kernel selection problem with a multi-resolution kernel selection strategy that adapts the sub-sample density to the kernel bandwidth over several levels of resolution. Furthermore, StreaMRAK addresses the curse of dimensionality and kernelization in a novel way, through the sub-sampling scheme.

1.1. Setting

We consider a finite sample data-cloud $\mathcal{X}$, $|\mathcal{X}| = n$, that is sampled i.i.d. according to a fixed but unknown distribution $\mathcal{P}$ over $\mathbb{R}^D$. The target is a bounded and continuous function $f : \mathbb{R}^D \to \mathbb{R}$. We assume that the points in $\mathcal{X}$ are placed in a *sequence* and that their order is random. [2] Each instance $\mathbf{x}_i \in \mathcal{X}$, for $i \in [n]$, paired with a label y_i where $y_i = f(\mathbf{x}_i) + \varepsilon_i$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma)$ represents noise. The task of learning is to train a model $\widehat{f}$ that is a good approximation of the target function f.

In this work, we think about the intrinsic dimension of $\mathcal{X}$ as a local quantity, meaning it depends on the region $\mathcal{A} \subseteq \mathcal{X}$ and the radius r at which we consider the point cloud. To estimate the local intrinsic dimension in a ”location and resolution sensitive” manner, we use the concept of the doubling dimension of a set, defined in Def. 1.2. See [23,24] for related definitions.

Definition 1.1 (Covering number)**.** Consider a set $\mathcal{A}$ and a ball $\mathcal{B}(\mathbf{x}, r)$, with $r > 0$ and $\mathbf{x} \in \mathcal{A}$. We say that a finite set $\mathcal{S} \subset \mathcal{B}(\mathbf{x}, r)$ is a covering of $\mathcal{B}(\mathbf{x}, r)$ in $\mathcal{A}$ if $\mathcal{A} \cap \mathcal{B}(\mathbf{x}, r) \subset \cup_{\mathbf{x}_i \in \mathcal{S}} \mathcal{B}(\mathbf{x}_i, r/2)$. We define the covering number $\kappa(\mathcal{A}, \mathbf{x}, r)$ as the minimum cardinality of any covering of $\mathcal{B}(\mathbf{x}, r)$ in $\mathcal{A}$.

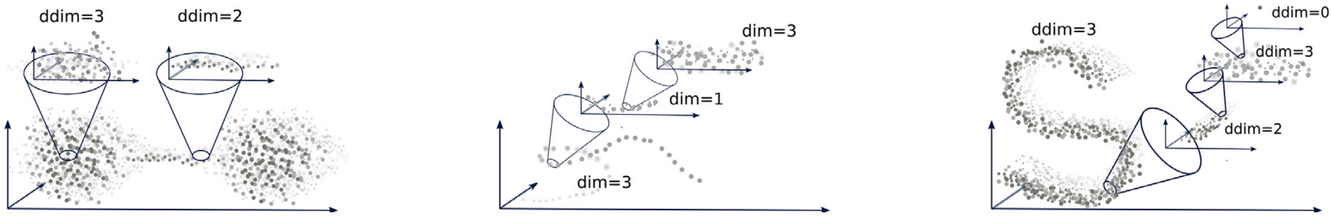
Definition 1.2 (Doubling dimension)**.** The doubling dimension $\texttt{ddim}(\mathcal{A}, r)$ of a set $\mathcal{A}$ is defined as $\texttt{ddim}(\mathcal{A}, r) = \lceil \log \kappa(\mathcal{A}, \mathbf{x}, r) \rceil$. For an interval $\mathcal{I} \subset \mathbb{R}_{>0}$ we define the doubling dimension as the least upper bound over $r \in \mathcal{I}$, that is $\texttt{ddim}(\mathcal{A}, \mathcal{I}) = \max_{r \in \mathcal{I}} \texttt{ddim}(\mathcal{A}, r)$.

We say that the intrinsic dimension of $\mathcal{X}$ changes with the location if there exist $\mathcal{A}_1, \mathcal{A}_2 \subset \mathcal{X}$ such that $\texttt{ddim}(\mathcal{A}_1, r) \neq \texttt{ddim}(\mathcal{A}_2, r)$ for $r > 0$. Similarly, we say that the intrinsic dimensionality of $\mathcal{X}$ changes with the resolution, if there exist $r_1 \neq r_2$ such that the doubling dimension $\texttt{ddim}(\mathcal{A}, r_1) \neq \texttt{ddim}(\mathcal{A}, r_2)$ for $\mathcal{A} \subset \mathcal{X}$.

In Fig. 1 we consider three examples to provide further insight on the doubling dimension. In Fig. 1a we see a domain shaped like a dumbbell, where the spheres are high dimensional, and the bar connecting them is lower-dimensional,

[1] Simulink$^{\text{TM}}$, a companion software to Matlab$^{\text{TM}}$ supports streaming but has a much more limited computational model, targeted at signal processing applications.

[2] The assumption that the sequence is randomly ordered allows us to draw statistical conclusions from prefixes.

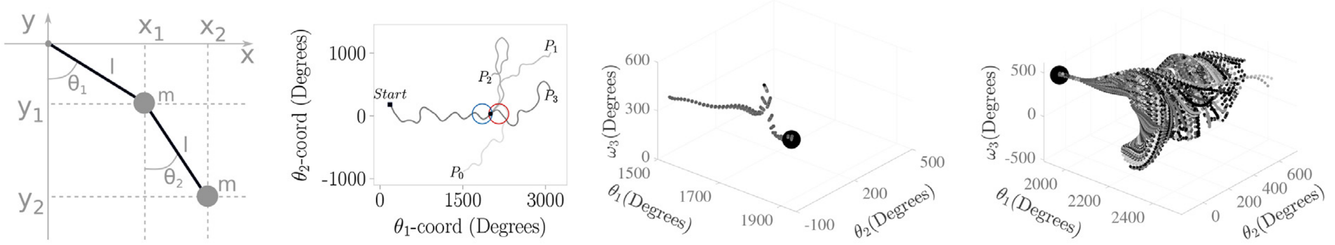


(a) Intrinsic dimension changes with location.

(b) Noise has high intrinsic dimension.

(c) Intrinsic dimension changes with the resolution.

Fig. 1. Three examples of variation in the intrinsic dimension. The coloring of the point clouds illustrates depth.



(a) Double pendulum

(b) 2-dim projection of the pendulum training data $\mathcal{X}$.

(c) 3-dim projection of the training data in the blue circle in (b).

(d) 3-dim projection of the training data in the red circle in (b).

Fig. 2. (a) Illustration of a double pendulum. Here l and m are the length and mass of the pendulum rods, and θ_1, θ_2 are the angles. Furthermore, (x_1, y_1) and (x_2, y_2) are the positions of the point masses of the two pendulums. (b) Phase diagram of four double pendulums P_0, P_1, P_2, P_3, iterated for $T = 500$ time steps. The bifurcation point at step $T = 300$ is indicated with a black solid circle. (c) and (d) includes the ω_1 axis and zoom in on respectively the blue and red circles in (b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

showing how the dimension can change with the location. Meanwhile, Fig. 1b illustrates a lower-dimensional manifold, embedded in $\mathbb{R}^3$, with manifold noise ζ_m. We see that when the resolution is sufficiently small, so that $r \approx \zeta_m$, the doubling dimensionality increases towards the dimension of the ambient space $\mathbb{R}^D$. Furthermore, Fig. 1c shows a point cloud that is locally 2-dimensional, but is embedded in a 3-dimensional space. By reducing r we can resolve this lower dimensionality, but if it is reduced further, we would eventually resolve the noise level, and the doubling dimension increases again.

As an example of how the intrinsic dimension might change with respect to regions and resolutions, we consider a double pendulum system, a well-known chaotic system that depends heavily on its initial conditions [25]. Systems with multiple pendulum elements are well known in engineering applications such as mechanical and robotic systems with several joints and are studied for their chaotic properties [26].

In Fig. 2b we visualize the trajectory of four pendulums P_0, P_1, P_2, P_3, for which the trajectories are indistinguishable until a bifurcation occurs around $T = 300$ time steps, and the trajectories start to diverge. In Fig. 2c and Fig. 2d we zoom in on the trajectory of all 500 pendulums in regions before and after the bifurcation. These two regions, $\mathcal{A}_1$ and $\mathcal{A}_2$, are indicated by a blue and red circle, respectively, in Fig. 2b. From the figures, it is clear that learning the trajectory in $\mathcal{A}_1$ is significantly easier than in $\mathcal{A}_2$, where learning the trajectory is more affected by the curse of dimensionality.

1.2. Contribution and comparison to related work

Contributions of this work can be divided into three components.

(C1) A multi-resolution variation of the state-of-the-art KRR solver FALKON [2], using the LP, which refines the predictions at each level of resolution by regressing on the errors from the previous level.
(C2) A novel sub-sampling scheme for kernel methods, tailored for use in combination with the LP, that can handle the curse of dimensionality and does not require the data to be *in-memory*.
(C3) Development of a streaming variation of FALKON, where the time and memory requirements depend on the doubling dimensionality and the level of resolution, instead of the number of training points. see Props. 5.3–5.5.

In the following, we give further details on these contributions and compare them to related work. The computational backbone of StreaMRAK is based on the state-of-the-art KRR solver FALKON [2], which among other things combines sub-sampling and preconditioning to process large data sets efficiently. However, FALKON relies on selecting an optimal kernel bandwidth, which can be inefficient in streaming.

Our first contribution (C1) addresses the issue of selecting an optimal bandwidth by introducing a multi-resolution reformulation of FALKON using a changing bandwidth variation of the LP [27–29]. This strategy is inspired by the success of existing multi-resolution approaches [4,30–36] and, for online learning, adaptive bandwidth approaches [37–39]. Our scheme combines the LP with a localized kernel, which gives a frequency and location-based discretization, similar to wavelet analysis that have shown great success in numerous applications. However, typical wavelet architectures [40–46] require upfront construction of a wavelet basis, which is not compatible with a data-adaptive kernel.

In this work, we aim to show that the LP is a viable multi-resolution scheme and can be modified to the streaming setting. Furthermore, we provide convergence bounds for the LP in the context of radial kernels and KRR, and show experimentally that it improves the estimation accuracy.

Let us now discuss our second contribution (C2). FALKON addresses the curse of kernelization by combining Nyström sub-sampling, conjugate gradient, and preconditioning, and achieves time and memory requirements of $\mathcal{O}(n\sqrt{n})$ and $\mathcal{O}(n)$ respectively, where n is the number of samples. In recent years there have been several efforts to address the curse of kernelization in similar ways through sub-sampling techniques such as sketching [3,5], randomized features [47–50] and Nyström sub-sampling [51–55]. However, despite their successes, these techniques are in principle *in-memory* type algorithms since they require access to the training data in advance of the training and are not optimized for streaming.

Furthermore, FALKON selects the sub-samples uniformly over the input domain $\mathcal{X}$. However, when learning with a radial kernel, the density of samples should be related to the bandwidth of the kernel. Otherwise, a too-small bandwidth will lead to bad interpolation properties, while a too-large bandwidth gives an ill-conditioned system [34]. Since the LP scheme reduces the kernel bandwidth at each level of resolution, it would be problematic to use the same sub-sample density. Furthermore, due to the curse of dimensionality, the covering number increases exponentially with the doubling dimension. Therefore, if doubling dimensionality varies across different regions of the domain $\mathcal{X}$, as illustrated by Fig. 1a, then the number of sub-samples necessary to maintain the density for a given bandwidth will also vary.

Our second contribution (C2) provides an alternative sub-sampling strategy, adapting the sub-sampling density to the kernel bandwidth. This strategy is based on a damped cover-tree (DCT), which is a modified version of the cover-tree (CT) [24], a tree-based data structure with $\mathcal{O}(n)$ memory and $\mathcal{O}(c^6 n \log n)$ time.

A problem with an adaptive sub-sampling strategy is its vulnerability to the curse of dimensionality. In regions of high doubling dimensions, the number of samples to achieve a certain density increases exponentially, as quantified by Def. 1.2. This means that the number of sub-samples from the CT will quickly grow too large for efficient computing. The danger is to waste resources on samples from subsets and levels where the doubling dimension is so large that good interpolation cannot be achieved for any viable sample sizes. This would only serve to slow down the computation and not increase the precision.

Due to this, the DCT introduces a damping property, which gradually suppresses the selection of sub-samples where the doubling dimensionality is large. This has the additional advantage of allowing to choose more sub-samples from regions where the doubling dimensionality is small. Thus, the DCT can diminish the impact of the curse of dimensionality. Furthermore, the DCT can be built continuously as new samples come in, making it ideal for a streaming computational model.

Our third contribution (C3), is the streaming capabilities of StreaMRAK. In particular, the sub-sampling and kernel construction allows for continuous integration of new training points. Furthermore, the DCT, the multi-resolution construction, and the KRR solver can all be multi-threaded and parallelized.

1.3. Organization of the paper

The paper is organized as follows. Section 2 introduces kernel methods and the FALKON algorithm, as well as the LP. Section 3 introduces the adaptive sub-sampling scheme and the DCT. StreaMRAK is described in Section 4 and an analysis of the algorithm is given in Section 5. Finally, Section 6 presents several numerical experiments and Section 7 gives an outlook for further work. The Appendix includes further mathematical background and the proofs.

1.4. Notation

We denote vectors $\mathbf{a} \in \mathbb{R}^D$ with boldface and matrices $\mathbf{A} \in \mathbb{R}^{n\times m}$ with bold uppercase, and $\mathbf{A}^\top$ denotes the matrix transpose. We use $\mathbf{K}_{nm}$ for kernel matrices, where the dimensionality is indicated by the subscripts. We reserve n for the number of training samples and m for the number of sub-samples. The ij-th element of a kernel matrix is denoted $[\mathbf{K}_{nm}]_{ij}$, while for other matrices we use $\mathbf{A}_{ij}$. The notation a_i indicates i-th element of a vector $\mathbf{a}$. Furthermore, we use $f([\mathbf{x}_n])$ to denote $(f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))^\top \in \mathbb{R}^n$, and $[m]$ to denote $\{i\}_{i=1}^m$. The notation $\mathbf{x}_i$ indicates the i-th training example. We use $\mathbf{a}^{(l)}$ and $\mathbf{A}^{(l)}$, where l refers to a specific level in the LP and the DCT. We take $\|\cdot\|$ to be the L^2 norm and $\|\cdot\|_{\mathcal{H}}$ to be the RKHS norm. We denote the intrinsic dimension of a manifold with d and the dimension of the embedding with D. By $\mathbb{1}_{\mathcal{S}}(\mathbf{x})$ we denote the indicator function, which evaluates to 1 if $\mathbf{x} \in \mathcal{S}$ and 0 otherwise, of a set $\mathcal{S} \subset \mathbb{R}^D$.

2. Kernel methods

Consider a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, defined on an input space $\mathcal{X} \subset \mathbb{R}^D$. Given data $\{(\mathbf{x}_i, y_i) : i \in [n]\}$ of samples from $\mathcal{X} \times \mathbb{R}^D$, kernel ridge regression computes an estimator by minimising

$$\widehat{f}_{n,\lambda} = \underset{f\in\widetilde{\mathcal{H}}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i) - y_i)^2 + \lambda\|f\|_{\mathcal{H}}^2,$$

where $\mathcal{H}$ is the Hilbert space induced by the kernel. This allows to reduce the problem to a linear system

$$(\mathbf{K}_{nn} + \lambda n \mathbf{I}_n)\alpha = \mathbf{y}, \text{ for } [\mathbf{K}_{nn}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j), \text{ and } \mathbf{y} = (y_1, \dots, y_n)^\top. \tag{2.1}$$

Coefficients $\alpha = (\alpha_1, \dots, \alpha_n)^\top$ define the estimator by $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i k(\mathbf{x}, \mathbf{x}_i)$. However, solving (2.1) using traditional methods has a time complexity of $\mathcal{O}(n^2)$, which can be costly for large n [2].

FALKON [2] addresses this issue by sub-sampling the columns of $\mathbf{K}_{nn}$, which reduces the effective complexity while maintaining accuracy. Namely, denote $\Gamma_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and for $m \ll n$ let $\widetilde{\Gamma}_m = \{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_m\}$ be Nyström centers (i.e. a randomly selected subset of Γ_n). Minimizing

$$\widehat{f}_{n,m,\lambda} = \underset{f \in \widetilde{\mathcal{H}}_M}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n (f(\mathbf{x}_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2, \tag{2.2}$$

where $\widetilde{\mathcal{H}}_m = \overline{\operatorname{span}}\{k(\cdot, \tilde{\mathbf{x}}_j) : j \in [m]\}$, leads to a linear system

$$\mathbf{H}\widetilde{\alpha} = \mathbf{z}, \text{ for } \mathbf{H} = \mathbf{K}_{nm}^\top \mathbf{K}_{nm} + \lambda n \mathbf{K}_{mm}, \text{ and } \mathbf{z} = \mathbf{K}_{nm}\mathbf{y}.$$

Here $[\mathbf{K}_{nm}]_{ij} = k(\mathbf{x}_i, \tilde{\mathbf{x}}_j) \in \mathbb{R}^{n \times m}$ is the column-subsampled matrix and the estimator is given by $\widehat{f}_{n,m,\lambda}(\mathbf{x}) = \sum_{j=1}^m \widetilde{\alpha}_j k(\mathbf{x}, \tilde{\mathbf{x}}_j)$. To further reduce the time complexity FALKON uses a suitable preconditioner to reduce the condition number. The preconditioner is defined as $\mathbf{B}\mathbf{B}^\top = (n/m\mathbf{K}_{mm}^2 + \lambda n \mathbf{K}_{mm})^{-1}$, which is a natural (lower complexity) approximation of the ideal preconditioner $\mathbf{A}\mathbf{A}^\top = (\mathbf{K}_{nm}^\top \mathbf{K}_{nm} + \lambda n \mathbf{K}_{mm})^{-1}$. We now solve for $\widetilde{\alpha}$ from the system of equations

$$\mathbf{B}^\top \mathbf{H} \mathbf{B} \beta = \mathbf{B}^\top \mathbf{z}, \text{ for } \mathbf{H} = \mathbf{K}_{nm}^\top \mathbf{K}_{nm} + \lambda n \mathbf{K}_{mm}, \ \mathbf{z} = \mathbf{K}_{nm}\mathbf{y}, \text{ and } \widetilde{\alpha} = \mathbf{B}\beta. \tag{2.3}$$

This is solved iteratively, using the conjugate gradients with early stopping. Choosing $m = \mathcal{O}(\sqrt{n})$ still ensures optimal generalisation (i.e. same as KRR), while reducing the computational complexity to $\mathcal{O}(n\sqrt{n})$.

2.1. Streaming adaptation of FALKON

Matrices and vectors involved in the linear system in (2.3) can be separated into two classes: those that depend only on sub-samples in $\widetilde{\Gamma}_m$; and those ($\mathbf{K}_{nm}^\top \mathbf{K}_{nm}$ and $\mathbf{z}$) that also depend on all the training points Γ_n. Critically, terms in both groups are all of size m, which allows to reduce the complexity. Consider now the set of sub-samples $\widetilde{\Gamma}_m$ to be fixed, and assume new training points, in the form $\{(\mathbf{x}_q, y_q) : q = n+1, \dots, n+t)\}$, are coming in a stream. We can then update the second class of terms according to

$$\left[(\mathbf{K}_{(n+t)m})^\top \mathbf{K}_{(n+t)m}\right]_{ij} = \left[(\mathbf{K}_{nm})^\top \mathbf{K}_{nm}\right]_{ij} + \sum_{q=n+1}^{n+t} k(\mathbf{x}_q, \tilde{\mathbf{x}}_i) k(\mathbf{x}_q, \tilde{\mathbf{x}}_j), \tag{2.4}$$

$$\left[(\mathbf{K}_{(n+t)m})^\top \mathbf{y}\right]_i = z_i + \sum_{q=n+1}^{n+t} k(\mathbf{x}_q, \tilde{\mathbf{x}}_i) y_q. \tag{2.5}$$

Thus, only sub-samples $\widetilde{\Gamma}_m$, matrices $\left(\mathbf{K}_{nm}\right)^\top \mathbf{K}_{nm}$, $\mathbf{K}_{mm}$ and $\mathbf{z}$, need to be stored. However, in order to continuously incorporate new training points into Eqs. (2.4) and (2.5), sub-samples $\widetilde{\Gamma}_m$ must be determined in advance. Whereas this works if all the data is provided beforehand, it cannot be done if the data arrives sequentially. In this work, we address this through a multi-resolution framework. The overall estimator is composed of a sequence of estimators defined at different resolution levels of the domain. Correspondingly, the set of sub-samples $\widetilde{\Gamma}_m$ consists of smaller sets $\widetilde{\Gamma}_{m^{(l)}}^{(l)}$ that correspond to individual levels of resolution. The sets $\widetilde{\Gamma}_{m^{(l)}}^{(l)}$ are filled as the data streams in, and once a set for a given level is deemed complete, we proceed with updating (2.4) and (2.5).

Further details of how the sets $\widetilde{\Gamma}_{m^{(l)}}^{(l)}$ are constructed, and the corresponding criteria, are provided in Sections 3 and 4. We begin by describing the multi-resolution framework of estimators.

2.2. The laplacian pyramid

The LP [27,29] is a multi-resolution regression method for extending a model $\widehat{f}$ to out-of-sample data points $\mathbf{x} \in \mathcal{X}/\Gamma_n$. The LP can be formulated for radial kernels in the form

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{r}\right), \tag{2.6}$$

where $r > 0$ is a shape parameter that determines the decay of Φ with respect to $\|\mathbf{x}_i - \mathbf{x}_j\|$, see [17]. The idea underpinning the LP is to approximate the target function sequentially, where at each stage we regress on the errors from the previous stage. In other words, we begin with a rough approximation using a large shape parameter for which Φ decays slowly and then improve the approximation by fitting the resulting error and reducing the shape parameter. In the LP, the estimator at level $L \in \mathbb{N}$ is defined recursively as

$$\widehat{f}^{(L)}(\mathbf{x}) = \sum_{l=0}^{L} s^{(l)}(\mathbf{x}) = s^{(L)}(\mathbf{x}) + \widehat{f}^{(L-1)}(\mathbf{x}), \tag{2.7}$$

where $\hat{f}^{(0)} = s^{(0)}$, and $s^{(l)}(\mathbf{x})$ is a correction term defined by

$$s^{(l)}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i^{(l)} k^{(l)}(\mathbf{x}, \mathbf{x}_i). \tag{2.8}$$

The coefficients $\alpha^{(l)} = (\alpha_1^{(l)}, \ldots, \alpha_n^{(l)})^\top$ are computed by conducting KRR on the residuals, i.e. errors, from the estimator at the previous level. Namely, $\alpha^{(l)} = (\mathbf{K}_{nn}^{(l)} + \lambda n\mathbf{I})^{-1}\mathbf{d}^{(l)}$, where

$$\mathbf{d}^{(l)} = \begin{cases} \mathbf{y}, & \text{if} \quad l = 0 \\ \mathbf{y} - \widehat{f}^{(l-1)}([\mathbf{x}_n]), & \text{otherwise} \end{cases}. \tag{2.9}$$

For a FALKON adaption of this scheme, we only need to modify how per-level coefficients are computed. Following (2.3) we iteratively solve

$$(\mathbf{B}^{(l)})^\top \mathbf{H}^{(l)} \mathbf{B}^{(l)} \beta^{(l)} = (\mathbf{B}^{(l)})^\top \left(\mathbf{K}_{nm}^{(l)}\right)^\top \mathbf{d}^{(l)}, \tag{2.10}$$

where $\mathbf{B}^{(l)}$ is the corresponding preconditioner, and $\mathbf{H}^{(l)} = (\mathbf{K}_{nm}^{(l)})^\top \mathbf{K}_{nm}^{(l)} + \lambda n \mathbf{K}_{mm}^{(l)}$, and set $\tilde{\alpha}^{(l)} = \mathbf{B}^{(l)}\beta^{(l)}$.

Remark 2.1. In this paper, we construct the kernel matrices $\mathbf{K}^{(l)}$ on a particular class of radial kernels, namely the Gaussian kernel

$$k^{(l)}(\mathbf{x}, \tilde{\mathbf{x}}_i) = \exp\left(-\frac{\|\mathbf{x} - \tilde{\mathbf{x}}_i\|^2}{2r_l^2}\right),$$

where $r_l > 0$ is the shape parameter (the kernel bandwidth) at level l.

3. The damped cover tree

This work introduces a data-driven sub-sampling method that we call the damped cover-tree (DCT). The DCT is a modification of the cover-tree (CT) [24], a data structure based on partitioning a metric space, initially designed to facilitate nearest neighbor search. The goal of the DCT is to modify and simplify the CT to allow a viable sub-sampling scheme.

Let $(\mathcal{X}, \|\cdot\|)$ be a normed space where the input domain $\mathcal{X} \subset \mathbb{R}^D$ is bounded, such that the diameter $r_0 = \mathrm{diam}(\mathcal{X})$ is finite. The DCT is a tree structure where each node p of the tree is associated with a point $\mathbf{x}_p \in \mathcal{X}$, and which is built sequentially as data points arrive. Furthermore, let Q_l be a set (herein called a cover-set) containing all the nodes at a level $l \geq 0$ in the given tree. A level is associated with an integer l and a radius $r_l = 2^{-l} r_0$, where $l = 0$ denotes the root level containing only one node and l increases as we descend deeper into the tree. DCT has three invariants, of which the first two are also invariants of the CT.

(I1) (Covering invariant) For all $p \in Q_{l+1}$ there exists $q \in Q_l$ such that $\|\mathbf{x}_q - \mathbf{x}_p\| < r_l$.
(I2) (Separation invariant) For all $q, p \in Q_l$ where $\mathbf{x}_q \neq \mathbf{x}_p$, we have $\|\mathbf{x}_q - \mathbf{x}_p\| > r_l$.

We add that the standard CT includes a third invariant, the so-called nesting invariant, which requires $Q_l \subseteq Q_{l+1}$, but this is not desired for our purpose.

To introduce the last invariant of the DCT, we first need the following definition.

Definition 3.1 (The covering fraction)**.** Let $p \in Q_l$ be a node, and $\mathbf{x}_p$ the associated point in $\mathcal{X}$. Furthermore, let $\widetilde{C}_p = \{c_i\}_{i=1}^k$ be the children of p, and $\mathbf{x}_{c_i}$ the corresponding points in $\mathcal{X}$. The covering fraction of a node p is defined as

$$\mathfrak{cf}(p) = \frac{\mathrm{Vol}\left(\mathcal{X} \cap \mathcal{B}(\mathbf{x}_p, r_l) \cap \bigcup_{c_i \in \widetilde{C}_p} \mathcal{B}(\mathbf{x}_{c_i}, r_{l+1})\right)}{\mathrm{Vol}\left(\mathcal{X} \cap \mathcal{B}(\mathbf{x}_p, r_l)\right)}.$$

The covering fraction is the proportion of the volume of $\mathcal{B}(\mathbf{x}_p, r_l)$ that is covered by balls around its children of half the radius. This quantity is directly related to (I2), which enforces the radius r_l to reduce by a factor of 2 for each new level, starting from an initial radius $r_0 > 0$. The covering fraction allows us to capture the vulnerability of the standard CT to the curse of dimensionality.

For example, consider two regions $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{X}$, for which the doubling dimension at radius r_l is $\mathtt{ddim}(\mathcal{A}_1, r_l) > \mathtt{ddim}(\mathcal{A}_2, r_l)$. A node $p \in \mathcal{A}_1$ at level l will then need exponentially more children to be covered, than a node $q \in \mathcal{A}_2$ at the same level l. This exacerbates the deeper we go into the tree. Therefore, the CT would have significantly more nodes from regions where the doubling dimension is large.

We recall now that sub-sampling is in kernel methods intended to reduce the computational complexity. For this purpose, it is desirable to keep the number of sub-samples from each level within a budget of reasonable size. On the other hand, a too low sub-sample density will lead to poor interpolation performance. Due to the exponential growth of the number of nodes with respect to the doubling dimension, it would be desirable to avoid wasting our budget on sub-samples from regions and radii with a large doubling dimension, as this would require dedicating an (exponentially) large number of points to achieve good interpolation, which is not feasible. Moreover, in high dimensional regions, we likely cannot learn anything more than a simple function, for which a lower sampling density would suffice.

To reduce the number of sub-samples from regions of large doubling dimensionality, we introduce the following damping invariant as the third invariant of the DCT.

(I3) (Damping invariant) Let $\mathcal{D}_{\mathfrak{cf}} \in (0,1)$ be some threshold and let $\widetilde{C}_p$ and $\mathfrak{cf}(p)$ be as in Def. 3.1. Then any node q whose parent node p does not satisfy $\mathfrak{cf}(p) \geq \mathcal{D}_{\mathfrak{cf}}$ does not have children of its own.

The damping invariant forces the tree to devote more resources to regions of lower doubling dimension by making it harder for nodes in regions with higher doubling dimensions to have children. In other words, the practical effect of the damping invariant is to stop the vertical growth of the DCT if the doubling dimension becomes large. This is because the covering number grows exponentially with the dimensionality, ensuring $\mathfrak{cf}(p) \geq \mathcal{D}_{\mathfrak{cf}}$ gets correspondingly harder to achieve.

Remark 3.2. In Section 5.1, we analyze the damping invariant in more detail and show how the damping suppresses vertical growth of the DCT more for regions of high doubling dimension than for regions of lower doubling dimensionality.

3.1. Construction of the DCT

We now discuss how the DCT is constructed and updated as the data streams in. First, it is important to restate that we use the DCT to replace the Nyström sampling, which was in FALKON used to reduce the complexity of the ridge regressor. Consequently, not all of the streamed data (that is, not every training point) will be added to the tree, but only those whose inclusion into the tree would not violate the invariants (I1)-(I3). In other words, the tree consists of only those training points that help resolve the data space at the relevant resolution level. Thus, each node p in the DCT is associated with a unique training sample $\mathbf{x}_p$, but not every training sample will be represented by a node in the tree. Note that this is different from the standard CT, which aims to organize all of the training data into a geometrical leveled data structure.

The construction of the DCT consists of a series of checks which examine whether adding a given data point to the DCT would, or would not, violate invariants (I1)- (I3). When a new point $\mathbf{x}_q$ arrives from the data stream the goal is to identify the deepest level l for which there exists a node p such that $\|\mathbf{x}_q - \mathbf{x}_p\| \leq r_l$. This corresponds to finding the nearest node in the tree that could serve as a parent.

We achieve this in the following way. The first training point is identified as the root node to which we associate the radius r_0. For each new point, we proceed in a top-down manner, starting from the root node[3]. We then check whether $\mathbf{x}_q$ would violate the separation invariant at the next level. In other words, if there exists a node p such that $\|\mathbf{x}_q - \mathbf{x}_p\| < r_l$. If such a node does not exist, then $\mathbf{x}_q$ is added to the set of children of the root node, and we update the covering fraction estimate for the root node. Otherwise, if such a node does exist, we repeat the process, checking the separation invariant among the children of the corresponding node, and proceed further down the tree.

Assume we arrived to a node p at level $l \geq 1$, and we have $\|\mathbf{x}_q - \mathbf{x}_p\| \leq r_l$. We then check if p is allowed to have children, that is if the damping invariant is satisfied. If it is not satisfied, the point $\mathbf{x}_q$ is dismissed (it is not added to the tree). On the other hand, if p is allowed to have children, we check whether the separation invariant holds, i.e., if there exists a child c of the node p such that $\|\mathbf{x}_q - \mathbf{x}_c\| < r_{l+1}$. If that were the case, the separation invariant would be violated, and the recursion is applied again by considering c as the potential parent node. However, if such a child does not exist, that is, if the separation invariant is not violated, then $\mathbf{x}_q$ is added to the set of children of the node p. More details are given in Alg. A.1.

Some comments are needed to elucidate how are the steps described above applied in practice. First, note that the covering fraction from Def. 3.1 cannot be calculated explicitly, since the volume terms require knowing the intrinsic dimensionality. Therefore, it is necessary to use an estimator instead. For this purpose, we interpret $\mathfrak{cf}(p)$ as the probability that a sample $\mathbf{x} \sim \text{Uni}(\mathcal{B}(\mathbf{x}_p, r))$ will be within $\mathcal{B}_c := \bigcup_{c_i \in \widetilde{C}_p} \mathcal{B}(\mathbf{x}_{c_i}, r/2)$, where $\widetilde{C}_p$ are the children of p. This probability can be estimated by considering the checks of the separation invariant (I2), conducted on the last N points that were inside $\mathcal{B}(\mathbf{x}_p, r)$, as a series of independent random trials. We use the following running average as an estimator of the covering fraction

$$(\mathfrak{cf}(p))_t = (1-\alpha)(\mathfrak{cf}(p))_{t-1} + \alpha \mathbb{1}_{\mathcal{B}_c}(\mathbf{x}_t), \tag{3.1}$$

where $\mathbb{1}_{\mathcal{B}_c}(\mathbf{x}_t)$ is the indicator function, and $\alpha > 0$ is a weighting parameter. This approximates a weighted average of the outcome of the N last draws (cf. Appendix B). Note that this reduces the memory requirements, since instead of storing N trial outcomes for each node in the tree, as required had we used an average of the last N trials, we store only a single value for each node in the tree.

Second, the separation invariant is in practice too strict since it results in too few points added to the tree, and thus a worse kernel estimator. Moreover, checking the separation invariant adds to the computational complexity. Therefore, we

[3] We assume that all new points $\mathbf{x}_q$ are within a ball of radius r_0 around this node, which holds for a large enough r_0

introduce the following relaxation. Assume we have a new point $\mathbf{x}_q$ and arrived at a node p at level l. We then first conduct a random Bernoulli trial, with the failure probability

$$q_{\mathbf{x}} = \frac{1}{1 + \exp\left[h \tan\left(\pi\left(\|\mathbf{x}_q - \mathbf{x}_p\|/r_l - \frac{1}{2}\right)\right)\right]}, \tag{3.2}$$

where h is the hardness of the threshold. In other words, the probability of failure is proportional to the distance between $\mathbf{x}_q$ and $\mathbf{x}_p$ - the larger the distance, the more likely the failure. If the trial's outcome is a failure, then the check for the separation invariant is ignored, and the algorithm continues. If it is a success, we proceed by first checking the separation invariant. This means that the probability to ignore the separation invariant increases as $\mathbf{x}_q$ gets farther from $\mathbf{x}_p$.

3.2. Sub-sampling from the DCT

We now discuss how the DCT is used for sub-sampling the training points. By organizing the training points into cover-sets Q_l the DCT allows a hierarchical sub-sampling. Even though cover-sets Q_l significantly reduce the number of training points, they are for practical purposes still too large for efficient sub-sampling. Due to this, we restrict ourselves to a subset $\widetilde{\Gamma}^{(l)} \subseteq Q_l$ of candidate sub-samples called landmarks.

Definition 3.3 (Landmarks)**.** Let Q_l be the cover-set at level l in a DCT. We define the set of candidate landmarks at level l as $\widetilde{\Gamma}^{(l)} = \{\mathbf{x}_p \mid p \in Q_l \text{ and } \mathfrak{cf}(p) \geq \mathcal{D}_{\mathfrak{cf}}\}$, and the set of landmarks (of size m) as any subset $\widetilde{\Gamma}^{(l)}_m = \{\tilde{\mathbf{x}}^{(l)}_1, \dots, \tilde{\mathbf{x}}^{(l)}_m\} \subset \widetilde{\Gamma}^{(l)}$ of size m.

Some remarks are in order. First, by Def. 3.3, candidates for landmarks at level l are only those nodes allowed to have children (according to the damping invariant (I3)). This design choice implies that the set of candidate landmarks will contain more points from regions with a lower doubling dimension than points from regions with a higher doubling dimension. This is because the larger the doubling dimension is, the more children nodes are needed to cover a given parent node.

Second, Def. 3.3 suggests using only a subset of candidate landmarks as sub-samples. We refer to a result from [2] which states that good statistical accuracy of the estimator is achieved if the number of sub-samples is proportional to the square root of the number of samples. At level l we therefore use a set of landmarks which is of size $m^{(l)} = \delta_0\sqrt{|Q_l|}$, where $\delta_0 > 0$ is a constant.

The third point that requires attention concerns the question of when the landmarks should be selected. To that end, we use the covering fraction of a level, which, with a slight abuse of notation, we denote as $\mathfrak{cf}(Q_l)$. Moreover, we compute $\mathfrak{cf}(Q_l)$ as

$$(\mathfrak{cf}(Q_l))_t = (1-\alpha)(\mathfrak{cf}(Q_l))_{t-1} + \alpha \mathbb{1}_{\mathcal{B}_{\text{level}}}(\mathbf{x}_t), \tag{3.3}$$

where $\mathcal{B}_{level} = \bigcup\limits_{p \in Q_l} \mathcal{B}(\mathbf{x}^{(l)}_p, r_l)$. Moreover, analogously to the damping invariant, let $\mathcal{D}_{level} \in (0, 1)$ be some threshold. We then say that a level l is sufficiently covered when $\mathfrak{cf}(Q_l) \geq \mathcal{D}_{level}$.

Remark 3.4. We note that as the level increases, our estimate of $\mathfrak{cf}(Q_l)$ through Eq. (3.3) will be increasingly more sensitive to subsets $\mathcal{A} \subset \mathcal{X}$ of low doubling dimension than to subsets of large doubling dimension. This is because the damping invariant (I3) makes it harder for nodes in high dimensions to have children. Consequently, we will have fewer points in deeper levels that belong to high dimensional regions. Because of this, the estimator in Eq. (3.3) is biased towards using more sub-samples from lower dimensional regions.

Sub-sampling from a level l goes as follows. As training points arrive, we build the tree and continuously update the covering fraction of a level. Once that level is sufficiently covered, that is, once $\mathfrak{cf}(Q_l) \geq \mathcal{D}_{level}$, we extract the set of landmarks by sub-sampling $m^{(l)}$ points from the pool of candidate landmarks $\widetilde{\Gamma}^{(l)}$.

4. StreaMRAK

In this section, we present StreaMRAK and clarify how it synthesizes concepts from Sections 2 and 3, and utilizes them in a streaming context. The workflow of StreaMRAK can be divided into three threads that can run in parallel, subject to some inter-dependencies. These are the sub-sampling thread, the training thread, and the prediction thread. Overviews of these threads are given next, and the reader is referred to Algorithm A.2 in the Appendix for further details.

4.1. Sub-sampling thread

In the sub-sampling thread StreaMRAK collects and organizes the training data into a DCT. Namely, as new training pairs are collected, the covering (I1) and separation (I2) are checked, and the covering fraction is updated as described in Section 3.1. Moreover, the set of landmarks for each level is updated, as described in Section 3.2. Once the set of landmarks for a given level $\widetilde{\Gamma}^{(l)}_m$ is completed, the landmarks and the estimator for the corresponding level can be used in the remaining two threads.

4.2. Training thread

The model is trained at level l when two conditions are met. First, coefficients of the previous level $l-1$ in the LP must have been calculated, i.e. previous training thread must finish. Second, landmarks $\widetilde{\Gamma}^{(l)}_{m^{(l)}}$ at level l must be ready.

In the first step, we define the kernel matrix on the landmarks by

$$[\mathbf{K}^{(l)}_{mm}]_{ij} = k^{(l)}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j), \ \text{ for } \tilde{\mathbf{x}}_i \in \widetilde{\Gamma}^{(l)}. \tag{4.1}$$

In the second step we consider $\left(\mathbf{K}^{(l)}_{nm}\right)^{\top}\mathbf{K}^{(l)}_{nm} \in \mathbb{R}^{m^{(l)}\times m^{(l)}}$ and $\left(\mathbf{K}^{(l)}_{nm}\right)^{\top}\mathbf{d}^{(l)}_{n} \in \mathbb{R}^{m^{(l)}}$ which in addition to landmarks depend on the training points. They are updated continuously as new training points come in, according to Eq. (2.4) and Eq. (2.5). However, they are not updated indefinitely, but only until new training points do not significantly alter the matrices according to the following criterion.

Definition 4.1. (Sufficient training points) Let $\mathbf{A}_n := (\mathbf{K}^{(l)}_{nm})^{\top}\mathbf{K}^{(l)}_{nm}$, and $\mathbf{b}_n := \left(\mathbf{K}^{(l)}_{nm}\right)^{\top}\mathbf{d}^{(l)}_{n}$. Let $\delta_1, \delta_2, \delta_3 > 0$ be three constants. We consider the number of training points at a level l sufficient when either $n \geq \delta_3$ or

$$\left\| \frac{\mathbf{A}_n}{n} - \frac{\mathbf{A}_{n+1}}{n+1} \right\|_{\infty} \leq \delta_1 \ \text{and} \ \left\| \frac{\mathbf{b}_n}{n} - \frac{\mathbf{b}_{n+1}}{n+1} \right\| \leq \delta_2.$$

After enough training samples are collected according to Def. 4.1, the correction term $s^{(l)}$ is obtained by solving for the coefficients $\widetilde{\alpha}^{(l)}_1, \ldots, \widetilde{\alpha}^{(l)}_{m^{(l)}}$ using Eq. (2.10). The new prediction model $\widehat{f}^{(L)}$ is obtained by adding $s^{(l)}$ to the previous model, according to Eq. (2.7).

4.3. Prediction thread

In this thread StreaMRAK makes provides the latest version of the trained LP model in Eq. (2.7). This means that if L is currently the highest level that has been trained, the prediction for new points $\mathbf{x}$ is made using the model $\widehat{f}^{(L)}(\mathbf{x})$.

5. Analysis

In this section, we first analyze the damping invariant of the DCT. We then offer theoretical results on the convergence properties of the LP in the context of KRR. Finally, we offer estimates of the time and memory requirements of StreaMRAK. We introduce two auxiliary results.

Lemma 5.1. *Consider a ball $\mathcal{B}(\mathbf{x}, r) \in \mathbb{R}^D$ and let $\delta > 0$. The number of points in any (discrete) set of points within $\mathcal{B}(\mathbf{x}, r)$ that are at least δ apart, $S = \{\mathbf{x}_i \in \mathcal{B}(\mathbf{x}, r) | d(\mathbf{x}_i, \mathbf{x}_j) \geq \delta \text{ for } i \neq j\}$, is bounded by $|S| \leq \left(\frac{2r}{\delta} + 1\right)^D$.*

Proof. Since the points in S are at least δ apart, it follows that the balls $\mathcal{B}(\mathbf{x}_i, \delta/2)$ are disjoint. Consider now the ball $\mathcal{B}(\mathbf{x}, r+\delta/2)$. All of the balls $\mathcal{B}(\mathbf{x}_i, \delta/2)$ are entirely contained within $\mathcal{B}(\mathbf{x}, r+\delta/2)$. Since the balls $\mathcal{B}(\mathbf{x}_i, \delta/2)$ are disjoint, it follows that

$$|S| \leq \mathrm{Vol}\left(\mathcal{B}(\mathbf{x}, r+\delta/2)\right) / \mathrm{Vol}\left(\mathcal{B}(\mathbf{x}_i, \delta/2)\right) = \left(\frac{2r}{\delta} + 1\right)^D.$$

□

Lemma 5.2. *Consider a domain $\mathcal{X} \in \mathbb{R}^D$, a ball $\mathcal{B}(\mathbf{x}_p, r) \subset \mathcal{X}$ and let $S = \{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{B}(\mathbf{x}_p, r) | \|\mathbf{x}_i - \mathbf{x}_j\| \geq \delta \text{ for } i \neq j\}$. Furthermore, let the doubling dimension of the set S be $\mathtt{ddim} := \mathtt{ddim}(S, r)$. We let $c_d := |S|$ when $\mathfrak{cf}(p) = 1$. We then have $2^{\mathtt{ddim}-1} \leq c_d \leq 5^{\mathtt{ddim}}$.*

Proof. The upper bound on c_d follows from Lemma 5.1 with $r = r_0$ and $\delta = r_0/2$. The lower bound follows from the definition of the doubling dimension 1.2. □

5.1. Analysis of the DCT

As discussed in Section 3, the DCT adds a given training point to the set of nodes of the tree if conditions (I2) and (I3) are satisfied, and the points are otherwise discarded. In particular, the damping invariant (I3) makes it harder for a node to have children. The guiding idea is that damping should reduce the impact of the curse of dimensionality by making it harder for nodes in regions of higher doubling dimension to have children, and in doing so it should effectively stop the vertical growth of the tree in corresponding regions. Therefore, it is critical to understand how and to what degree the damping affects high dimensional regions more than low dimensional ones.

In a statistical sense, the damping should treat all nodes in regions of the same doubling dimension equally. Therefore, to gain insight into the damping, it suffices to analyze its effects concerning the doubling dimension on a single node p. In this case, the effect of damping can be measured by analyzing how many training points must pass through p, in the sense of Alg. A.1, before children of p are allowed to have children of their own. This can be modeled by considering the expected number of training points $\mathbf{x}_i \sim \mathrm{Uni}\big(\mathcal{B}(\mathbf{x}_p, r)\big)$ necessary to cover $\mathcal{B}(\mathbf{x}_p, r)$ with balls of radius $r/2$ around points $\mathbf{x}_i$.

Consider $\mathtt{x}_i \sim \mathrm{Uni}\big(\mathcal{B}(\mathtt{x}_p, r)\big)$, and let a set $\mathcal{S}_p$ be built in a succession of trials $i = 1, \ldots, N_t$ so that

$$\mathtt{x}_i \in \mathcal{S}_p \text{ if } \|\mathtt{x}_i - \mathtt{x}\| \geq \frac{r}{2} \text{ for all } \mathtt{x} \in \mathcal{S}_p.$$

In other words, a newly sampled point $\mathtt{x}_i$ will only be added to the set $\mathcal{S}_p$ if it its pairwise distances from all the points that are already in $\mathcal{S}_p$ are at least $r/2$.

Problem 1. Let $\widetilde{C}_p$ denote the set of children of the node p, constructed from the above-described trials. What is the expected number of trials N_t needed to ensure $\mathfrak{cf}(p) = 1$?

Since there is no unique set $\mathcal{S}_p$ such that the corresponding set of children $\widetilde{C}_p$ ensures $\mathfrak{cf}(p) = 1$, the sample space for Problem 1 corresponds to all admissible sets $\mathcal{S}_p$, which vary in both the number and the location of points they contain. Characterizing all such sets corresponds to a disordered sphere packing problem [56], which is an NP-hard combinatorial problem [57]. For a theoretical analysis of this problem, defining a probability measure over the sample space is necessary. However, in this level of generality, neither the sample space nor the probability measure admit a workable definition, with currently available mathematical tools [56]. Although some theoretical insights are possible under simplifications on the sample space, this analysis is restrained to a limited number of spheres and configurations.

Due to these difficulties, we consider a simplified setting where we instead consider an average case. If the set $\mathcal{S}_p$ is such that $\mathcal{B}(\mathtt{x}_i, r) \subset \bigcup_{\mathtt{x}_i \in \mathcal{S}_p} \mathcal{B}(\mathtt{x}_i, r/2)$, which corresponds to $\mathfrak{cf}(p) = 1$, then each of the balls $\mathcal{B}(\mathtt{x}_i, r/2)$ occupies on average $\frac{1}{|\mathcal{S}_p|}$ of the total volume of $\mathcal{B}(\mathtt{x}_p, r)$, assuming none of the balls are covered by a union of other balls. Therefore, as $\mathcal{S}_p$ is being built, adding a point to $\mathcal{S}_p$ will, on average, reduce the unoccupied volume of $\mathcal{B}(\mathtt{x}_p, r)$ by $\frac{1}{|\mathcal{S}_p|}$. Moreover, it can be shown that the number of elements in such a set satisfies $2^{\mathtt{ddim}-1} \leq |\mathcal{S}_p| \leq 5^{\mathtt{ddim}}$, see Lemma 5.2, where $\mathtt{ddim} := \mathtt{ddim}(\mathcal{S}_p, r)$ is the doubling dimension of $\mathcal{S}_p$. Based on these considerations we introduce a simplified setting for the average case of Problem 1.

Assumption 1. Problem 1 can be approximated by dividing the ball $\mathcal{B}(\mathtt{x}_p, r)$ into a union of c_d fixed (and known) disjoint bins $\mathcal{B}_i$ of size $(1/c_d)\,\mathrm{Vol}\big(\mathcal{B}(\mathtt{x}_p, r)\big)$.

Note that the bins referred to in Assumption 1 correspond to regions around the children of the node p. Assumption 1 reduces the average case of Problem 1 to a form of the classical coupons collector's problem [58], which considers n coupons with the same probability of being drawn. Through a series of randomized trials with replacement, the goal is to obtain a copy of each coupon. Relevant for Problem 1 is estimating the stopping time T, which counts the number of trials before all coupons are collected, and which satisfies $\mathbb{E}[T] = nH_n$, where n denotes the number of coupons and H_n is the n-th harmonic number [58].

In terms of Problem 1, and under Assumption 1, we can therefore identify $T = N_t$, $n = |\mathcal{S}_p|$ and $\mathbb{E}[N_t|\mathrm{Node}\, p] = |\mathcal{S}_p|H_{|\mathcal{S}_p|}$. Combining the bound $\ln(n) + \frac{1}{2} \leq H_n \leq \ln(n) + 1$ (from [59]), with the bound on $|\mathcal{S}_p|$ from Lemma 5.2 we have

$$2^{\mathtt{ddim}-1}\big((\mathtt{ddim} - 1)\ln 2 + 1/2\big) \leq \mathbb{E}[N_t|\mathrm{Node}\, p] \leq 5^{\mathtt{ddim}}(\mathtt{ddim} \ln 5 + 1). \tag{5.1}$$

With the same strategy, we can bound the number of trials until the cover-fraction of a level reaches 1, as

$$2^{l(\mathtt{ddim}-1)}\big(l(\mathtt{ddim} - 1)\ln 2 + 1/2\big) \leq \mathbb{E}[N_t|\mathrm{Level}\, l] \leq 5^{l\mathtt{ddim}}(l\mathtt{ddim} \ln 5 + 1). \tag{5.2}$$

From Eq. (5.1) we see that the number of training points $\mathbb{E}[N_t|\mathrm{node}\, p]$ grows exponentially with the doubling dimensionallity d. In other words, significantly more trials are needed to achieve $\mathfrak{cf}(p) = \mathcal{D}_{\mathfrak{cf}}$ for nodes in regions with a large doubling dimension than it is for nodes in regions with a lower doubling dimension. Consequently, through the damping invariant, the DCT restricts the vertical growth of the tree comparatively more the higher the doubling dimension of the local region.

5.2. Time and memory requirements

This section analyzes the memory requirements of StreaMRAK, which involve storing the DCT and the linear system components used to update the coefficients. Furthermore, we consider the computational requirements, which consist in solving the coefficient equations. Both the memory and computational requirements need to be analyzed per level l of the tree due to the multi-resolution nature of the estimator and the tree organization of the data.

For the analysis, we consider a simplified setting where we assume that the doubling dimension is constant for all levels and all subsets of $\mathcal{X}$, and that the number of children c_d is the same for all nodes. At the end of the section we describe a more general setting.

In the following, we assume that the growth of the DCT stops at a level L. In other words, level L is the last level at which there are nodes. In practice, the growth of the DCT slows down exponentially fast with the product of the doubling dimension $\mathtt{ddim} := \mathtt{ddim}(\mathcal{X}, r_L)$ and the level l. This can be seen from Eq. (5.2), which shows that the number of training points necessary to fill up a level grows exponentially with $l\mathtt{ddim}$. Therefore, in practice, no new levels will be added to the DCT when $l\mathtt{ddim}$ is large enough, which effectively makes the last level L independent of the number of training points.

Furthermore, from Lemma 5.2 we know that c_d is bounded by $2^{\mathtt{ddim}-1} \leq c_d \leq 5^{\mathtt{ddim}}$, which shows that also c_d is independent of the number of training points.

Proposition 5.3. *The memory requirement of StreaMRAK is* $\mathcal{O}\big(\sum_{l=0}^{L} c_d^l\big)$.

Proof. The memory requirement of the DCT is determined by the number of nodes in the tree. Given that the number of children is the same for all nodes. If the number of children per node is c_d, then the total number of nodes at level l is c_d^l. Thus, the memory needed to store the DCT with L levels is $\mathcal{O}(\sum_{l=0}^{L} c_d^l)$.

To store the linear system on level l we need the matrices $\big(\mathbf{K}_{nm^{(l)}}\big)^\top \mathbf{K}_{nm^{(l)}}$, $\mathbf{K}_{m^{(l)}m^{(l)}} \in \mathbb{R}^{m^{(l)} \times m^{(l)}}$ and the vector $\mathbf{z} \in \mathbb{R}^{m^{(l)}}$. The number of landmarks $m^{(l)}$ at level l is chosen as $m^{(l)} = \delta_0 \sqrt{|Q_l|}$, where $|Q_l|$ is the number of nodes at level l. Since $|Q_l|$ is $\mathcal{O}(c_d^l)$, it follows that $m^{(l)} \times m^{(l)}$ is also $\mathcal{O}(c_d^l)$ per level, and the desired conclusion follows. □

Note that with a fixed L and n larger than $\mathcal{O}\big(\sum_{l=0}^{L} c_d^l\big)$, then the memory requirement is independent of n. We also note that if the deepest level satisfies $L \to \infty$, then the number of nodes is determined by the number of training points, and the memory requirement would thus, in the worst case, become $\mathcal{O}(n)$, the same as for the standard cover-tree.

Next, we discuss the construction of the DCT, where adding a new point to the set of nodes requires a search through the tree.

Proposition 5.4. *Inserting a new point into the DCT, cf. Algorithm A.1, requires* $\mathcal{O}(c_d L)$ *operations.*

Proof. For a point $\mathbf{x}_q \in \mathcal{X}$ to be analyzed at level L, we need to have analyzed it at the previous $l < L$ levels. At each level, we must, in the worst case, check the separation invariant with all children of the current potential parent $p^{(l)}$, before finding a node c such that $\|\mathbf{x}_q - \mathbf{x}_c\| \leq 2^{-l} r_0$, that would serve as the next potential parent. This requires at most c_d operations per level, giving Lc_d total operations over the L levels. The same number of operations is necessary if a node is discarded at level L. □

Lastly, we analyze the computational requirements for solving the linear system.

Proposition 5.5. *The time requirement for solving the linear system in Eq. (2.3) is* $\mathcal{O}\big(\delta_3 m^{(l)} + \big(m^{(l)}\big)^3\big)$ *per level, where* δ_3 *is given in Def. 4.1,*

Proof. The time requirement of FALKON is $\mathcal{O}(nmt + m^3)$ where n is the number of training points, m the number of landmarks and t the number of iterations of the conjugate gradient (which has an upper bound). By Def. 4.1, StreaMRAK uses at most δ_3 training samples at each level. Since $m^{(l)}$ is the number of landmarks at level l, the result follows. □

Assume that the domain $\mathcal{X}$ can be divided into disjoint subsets $\mathcal{A}_1, \dots, \mathcal{A}_t \subset \mathcal{X}$ for which the doubling dimension $\mathtt{ddim}(\mathcal{A}_i, r_l)$ differs based on $\mathcal{A}_i$ and radius r_l. Let the number of children of a node $\mathbf{x}_p \in \mathcal{A}_i$ at level l be $c_{d,i,l}$. In this scenario, the growth of the DCT will stop at different levels L_i for different subsets $\mathcal{A}_i$. The final time and memory requirements would therefore be the sum of the contribution from each subset $\mathcal{A}_i$. In other words, the memory would be $\mathcal{O}(\sum_{i=1}^{t} \sum_{l=0}^{L_i} c_{d,i,l}^l)$, and similarly the time requirement per point insertion would be $\mathcal{O}(\sum_{i=1}^{t} \sum_{l=0}^{L_i} c_{d,i,l})$. We note that $c_{d,i,l}$ and L_i depend on the dimensionality of the data, but are independent of n. Therefore, so are the time and memory requirements.

5.3. Convergence of the LP formulation of the KRR

This section analyzes the conditions for which the LP approximates the training data $y_i = f(\mathbf{x}_i)$, with respect to the number of levels. A similar analysis was previously done for the LP in the context of kernel smoothers [28]. However, to the best of our knowledge, this is the first time the LP formulation of KRR has been analyzed in this way.

Theorem 5.6. *Let* $\widehat{f}^{(l)}$ *be the LP estimator defined in Eq. (2.7) and let* λ *be a regularization parameter. Furthermore, let* $0 < \sigma_{l,n} \leq \dots \leq \sigma_{l,1}$ *be the eigenvalues of* $\mathbf{K}_{nn}^{(l)}$. *For* $L > 0$ *we then have*

$$\|\widehat{f}^{(L+1)}([\mathbf{x}_n]) - f([\mathbf{x}_n])\| \leq \prod_{l=0}^{L} (1 - \varepsilon(l)) \|\widehat{f}^{(0)}([\mathbf{x}_n]) - f([\mathbf{x}_n])\|, \quad \text{where} \quad \varepsilon(l) = \frac{\sigma_{l,n}}{n\lambda + \sigma_{l,n}}.$$

Proof. From the recurrence relationship for the residuals $\mathbf{d}^{(l)}$ in Eq. (2.9) it follows by induction that

$$\widehat{f}^{(l+1)}([\mathbf{x}_n]) - f([\mathbf{x}_n]) = (\mathbf{I} - \mathbf{P}_{nn}^{(l)})(\widehat{f}^{(l)}([\mathbf{x}_n]) - f([\mathbf{x}_n]), \tag{5.3}$$

where $\mathbf{P}_{nn}^{(l)} := \mathbf{K}_{nn}^{(l)} (\mathbf{K}_{nn}^{(l)} + \lambda n \mathbf{I})^{-1}$, cf. Lemma C.1. It follows that

$$\|\widehat{f}^{(l+1)}([\mathbf{x}_n]) - f([\mathbf{x}_n])\| \leq \|\mathbf{I} - \mathbf{P}_{nn}^{(l)}\| \|\widehat{f}^{(l)}([\mathbf{x}_n]) - f([\mathbf{x}_n])\|. \tag{5.4}$$

Consider the SVD $\mathbf{K}_{nn}^{(l)} = \mathbf{U\Sigma U}^\top$ where $\mathbf{\Sigma} = \operatorname{diag}(\sigma_{l,i})$ and $\sigma_{l,n} \leq \cdots \leq \sigma_{l,1}$. We then have

$$\|\mathbf{I} - \mathbf{P}_{nn}^{(l)}\| = \left\|\mathbf{U}\operatorname{diag}\left(\frac{n\lambda}{n\lambda + \sigma_{l,i}}\right)\mathbf{U}^\top\right\| = \left\|\operatorname{diag}\left(\frac{n\lambda}{n\lambda + \sigma_{l,i}}\right)\right\| = \frac{n\lambda}{n\lambda + \sigma_{l,n}} := 1 - \varepsilon(l), \tag{5.5}$$

and Thm. 5.6 follows recursively from Eq. (5.4) and Eq. (5.5). □

From Thm. 5.6 it follows that the LP estimator will converge as $l \to \infty$, since $\sigma_{l,n} > 0$ and therefore $1 - \varepsilon(l) \in (0, 1)$ for all l. In Thm. 5.7 we characterise how $\varepsilon(l)$ depends on the level l to give insight on the nature of this convergence.

Theorem 5.7. *The LP estimator $\widehat{f}^{(l)}$ from Eq. (2.7) converges with increasing level L to the training data $f(\mathbf{x}_i)$, cf. Thm. 5.6, with the rate $\prod_{l=0}^{L}(1 - \varepsilon(l))$, where*

$$1 - \varepsilon(l) \leq \left(1 + C_{1,D}2^{-Dl}\exp\left(-C_{2,D}4^{-l}\right)/n\lambda\right)^{-1}, \tag{5.6}$$

for

$$C_{1,D} = \frac{1}{2}(6\sqrt{2})^D\Gamma(D/2+1)^{\frac{D-1}{D+1}}\left(\frac{\pi}{9}\right)^{\frac{D}{D+1}}\left(\frac{r_0}{\delta}\right)^D \quad \text{and} \quad C_{2,D} = 1152\left(\frac{\pi\Gamma^2(D/2+1)}{9}\right)^{\frac{2}{D+1}}\left(\frac{r_0}{\delta}\right)^2,$$

where Γ is the gamma function.

Furthermore, for $l > \log_2(\sqrt{D/2}(r_0/\delta))$ we have the tighter bound

$$1 - \varepsilon(l) < \left(1 + \left(1 - 2^{1+\frac{1}{\ln 2}(C_3D - g(l))}\right)/n\lambda\right)^{-1}, \tag{5.7}$$

where $g(l) = 4^{l-\log_2 r_0/\delta}$ and $C_3 = (\ln(1 + 1/4) + 2\ln 2)$.

Proof. (Eq. (5.6)) We bound $1 - \varepsilon(l) := n\lambda/(n\lambda + \sigma_{l,n})$ by bounding the smallest eigenvalue of the kernel matrix $[\mathbf{K}_{nn}^{(l)}]_{ij} = \Phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$, namely $\sigma_{l,n}$. To do so we assume that there exists a lower bound on the minimal distance between any two points $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, defined as $\delta := \min\limits_{i \neq j \in \mathcal{X}} \|\mathbf{x}_i - \mathbf{x}_j\| > 0$.

Consider the Gaussian $\Phi(\mathbf{x}) = \exp(-\beta\|\mathbf{x}\|_2^2)$, $\beta > 0$, with the Fourier transform $\widehat{\Phi}(\omega) = (\pi/\beta)^{D/2}\exp(-\|\omega\|_2^2/4\beta)$. From [60, Corollary 12.4] we have the bound

$$\sigma_{l,n} \geq C_D 2^D (2\beta)^{-D/2}\delta^{-D}\exp(-4M_D^2/(\delta^2\beta)),$$

where

$$M_D = 12\left(\frac{\pi\Gamma^2(D/2+1)}{9}\right)^{1/(D+1)} \quad \text{and} \quad C_D = \frac{1}{2\Gamma(D/2+1)}\left(\frac{M_D}{2^{3/2}}\right)^D.$$

With $\beta = (\sqrt{2}2^{-l}r_0)^{-2}$ and inserting for M_D and C_D we then have

$$\begin{aligned}\sigma_{l,n} &\geq C_D 2^D 2^{-Dl}\left(\tfrac{r_0}{\delta}\right)^D \exp\left(-(2\sqrt{2}M_D)^2(r_0/\delta)^2 4^{-l}\right) \\ &= \tfrac{1}{2}(6\sqrt{2})^D\Gamma(D/2+1)^{\frac{D-1}{D+1}}\left(\tfrac{\pi}{9}\right)^{\frac{D}{D+1}}\left(\tfrac{r_0}{\delta}\right)^D 2^{-Dl} \\ &\quad \cdot \exp\left(-1152\left(\tfrac{\pi\Gamma^2(D/2+1)}{9}\right)^{\frac{2}{D+1}}\left(\tfrac{r_0}{\delta}\right)^2 4^{-l}\right) := B(l),\end{aligned} \tag{5.8}$$

The bound in Eq. (5.6) follows from this result. □

Proof. (Eq. (5.7)) When the level l becomes sufficiently large, the kernel matrix $\mathbf{K}_{nn}^{(l)}$ becomes diagonally dominant, and we can therefore bound the eigenvalues using Garschgorins Theorem [61, Thm. 1.1], which gives

$$|\sigma_{l,i} - [\mathbf{K}_{nn}^{(l)}]_{jj}| = |\sigma_{l,i} - 1| < \sum_{\substack{q=1,\\ q\neq j}}^{n} |[\mathbf{K}_{nn}^{(l)}]_{jq}| \quad \text{for} \quad i, j \in [n]. \tag{5.9}$$

To find a more explicit bound, we analyze the sum on the right-hand side. Consider a family of annuli $\{R_t\}_{t=0}^{\infty}$ where $R_t = \mathcal{B}(\mathbf{x}_j, 2^{t+1}\delta)\backslash\mathcal{B}(\mathbf{x}_j, 2^t\delta)$. Inspired by [28], we can interpret the right hand side of Eq. (5.9) as a sum over $\{R_t\}_{t=0}^{\infty}$. The entries of $\mathbf{K}_{nn}^{(l)}$ are defined as

$$[\mathbf{K}_{nn}^{(l)}]_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2r_l^2}\right), \quad \forall i, j \in [n],$$

where $r_l = 2^{-l}r_0$ for $r_0 > 0$. It follows

$$\sum_{\substack{q=1,\\ q\neq j}}^{n} |[\mathbf{K}_{nn}^{(l)}]_{jq}| = \sum_{t=0}^{\infty}\sum_{\mathbf{x}_q \in R_t} k^{(l)}(\mathbf{x}_j, \mathbf{x}_q) \leq \sum_{t=0}^{\infty}\left(\tfrac{2^{t+2}\delta}{\delta} + 1\right)^D \exp\left(-(2^t\delta 2^{-1/2}r_l^{-1})^2\right),$$

where in the first term on the right-hand side we bound the number of summands using Lemma 5.1, and in the second we use $\|\mathbf{x}_q - \mathbf{x}_j\| \geq 2^t\delta$ for $\mathbf{x}_q \in R_t$. Note now that for all $T \geq 1$ there exists $C_T > 0$ such that $\exp(-r^2) \leq C_T r^{-T}$ holds for all $r > 0$. Such a constant is given by the Lambert W function and satisfies $C_T = \left(\frac{T}{2e}\right)^{T/2}$. Moreover, $2^{t+2} + 1 \leq 2^{t+2+\alpha}$, for $\alpha \geq \ln(1+1/4)/\ln(2)$. Thus,

$$\sum_{\substack{q=1,\\ q\neq j}}^{n} |[\mathbf{K}_{nn}^{(l)}]_{jq}| \leq C_T\left(\frac{r_l}{\delta}\right)^T 2^{(2+\alpha)D+T/2}\sum_{t=0}^{\infty}2^{t(D-T)} \leq 2\cdot 2^{D(2+\alpha)-T/2(1+1/\ln(2))}T^{T/2}\left(\frac{r_l}{\delta}\right)^T,$$

where in the last step we have used $\exp(1) \geq 2^{1+1/\ln(2)}$ along with $\sum_{t=0}^{\infty} 2^{t(D-T)} \leq 2$, which holds for $D - T < 0$. We let $r_l = r_0 2^{-l}$ and define $F(T) := 2^{-T/2(1+1/\ln(2))}T^{T/2}2^{-lT}\left(\frac{r_0}{\delta}\right)^T$. From Lemma C.4 we have the minimum of $F(T)$, which together with the choice $\alpha = \ln(1+1/4)/\ln 2$, gives

$$\sigma_{l,n} > 1 - \sum_{\substack{q=1,\\ q\neq j}}^{n}\left|\left[\mathbf{K}_{nn}^{(l)}\right]_{jq}\right| \geq 1 - 2^{1+\frac{1}{\ln 2}((\ln(1+1/4)+2\ln 2)D-g(l))}, \quad g(l) = 4^{l-\log_2 r_0/\delta}.$$

By defining $C_3 = (\ln(1+1/4)+2\ln 2)$ and using that $1-\varepsilon(l) := n\lambda/(n\lambda+\sigma_{l,n})$ this leads to the bound in Eq. (5.7). We note that this bound holds for $T^* > D$ which means that $l > \log_2(\sqrt{D/2}r_0/\delta)$. □

We note that the bound in Eq. (5.6) underestimates the rate of convergence for lower levels but improves as the levels increase. Furthermore, Thm. 5.7 shows that the convergence rate increases with the level l. In fact, the bound in Eq. (5.6) can be simplified with an *a fortiori* bound of the same form, where $C_{1,D} = \frac{1}{2}\left(\frac{12.76}{2^{3/2}}\right)^D\left(\frac{D^D}{\Gamma(D/2+1)}\right)\left(\frac{r_0}{\delta}\right)^D$ and $C_{2,D} = (12.76\sqrt{2}D)^2(r_0/\delta)^2$, which ensures that $1-\varepsilon(l)$ decreases monotonically for $l < \log_2(\sqrt{D/2}(r_0/\delta)) + \log_2(25.52\sqrt{2})$. see Remark C.2 and Corollary C.3.

On the other hand, when $l > \log_2(\sqrt{D/2}(r_0/\delta))$ the tighter bound from Eq. (5.7) ensures that $1-\varepsilon(l)$ continues to decreases monotonically. Moreover, as $l \to \infty$ each new level reduces the residual error by $(1+1/n\lambda)^{-1}$. We can also observe that the convergence rate is reduced by the number of training points n, but this effect can be mitigated by reducing the regularization parameter λ. We also note that Thm. 5.6 and Thm. 5.7 are derived for a vector of numbers on the training data $\Gamma_n \subset \mathcal{X}$, without assumptions on the target function. In other words, the LP estimator can approximate the training data for any function $f : \Gamma_n \to \mathbb{R}$, to arbitrary precision, by including sufficiently many levels.

Corollary 5.8. *If the residual* $\mathbf{d}^{(l)} = (\widehat{f}^{(l)}([\mathbf{x}_n]) - f([\mathbf{x}_n]))$ *at level* l *only projects non-trivially onto the eigenvectors with eigenvalue* $\sigma_{l,n} \geq \sigma_{\text{cut-off}}$, *then we say the residual is spectrally band-limited with respect to the kernel. If the residual* $\mathbf{d}^{(l)}$ *is spectrally band-limited, then* $1-\epsilon(l) < n\lambda/(n\lambda+\sigma_{\text{cut-off}})$.

Proof. Follows from Eq. (5.3)-(5.5) with $P_{nn}^{(l)} = P_{nn,k}^{(l)} + \left(P_{nn,k}^{(l)}\right)^{\perp}$, where $P_{nn,k}^{(l)}$ is the projection on the eigenvectors associated with the k largest eigenvalues and $\left(P_{nn,k}^{(l)}\right)^{\perp}(\widehat{f}^{(l)}([\mathbf{x}_n]) - f([\mathbf{x}_n])) = 0$. □

6. Experiments

This section presents comparative numerical experiments of the proposed estimator on three problems. In Section 6.1 we consider a one-dimensional regression problem, and in Section 6.2 we consider a dumbbell-shaped domain that consists of two 5-dimensional spheres connected by a 2-dimensional plane. Lastly, in Section 6.3, we forecast the trajectory of a double pendulum, which is a well-known chaotic system [25].

We compare StreaMRAK with FALKON [2] and an LP modification of KRR (LP-KRR). Both FALKON and LP-KRR rely on the standard Nyström sub-sampling [51,52]. Furthermore, FALKON does not rely on a multi-resolution scheme but uses instead a single bandwidth, found by cross-validation.

Throughout the experiments, we set the threshold for the number of sub-samples (landmarks) in StreaMRAK to be $10\sqrt{|Q_l|}$, where Q_l is the set of nodes at level l in the DCT. We note that to choose the sub-sample size, FALKON and LP-KRR require n to be known beforehand. For FALKON we let the number of Nyström landmarks be $10\sqrt{n}$, where n is the number of training samples. Meanwhile, for LP-KRR we sub-sample $\sqrt{n}$ Nyström landmarks, which are then used for all levels.

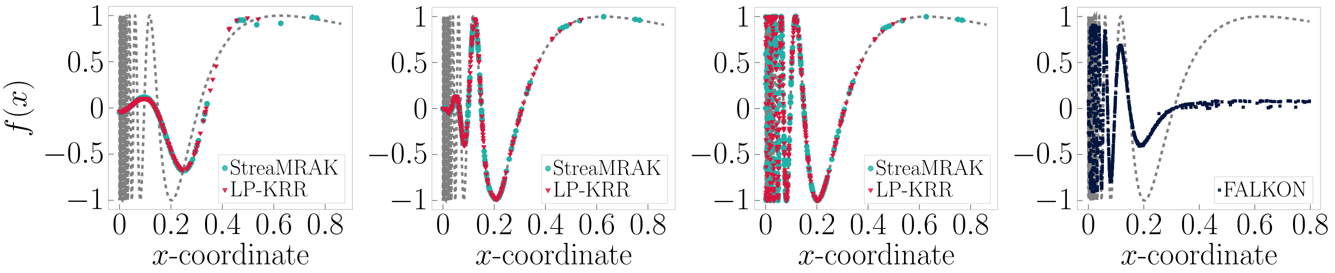
We also need to pre-select the number of training points for LP-KRR and FALKON. For FALKON we use the entire training set, as in [2]. Similarly, it is also common for the LP to use the entire training set at each level [27,28]. However, for large data sets, it might be better to include fewer data points. Therefore, we also use a version of the LP-KRR where we divide the total training data equally between the levels.

Throughout the experiments, we measure the performance of StreaMRAK, FALKON, and LP-KRR by estimating the mean square error

$$MSE(y, y_{pred}) = \frac{1}{\Upsilon\Lambda}\sum_{k=1}^{\Upsilon}\frac{1}{n_k}||\mathbf{y}_k - \mathbf{y}_k^{pred}||^2, \text{ with } \Lambda = \max_{\substack{k\in[\Upsilon]\\ i\in[n_k]}}[\mathbf{y}_k]_i - \min_{\substack{k\in[\Upsilon]\\ i\in[n_k]}}[\mathbf{y}_k]_i, \tag{6.1}$$

Table 1
Comparison of StreaMRAK, LP-KRR and FALKON for the target in Eq. (6.2). For each level l we show the number of landmarks, the mean square error (MSE), and the accumulated time to train the prediction model (Time). In parenthesis, in the time column of the FALKON row, is the time to find the optimal bandwidth through cross-validation.

	Level	# Landmarks	MSE	Time
	5	47	2.55×10^{-1}	77 s
StreaMRAK	10	392	3.69×10^{-2}	116 s
	15	1525	$8.63\times10-6$	497 s
	16	2302	$6.18\times10-6$	1194 s
	5	1483	2.56×10^{-1}	143 s
LP-KRR (1) $n_l = 1.1\times10^5$	10	1483	3.65×10^{-2}	413 s
	15	1483	$8.72\times10-6$	825 s
	16	1483	$6.85\times10-6$	922 s
	18	1483	$6.55\times10-6$	1136 s
	5	1483	2.56×10^{-1}	2963 s
LP-KRR (2) $n_l = 2.2\times10^6$	10	1483	3.64×10^{-2}	8704 s
	18	1483	$8.91\times10-6$	23113 s
StreaMRAK	–	14830	5.7×10^{-3}	4642 s+(27930 s)



(a) Level 3 (b) Level 5 (c) Level 16 (d) Single level

Fig. 3. (a)-(d) shows the target function $f(x)$ from Eq. (6.2) as a grey dotted line. The light-blue circles indicates the predicted values made by StreaMRAK. Similarly the red triangles indicates the predictions made by LP-KRR and the dark blue squares the predictions made by FALKON. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where Υ is the number of test runs we average over, n_k is the number of test points at test run k, and $\mathbf{y}_k, \mathbf{y}_k^{pred} \in \mathbb{R}^{n_t}$ are the target values and predictions respectively, and Λ is the normalisation factor.

6.1. Multi-resolution benchmark

We consider the function,

$$f(x) = \sin\left(\frac{1}{x+0.01}\right), \text{ for } x \in \left[0, \frac{\pi}{4}\right]. \tag{6.2}$$

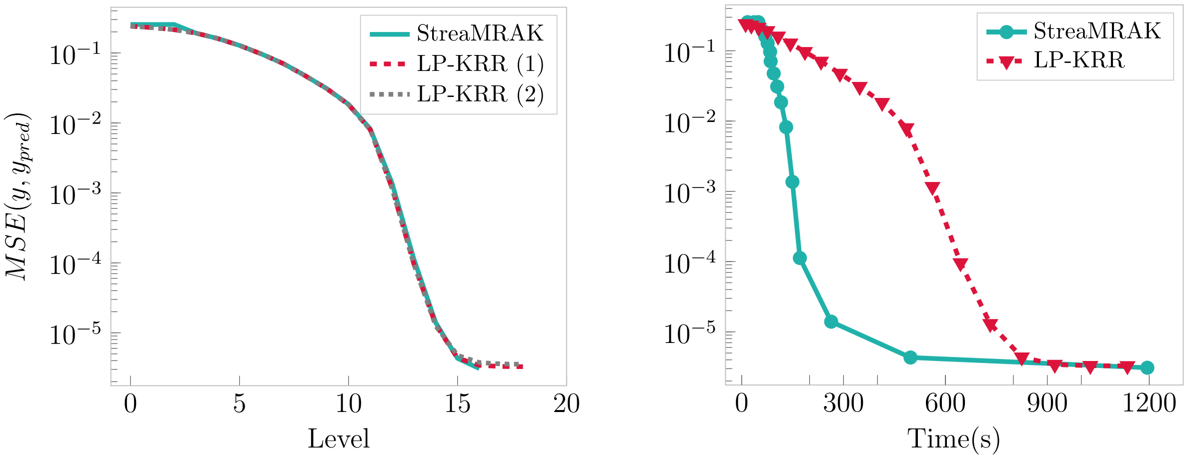
In the experiment we use a training set of $n = 2.2\times10^6$ samples and a test set of 1.3×10^5 samples. We use the non-uniform gamma distribution $\Gamma(\alpha,\beta)$ with $\alpha = 1$, $\beta = 2$ to sample the training data.

The number of training points used at each level in StreaMRAK is determined by setting δ_1 and δ_2 from Def. 4.1 to 10^{-3}. With this choice, StreaMRAK selects between 30244 and 40100 training points for each level. For comparison, FALKON uses all the 2.2×10^6 training points. Furthermore, for LP-KRR we run two experiments: LP-KRR (1) using 1.1×10^5 training points at each level and LP-KRR (2) using 2.2×10^6 training points at each level.

Results are presented in Table 1, and the prediction results are illustrated in Fig. 3a-3 d. The results show that StreaMRAK and both LP-KRR schemes perform much better than FALKON. The reason is that FALKON uses only one bandwidth r, while the multi-resolution schemes StreaMRAK and LP-KRR, utilize a bandwidth regime $r_l = 2^{-l}r_0$ that varies with the level l. The consequence is that StreaMRAK and LP-KRR approximate the low-frequency components of f when the bandwidth is large, and then target the high-frequency components of $f(x)$ gradually as the bandwidth decreases. These results illustrate the benefits of a multi-resolution scheme over a single bandwidth scheme.

From Table 1, we also observe that LP-KRR (2) is significantly slower than StreaMRAK and LP-KRR (1). This is because it uses the entire training set at each level. Therefore, since LP-KRR (1) and LP-KRR (2) achieve comparable precision, we see that including all training points at each level is not always necessary.

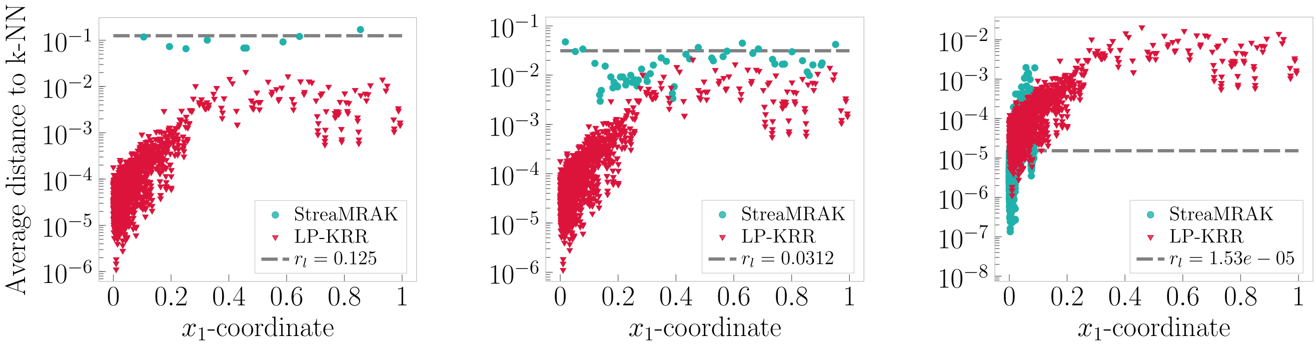
A closer comparison of StreaMRAK and LP-KRR is given in Fig. 4. In particular, in Fig. 4a we see that the two algorithms achieve very similar precision. However, comparing the training times in Fig. 4b, we see that StreaMRAK trains each level faster and therefore achieves better precision earlier than LP-KRR (1).



(a) Mean square error

(b) Training time

Fig. 4. Comparison of StreaMRAK and LP-KRR. (a) shows the mean square error calculated according to Eq. (6.1) with the target function from Eq. (6.2). Along the x-axis is the number of levels included in the model. (b) The x-axis shows the accumulated training time until a level in the LP is completed. The y-axis shows the MSE of the prediction using the currently available model. The blue circles indicate the prediction error of StreaMRAK and the red triangles indicate the prediction error of LP-KRR (1). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Level 3

(b) Level 5

(c) Level 16

Fig. 5. (a)-(c) shows the landmarks with their position along the x_1 axis and the average distance to their 2 nearest neighbors along the y-axis. Here the red triangles are the Nyström landmarks of LP-KRR and the light-blue circles the landmarks of StreaMRAK. The grey dotted line is the bandwidth at the given level. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In Fig. 5 we show the average distance of each landmark to their 2 nearest neighbors (2-NN distance). Two aspects of the selection require attention. As opposed to LP-KRR, StreaMRAK selects landmarks such that the 2-NN distance is comparable to the bandwidth used at a specific level. In addition, StreaMRAK saves computational power by not choosing landmarks in regions where the 2-NN distance is too low compared to the bandwidth. In Fig. 5c this can be observed for level $l = 16$ for landmarks with $x \geq 0.2$. Due to the non-uniform sample distribution with a higher density around $x = 0$, the adaptive sub-sampling is able to select more landmarks in the region close to $x = 0$, where f oscillates with high frequency. Furthermore, StreaMRAK stops predicting at level 16 because level 17 is not yet covered with a high enough density of landmarks. Meanwhile, LP-KRR continues, but as seen from Fig. 4a the improvements after level 15 are not significant because the density of Nyström samples is too low compared to the bandwidth.

6.2. Adaptive sub-sampling benchmark

We consider a dumbbell-shaped domain embedded in $\mathbb{R}^5$, consisting of two 5-dimensional spheres connected by a 2-dimensional plane. A projection of the input domain in $\mathbb{R}^3$ is shown in Fig. 7 (a)-(c). Furthermore, as target we consider the following function,

$$f(\mathbf{x}) = \begin{cases} A\sin(Bx_1 + \phi) + (x_1 + 2), & 1 < x_1 < 3 \\ 1, & \text{otherwise} \end{cases}, \quad \text{for } \mathbf{x} \in [-1, 5] \times [-1, 1]^4, \tag{6.3}$$

where A, B and ϕ are chosen so that $f \in \mathcal{C}^1([-1, 5] \times [-1, 1]^4, \mathbb{R}^5)$. For the experiments, we consider a training set of 1.9×10^6 samples and a test set of 6×10^5 samples, all sampled uniformly at random from the input domain. We note that

Table 2
Comparison of StreaMRAK, LP-KRR, and FALKON predictions of the target function in Eq. (6.3). For each level l we show the number of landmarks, the mean square error (MSE), and the accumulated time to train the prediction model (Time). In parenthesis, in the time column of the FALKON row, is the time to find the optimal bandwidth through cross-validation.

	Level	# Landmarks	MSE	Time
StreaMRAK	4	352	1.29×10^{-3}	64 s
	5	2667	1.27×10^{-3}	1398 s
	6	1858	8.31×10^{-4}	1462 s
	8	1329	2.75×10^{-5}	2307 s
LP-KRR (1) $n_l = 1.8 \times 10^5$	4	1375	1.28×10^{-3}	386 s
	5	1375	1.26×10^{-3}	520 s
	6	1375	9.10×10^{-4}	671 s
	8	1375	3.30×10^{-4}	1064 s
	9	1375	3.10×10^{-4}	1287 s
LP-KRR (2) $n_l = 1.9 \times 10^6$	4	1375	1.34×10^{-3}	4160 s
	5	1375	1.30×10^{-3}	5570 s
	6	1375	9.44×10^{-4}	7168 s
	8	1375	3.16×10^{-4}	11125 s
	9	1375	3.01×10^{-4}	13334 s
FALKON	–	14830	6.8×10^{-4}	6590 s+(37561 s)

we purposefully chose a simple function in the high dimensional regions because complicated functions in high dimensions require far too many points to be satisfactorily learned.

To determine the number of training points for StreaMRAK, we let $\delta_1 = 1 \times 10^{-3}$ and $\delta_2 = 1 \times 10^{-4}$, cf. Def. 4.1. With this choice StreaMRAK selects between 30100 and 40100 training points for each level. FALKON again uses all the 1.9×10^6 training points and for LP-KRR we consider two settings: LP-KRR (1) using 1.8×10^5 training points at each level, and LP-KRR (2) using 1.9×10^6 training points at each level.

The results for StreaMRAK, LP-KRR, and FALKON are presented in Table 2. We observe that StreaMRAK achieves a better prediction than both FALKON and LP-KRR because it adapts the sub-sampling density to the level of resolution.

To understand the improvement in prediction accuracy, we need to discuss the effects of landmark selection. In Fig. 7a-7 c we show the projections of landmarks for StreaMRAK and LP-KRR on $\mathbb{R}^3$, and in Fig. 7d-7 f the average distance of each landmark to its 7 nearest neighbors. These distances are compared with the bandwidth r_l selected for the given level l. We see that StreaMRAK selects landmarks in regions where the average distance to nearest neighbors is comparable to the bandwidth. This means that in high dimensional regions, which correspond to $x_1 \in [-1, 1] \cup [3, 5]$, the algorithm effectively stops collecting landmarks since it cannot maintain high enough density. On the other hand, LP-KRR uses Nystrom sub-sampling, which imposes a uniform selection of landmarks. Consequently, a significant number of landmarks come from high-dimensional regions.

Moreover, Fig. 7 shows that in the case of LP-KRR, the average distance between the landmarks in high dimensional regions is larger than the bandwidth r_l when $l \geq 5$. As a knock-on effect, LP-KRR makes only small improvements in high dimensional regions for $l \geq 5$, as seen from Fig. 6b. Analogous behavior can be observed for StreaMRAK. However, since StreaMRAK devotes fewer resources to high dimensional regions, it sub-samples more from the low dimensional region, as illustrated in Fig. 6a. The consequence is that StreaMRAK makes bigger improvements in the low dimensional region than LP-KRR, as seen from Fig. 6b. Note that this was not the case in Section 6.1, where the two methods had similar behavior, but unlike here, the input domain in Section 6.1 did not consist of regions with different dimensionalities.

6.3. Forecasting the trajectory of a double pendulum

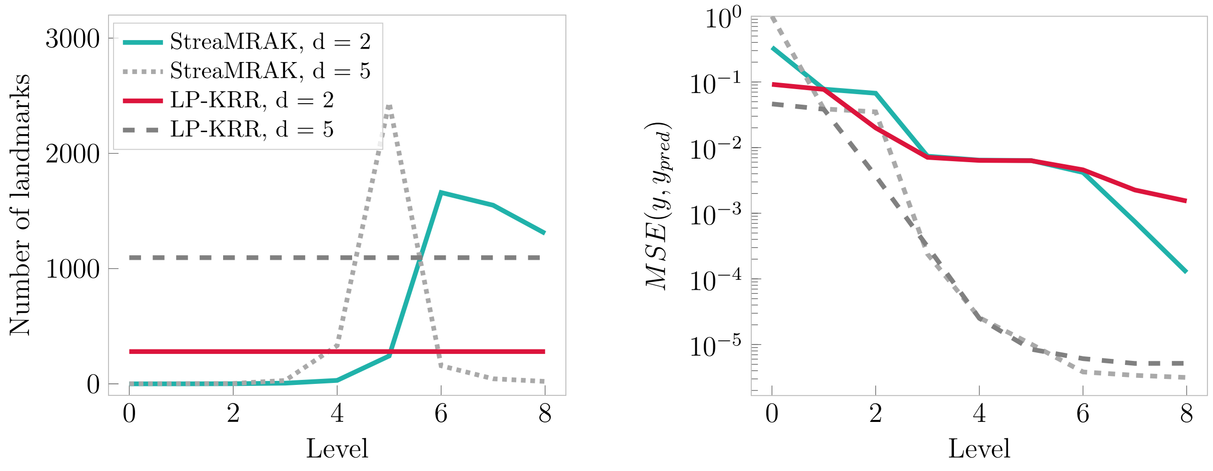
We consider the double pendulum, illustrated in Fig. 2a, which we model by the Lagrangian system

$$\mathcal{L} = ml^2(\omega_1^2 + \frac{1}{2}\omega_2^2) + ml^2\omega_1\omega_2 \cos(\theta_1 - \theta_2) + mgl(2\cos\theta_1 + \cos\theta_2), \tag{6.4}$$

under the assumption that the pendulums are massless rods of length $l_1 = l_2 = l$ with masses $m_1 = m_2 = m$ centered at the end of each rod. Here g is the standard gravity, $\omega_1 := \dot{\theta}_1$, $\omega_2 := \dot{\theta}_2$ are the angular velocities, and the angles θ_1, θ_2 are as indicated in Fig. 2a. For the experiments we let $m = 1$, $l = 1$ and $g = 10$.

The learning task is to forecast the trajectory of the pendulum, given only its initial conditions. We let $\mathbf{s}_t = [\theta_1(t), \theta_2(t), \omega_1(t), \omega_2(t)] \in \mathbb{R}^4$ be the state of the system at step $t \in \mathbb{N}$ and train StreaMRAK, LP-KRR and FALKON to learn how $\mathbf{s}_t$ maps to a later state $\mathbf{s}_{t+\Delta}$, for $\Delta \in \mathbb{N}$. The trained model $\widehat{f}$ is used to forecast the state $\mathbf{s}_T$ for $T >> 0$ by recursively predicting $\mathbf{s}_{t+\Delta} = \widehat{f}(\mathbf{s}_t)$ from the initial state $\mathbf{s}_0$ until $t = T$.

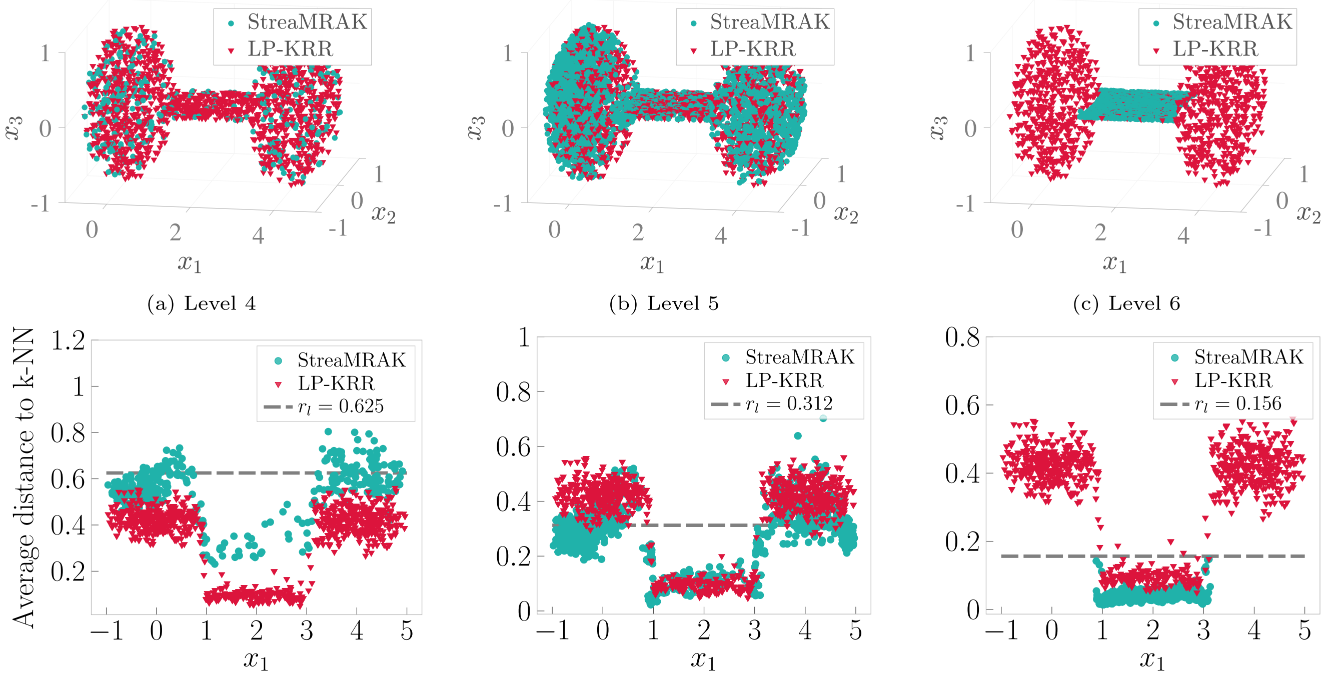
For the experiments we consider two settings: a low energy system $\mathbf{s}_0^{low} = [-20^\circ, -20^\circ, 0^\circ, 0^\circ]$ and a high energy system $\mathbf{s}_0^{high} = [-120^\circ, -20^\circ, -7.57^\circ, 7.68^\circ]$. For these systems, we initialize 8000 pendulums as $\mathbf{s}_0 \sim \mathcal{N}(\mathbf{s}, \sigma(\mathbf{s}))$ for $\mathbf{s} = \mathbf{s}_0^{low}, \mathbf{s}_0^{high}$



(a) Landmarks distribution

(b) Mean square error

Fig. 6. Comparison of StreaMRAK and LP-KRR (1) in the 2-dim and 5-dim regions of the Dumbbell domain. The solid blue line is StreaMRAK for dimension $d = 2$ while the solid red line is LP-KRR (1) for dimension $d = 2$. The grey dotted line is StreaMRAK for dimension $d = 5$ and the dark-grey dashed line is LP-KRR (1) for dimension $d = 5$ (a) shows the mean square error calculated according to Eq. (6.1). (b) shows the number of landmarks in the 2-dimensional and the 5- dimensional regions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Level 4

(b) Level 5

(c) Level 6

(d) Level 4

(e) Level 5

(f) Level 6

Fig. 7. In the figure, red triangles correspond to LP-KRR and light-blue circles to StreaMRAK. (a)-(c) shows the landmark distributions projected on $\mathbb{R}^3$ at level $l = 4, 5, 6$ respectively. (d)-(f) shows the average distance between the 7 nearest neighbors; bandwidth r_l is indicated with a dotted line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

respectively, where $\sigma(\mathbf{s}) = [0.025|\theta_1|,\ 0.15|\theta_2|,\ 0.3|\omega_1|,\ 0.3|\omega_2|]$. Each pendulum is iterated for 500 steps, which results in 5×10^6 training points distributed in $\mathbb{R}^4$. Furthermore, for the test data we consider 100 pendulums $\mathbf{s}_0 \sim \mathcal{N}(\mathbf{s}, 0.01|\mathbf{s}|)$ for $\mathbf{s} = \mathbf{s}_0^{low},\ \mathbf{s}_0^{high}$, iterated for 500 steps.

To determine the number of training points for StreaMRAK, we let $\delta_1, \delta_2 = 10^{-4}$, cf. Def. 4.1. With this choice StreaMRAK selects between 30219 and 70282 training points for each level for the low energy system, and between 36300 and 130200 for the high energy system. Meanwhile, FALKON uses all 5.0×10^6 training points and LP-KRR use 3.9×10^5 training points at each level.

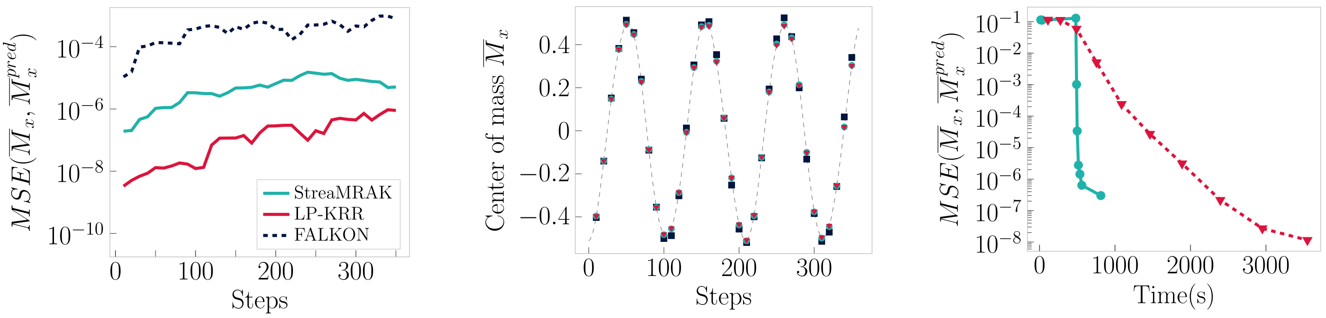
Results are presented in Table 3 and Table 4. Furthermore, to illustrate the prediction results we consider the center of mass $\overline{M}_x(\mathbf{s}_t) = \frac{1}{2}(x_1(t) + x_2(t)) \in \mathbb{R}$ at state $\mathbf{s}_t$, where $x_1, x_2 \in \mathbb{R}$ are the positions of the two pendulum masses as seen in

Table 3
Comparison of StreaMRAK, LP-KRR, and FALKON for the low energy system. For each level l we show the number of landmarks, the MSE at step T=50, and the accumulated time to train the prediction model (Time). In parenthesis, in the time column of the FALKON row, is the time to find the optimal bandwidth through cross-validation.

	Level	# Landmarks	MSE(T=50)	Time
StreaMRAK	2	1	1.12×10^{-1}	31 s
	5	66	3.39×10^{-5}	498 s
	7	659	$1.44\times10-6$	534 s
	9	4085	3.01×10^{-7}	812 s
LP-KRR	2	1979	5.93×10^{-2}	490 s
	5	1979	2.73×10^{-5}	1463 s
	7	1979	2.16×10^{-7}	2395 s
	9	1979	1.16×10^{-8}	3550 s
FALKON	–	19790	5×10^{6}	2934 s+(1498 s)

Table 4
Comparison of the StreaMRAK, LP-KRR, and FALKON for the high energy system. For each level l we show the number of landmarks, the MSE at step T=50, and the accumulated time to train the prediction model (Time). In parenthesis, in the time column of the FALKON row, is the time to find the optimal bandwidth through cross-validation.

	Level	# Landmarks	MSE(T=50)	Time
StreaMRAK	2	1	2.70×10^{-1}	49 s
	5	1106	8.53×10^{-3}	915 s
	7	6376	2.16×10^{-4}	1999 s
LP-KRR	2	1979	1.72×10^{-2}	522 s
	5	1979	5.09×10^{-3}	1474 s
	7	1979	1.39×10^{-4}	2431 s
FALKON	–	19790	5×10^{6}	23830 s+(11050 s)



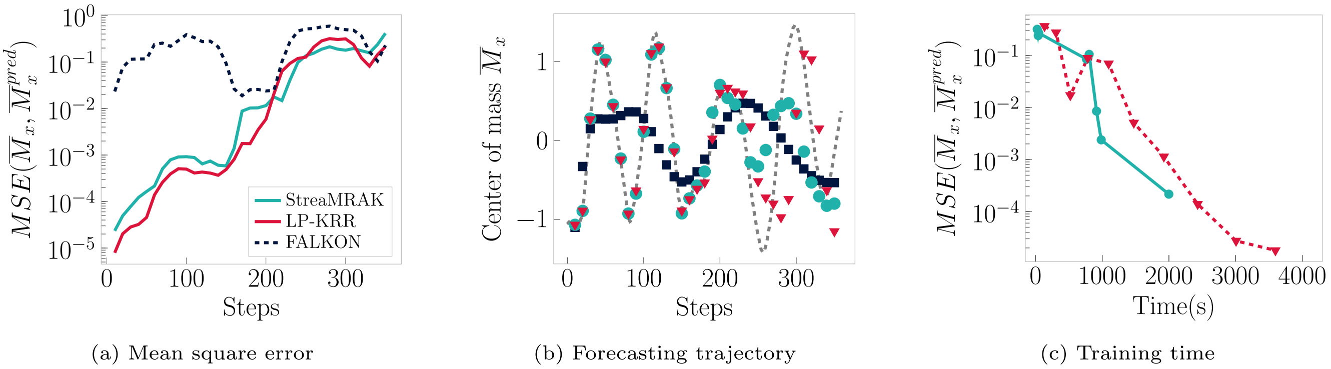
(a) Mean square error (b) Forecasting trajectory (c) Training time

Fig. 8. Comparison of StreaMRAK (light blue lines and circles), LP-KRR (red lines and triangles), and FALKON (dark blue dotted lines and squares) for the low energy pendulum. (a) Shows the mean square error of the center of mass $\overline{M}_x(\mathbf{s}_t)$ for the level 7 prediction, with step T along the x-axis. (b) shows the true center of mass trajectory as a grey dotted line and the predictions of StreaMRAK, LP-KRR, and FALKON at level 9. (c) The x-axis shows the accumulated training time until a level in the LP is completed. The y-axis shows the MSE of the predicted system state after $T = 50$ steps. We note that StreaMRAK includes 7 levels, while LP-KRR includes 9. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 2a. The prediction results are illustrated in Fig. 8 and Fig 9 for the low and high energy pendulums respectively. We calculate the MSE at each step t separately, such that for a given t we use Eq. 6.1 with $\mathbf{y}_k = \overline{M}_x(\mathbf{s}_t)$, $\mathbf{y}_k^{pred} = \overline{M}_x(\mathbf{s}_t^{pred})$ and $\Upsilon = 100$.

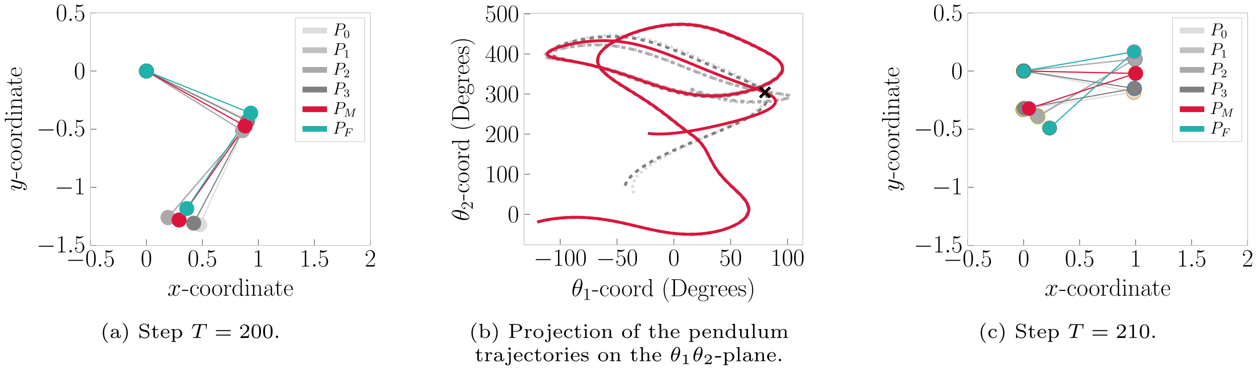
For the low energy system, we see from Fig. 8c how StreaMRAK is trained significantly faster than LP-KRR, although at a cost of reduced precision. The reduced training time of StreaMRAK is a consequence of the low doubling dimension of the training data, which allows the selection of far fewer landmarks for StreaMRAK than what is used at each level in LP-KRR.

For the high-energy pendulum, we see from Fig. 9c that StreaMRAK is again able to achieve good precision faster than LP-KRR. Furthermore, we see that the number of landmarks selected for StreaMRAK increases abruptly with the levels, reflecting the high doubling dimension of the training data. Due to this StreaMRAK stops the training after level 7, as



(a) Mean square error (b) Forecasting trajectory (c) Training time

Fig. 9. Comparison of StreaMRAK (light blue lines and circles), LP-KRR (red lines and triangles), and FALKON (dark blue dotted lines and squares) for the high-energy pendulum. (a) Shows the mean square error of the center of mass $\overline{M}_x(\mathbf{s}_t)$ for the level 7 prediction, with step T along the x-axis. (b) shows the true center of mass trajectory as a grey dotted line and the predictions of StreaMRAK, LP-KRR, and FALKON at level 9. (c) The x-axis shows the accumulated training time until a level in the LP is completed. The y-axis shows the MSE of the predicted system state after $T = 50$ steps. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Step $T = 200$. (b) Projection of the pendulum trajectories on the $\theta_1\theta_2$-plane. (c) Step $T = 210$.

Fig. 10. (a) Pendulum positions at $T = 200$ and (c) The positions at $T = 210$. In (a) and (c), P_M is the main pendulum with initial conditions $\mathbf{s}_0^{high}$, while P_F is the StreaMRAK forecast of the pendulum position. Similarly, $P_0 - P_3$ are four training pendulums with a perturbation of 0.5% on the initial angles θ_1 and θ_2 of the main pendulum. (b) Projection of the training data on the $\theta_1\theta_2$-plane. The thick red line is the main pendulum corresponding to P_M and the four grey dotted lines are the test pendulums $P_0 - P_3$, where the X indicates the time $T = 205$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the next levels require too many landmarks. By continuing for 2 more levels LP-KRR is able to achieve marginally better precision but at increased computational cost.

As seen in Fig. 9b, the forecasting of StreaMRAK and LP-KRR breaks down after $T \approx 200$ steps. In Fig. 10b we observe the trajectory of a pendulum with initial condition $\mathbf{s}_0^{high}$, as well as four pendulums with a 0.5% perturbation on the angles θ_1 and θ_2 in $\mathbf{s}_0^{high}$. We observe that after roughly $T = 205$ time steps the trajectory of the five pendulums diverge significantly from each other. Therefore, it seems that a bifurcation point occurs around this time, which may explain why all the algorithms are unable to make good forecasting beyond this point.

7. Outlook

Further development of StreaMRAK is intended with focus on four objectives.

(O1) Augmentation of the DCT to track the error at each node
(O2) Improve the estimator in Def. 4.1 and Eq. 3.1.
(O3) Refinement of previously fitted levels in the LP as new data arrives.
(O4) Further theoretical analysis of the LP.

Considering objective (O1) we intend to develop the DCT to track the error at each node. This way the growth can be restricted in regions where the error is small, which allows for more focus on regions where the error is large. The intention is that this will reduce the problem complexity even further, while also increasing the precision. Regarding objective (O2), a drawback with the estimator in Eq. 3.1 was already mentioned in Remark B.1 in Appendix B. Furthermore, for the estimator in Def. 4.1, we intend to implement and evaluate alternative ways to estimate the convergence of the matrices. Another focus area will be objective (O3), as we believe new information may be revealed as new training data arrive, and refinement of previously fitted levels can therefore be beneficial. Finally, the theoretical analysis in objective (O4) will focus on analyzing the generalization error for the LP, particularly in combination with the adaptive sub-sampling scheme.

Acknowledgement

We especially would like to thank Prof. Pieter Abeel at UC Berkeley and Asst. Prof. Sicun Gao at UC San Diego for their input on the double pendulum system, and for providing a code example for this system. We would also like to thank Sami Ortoleva at UC San Diego for his discussion on the analysis of the damped cover-tree. AO is part of the Simula-UCSD-UiO Research and Ph.D. training program (SUURPh), an international collaboration in computational biology and medicine funded by the Norwegian Ministry of Education and Research, ŽK is funded by UK EPSRC grant EP/T000864/1, AC is funded by NSF DMS 1819222, 2012266, and Russell Sage Foundation grant 2196 and YF is funded by the NIH grant NINDS (PHS) U19NS107466 Reverse Engineering the Brain Stem Circuits that Govern Exploratory Behavior.

Appendix A. Algorithms

We here denote nodes by p, q, c and $\mathbf{x}_p, \mathbf{x}_q, \mathbf{x}_c \in \mathcal{X} \subset \mathbb{R}^D$ are the corresponding points.

Algorithm A.1 INSERT(point q, node p, level l).

```
We assume $q$ already satisfies $\|\mathbf{x}_q - \mathbf{x}_p\| \le 2^{-l} r_0$.
if $\|\mathbf{x}_q - \mathbf{x}_c\| > 2^{-(l+1)} r_0$ for all $c \in Children(p)$ then
    Insert $q$ into $Children(c)$.
    UPDATE_COVERFRACTION($Parent(Q_l)$, “No parent found”)
    Break
else if $\|\mathbf{x}_q - \mathbf{x}_c\| < 2^{-(l+1)} r_0$ for some $c \in Children(p)$ then
    Consider all children of $c$, namely $Children(c)$
    if $Children(c)$ is empty then
        if Covering fraction of $p$, Def. 3.1, satisfy $\mathfrak{cf}(p) \ge \mathcal{D}_{\mathfrak{cf}}$ for some threshold $\mathcal{D}_{\mathfrak{cf}}$ then
            Insert q into $Children(c)$
            Break
        else
            UPDATE_COVERFRACTION(p, “parent found”){c is found to be a potential parent. However, since $\mathfrak{cf}(p) < \mathcal{D}_{\mathfrak{cf}}$ we can not add $q$ to $Children(c)$}
        end if
    else
        INSERT($q$, $c$, $l+1$)
    end if
end if
```

Algorithm A.2 STREAMRAK(point $\mathbf{x}$, target y).

```
Let $l$ be the level. Let $p_{(0)}$ be the root node, $r_0$ the radius of the root node.
Sub-sampling thread
Insert $\mathbf{x}$ into the cover tree with INSERT($\mathbf{x}$, $p_{(0)}$, $l = 0$). {See Alg. A.1}
if a new level has $\mathfrak{cf}(Q_l) \ge \mathcal{D}_{level}$. then
    Extract the landmarks at level $l$ as sub-samples, namely $\Gamma^{(l)}_{m^{(l)}}$.
end if
Training thread
Consider level $l$ and assume that the landmarks $\Gamma^{(l)}_{m^{(l)}}$ are extracted.
while $l$ is not sufficiently covered with training points according to Def. 4.1. do
    Update $\left[(\mathbf{K}^{(l)}_{nm})^\top \mathbf{K}^{(l)}_{nm}\right]_{ij}$ and $\mathbf{z}^{(l)}_i$ according to Eq. 2.4 and Eq. 2.5 as new samples $(\mathbf{x}, y)$ arrive, using the landmarks in $\widetilde{\Gamma}^{(l)}_m$ from Def. 3.3.
    Continuously check if matrices have converged.
    if Matrices converge according to Def. 4.1 then
        Update the StreaMRAK regression model $\widetilde{f}^{(L)}$, by including the correction term $s^{(l)}$ into the Laplacian pyramid, as described in Section 2.2. Let $L = l$ and update $l = l + 1$.
    end if
end while
```

Algorithm A.3 UPDATE_COVERFRACTION(node p, string s).

```
if s="No parent found" then
    Update covering fraction of $p$ with $\mathfrak{cf}(p) = (1-\alpha)\mathfrak{cf}(p)$
else if s= "parent found" then
    Update covering fraction of $p$ with $\mathfrak{cf}(p) = (1-\alpha)\mathfrak{cf}(p) + \alpha$
end if
```

Appendix B. Preparatory material

We offer preparatory material on the damped cover-tree and kernel methods.

B1. Preparatory material on the damped cover-tree

This section shows how the recursive formula in Eq. 3.1 approximates the weighted average of the outcome of the last N random trails. Where the trails are as described in Section 3.1. By expanding Eq. 3.1 we have $(\mathfrak{cf}(p))_t = (1-\alpha)^t(\mathfrak{cf}(p))_1 + \alpha\sum_{i=1}^{t-1}(1-\alpha)^i \mathbb{1}_{\mathcal{B}_c}(\mathbf{x}_{t-i})$. Since $(1-\frac{1}{N})^N \approx 1/e$, the first term becomes negligible when $t \gg N$. Similarly, all terms $i > N$ in the sum becomes negligible. This leaves,

$$(\mathfrak{cf}(p))_t \approx \frac{1}{N}\sum_{i=1}^{N}\left(1-\frac{1}{N}\right)^i \mathbb{1}_{\mathcal{B}_c}(\mathbf{x}_{t-i})$$

which is a weighted average of the outcome of the N last draws as claimed.

Remark B.1. We mention a weakness of the estimator in Eq. (3.1). As follows from Algorithm A.1, every time a new point $\mathbf{x}$ is not covered by the existing children, a new child is added. This consequently updates $\mathcal{B}_c$, leading to the posterior distribution $\mathrm{Prob}(\mathbb{1}_{\mathcal{B}_c}(\mathbf{x}) = 0|\mathbf{x})$ to changed every time $\mathbb{1}_{\mathcal{B}_c}(\mathbf{x}) = 0$.

B2. Preparatory material on Kernel methods

Kernel methods in the context of reproducing kernel Hilbert spaces (RKHS) offer a powerful approach to machine learning with a well-established mathematical foundation [1,62]. In this paper we consider an input space $\mathcal{X} \subset \mathbb{R}^D$, a corresponding target space $\mathcal{Y} \subset \mathbb{R}$ and let ρ be the probability distribution on $\mathcal{X} \times \mathcal{Y}$. Furthermore, we assume an RKHS $\mathcal{H}_k$ generated by a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. In other words, the eigenvalues $\sigma_i, \ldots, \sigma_n$ of the corresponding kernel matrix $\mathbf{K}_{nn} = (k(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$ satisfies $\sigma_i > 0$ for all $i \in n$. In this setting the inner product between two feature vectors $\phi(\mathbf{x}), \phi(\mathbf{x}') \in \mathcal{H}_k$ satisfies the property that $<\phi(\mathbf{x}), \phi(\mathbf{x}')>_{\mathcal{H}_k} = k(\mathbf{x}, \mathbf{x}')$. This relation, known as the "kernel trick" [63,64], effectively circumvents the need for explicit construction of non-linear mappings ϕ.

Given a training set $\{(\mathbf{x}_i, y_i) : i \in [n]\}$ sampled according to ρ with $\Gamma_n = \{\mathbf{x}_i : i \in [n]\}$, we formulate the kernel ridge regression (KRR) problem as

$$\widehat{f}_{n,\lambda} = \underset{f \in \widehat{\mathcal{H}}_n}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i) - y_i)^2 + \lambda\|f\|_{\mathcal{H}}^2, \tag{B.1}$$

where $\lambda > 0$ is a regularisation parameter and $\widehat{\mathcal{H}}_n = \overline{\mathrm{span}}\{k(\cdot, \mathbf{x}_i) : i \in [n]\}$ is a finite-dimensional subspace of $\mathcal{H}_k$. What is more, for all $f \in \widehat{\mathcal{H}}_n$ the Representer theorem [65,66] guarantees that there exists coefficients $\alpha_1, \ldots, \alpha_n$ such that the solution to Eq. (B.1) is on the form

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

Computing the KRR estimator is therefore reduced to solving the linear system

$$(\mathbf{K}_{nn} + \lambda\mathbf{I}_n)\boldsymbol{\alpha} = \mathbf{y},$$

where $\mathbf{y} = (y_1, \ldots, y_n)^\top$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top$, and $[\mathbf{K}_{nn}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Appendix C. Auxiliary results

Lemma C.1. *Let $\mathbf{d}^{(l)}$ be the residual at level l as defined in Eq. (2.9). We then have,*

$$\mathbf{d}^{(l+1)} = (\mathbf{I} - \mathbf{K}_{nn}^{(l)}(\mathbf{K}_{nn}^{(l)} + \lambda n\mathbf{I})^{-1})\mathbf{d}^{(l)}$$

Proof. Denote $\mathbf{s}^{(l)} = s^{(l)}([\mathbf{x}_n])$, and note that $\mathbf{s}^{(l)} = \mathbf{K}_{nn}^{(l)}(\mathbf{K}_{nn}^{(l)} + \lambda n\mathbf{I})^{-1}\mathbf{d}^{(l)}$. For $l = 1$, we have

$$\mathbf{d}^{(1)} = \mathbf{y} - \mathbf{s}^{(0)} = \mathbf{y} - \mathbf{K}_{nn}^{(l)}\boldsymbol{\alpha}^{(0)} = (\mathbf{I} - \mathbf{K}_{nn}^{(0)})(\mathbf{K}_{nn}^{(0)} + \lambda n\mathbf{I})^{-1}\mathbf{y}.$$

We proceed by induction. Assume the statement holds for an $l \geq 2$. We now have

$$\mathbf{d}^{(l+1)} = \mathbf{y} - \sum_{j=0}^{l}\mathbf{s}^{(j)} = \mathbf{d}^{(l)} - \mathbf{s}^{(l)} = \mathbf{d}^{(l)} - \mathbf{K}_{nn}^{(l)}(\mathbf{K}_{nn}^{(l)} + \lambda n\mathbf{I})^{-1}\mathbf{d}^{(l)} = (\mathbf{I} - \mathbf{K}_{nn}^{(l)}(\mathbf{K}_{nn}^{(l)} + \lambda n\mathbf{I})^{-1})\mathbf{d}^{(l)}.$$

□

Remark C.2. In [60, Thm. 12.3] they also offer an a fortiori bound corresponding to $M_D = 6.38D$, $C_{1,D} = \frac{1}{2}\left(\frac{12.76}{2^{3/2}}\right)^D\left(\frac{D^D}{\Gamma(D/2+1)}\right)\left(\frac{r_0}{\delta}\right)^D$ and $C_{2,D} = (12.76\sqrt{2}D)^2(r_0/\delta)^2$.

Corollary C.3. *We note that $B(l)$ from Eq. (5.8) has a maximum at*

$$l^* \quad = \frac{1}{2}\log_2\left(\frac{C_{2,D}\log 4}{D\log 2}\right) = \log_2\left(\sqrt{\frac{D}{2}}\left(\frac{r_0}{\delta}\right)\right) + \log_2\left(\frac{4M_D}{D}\sqrt{2}\right)$$

and is monotonically increasing with l on the interval $l \in (0, l^*)$. Furthermore, with the a fortiori expression for M_D from Remark C.2 we have

$$l^* = \log_2\left(\sqrt{\frac{D}{2}}\left(\frac{r_0}{\delta}\right)\right) + \log_2\left(25.52\sqrt{2}\right).$$

Lemma C.4. *The function*

$$F(T) := 2^{-\frac{T}{2}(1+1/\ln 2)}2^{-lT}T^{\frac{T}{2}}\left(\frac{r_0}{\delta}\right)^T$$

has minimum

$$F(T^*) = 2^{-\frac{1}{\ln 2}4^{l-\log_2(r_0/\delta)}}.$$

Proof. We can write $F(T)$ as

$$F(T) = 2^{-\frac{T}{2}(1+1/\ln 2)}2^{-lT}2^{T/2\log_2 T}2^{T\log_2(r_0/\delta)} = 2^{-T/2(B-\log_2 T)} = 2^{f(T)},$$

where $B = 1 + \frac{1}{\ln 2} + 2l - 2\log_2\left(\frac{r_0}{\delta}\right)$ and $f(T) = -T/2(B - \log_2 T)$. $F(T)$ is therefore minimized when $f(T)$ is minimized. Namely when

$$T^* = 2^{B-1/\ln 2} = 2^{1+1/\ln 2+2l-2\log_2(r_0/\delta)-1/\ln 2} = 2\cdot 4^{l-\log_2(r_0/\delta)}.$$

Inserting this back into the expression for $F(T)$ gives the desired result. □

References

[1] B. Schölkopf, A.J. Smola, Learning with kernels: Support vector machines, regularization, optimization, and beyond, 1st, MIT press, 2002.

[2] A. Rudi, L. Carratino, L. Rosasco, FALKON: An optimal large scale kernel method, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Proc. 31th Int. Conf. Neural Inf. Process. Syst., volume 30, 2017, pp. 3889–3899.

[3] A.E. Alaoui, M.W. Mahoney, Fast randomized kernel ridge regression with statistical guarantees, in: Proc. 28th Int. Conf. Neural Inf. Process. Syst., volume 1, 2015, pp. 775–783.

[4] Y. Zhang, J. Duchi, M. Wainwright, Divide and conquer kernel ridge regression: adistributed algorithm with minimax optimal rates, J. Mach. Learn. Res. 16 (2015) 3299–3340.

[5] H. Avron, K.L. Clarkson, D.P. Woodruff, Faster kernel ridge regression using sketching and preconditioning, J. Matrix. Anal. Appl. 38 (4) (2017) 1116–1138, doi:10.1137/16M1105396.

[6] E. Burnaev, I. Nazarov, Conformalized kernel ridge regression, in: Proc. 15th Int. Conf. Mach. Learn. Appl., 2017, pp. 45–52, doi:10.1109/ICMLA.2016.65.

[7] P. Exterkate, P.J. Groenen, C. Heij, D. van Dijk, Nonlinear forecasting with many predictors using kernel ridge regression, Int. J. Forecas 32 (3) (2016) 736–753, doi:10.1016/j.ijforecast.2015.11.017.

[8] M. Niu, S. Rogers, M. Filippone, D. Husmeier, Fast parameter inference in nonlinear dynamical systems using iterative gradient matching, in: Proc. 33rd Int. Conf. Mach. Learn. Res., 2016, pp. 1699–1707.

[9] M. Stock, T. Pahikkala, A. Airola, B. De Baets, W. Waegeman, A comparative study of pairwise learning methods based on kernel ridge regression, Neural Comput. 30 (8) (2018) 2245–2283, doi:10.1162/neco_a_01096.

[10] S. An, W. Liu, S. Venkatesh, Face recognition using kernel ridge regression, in: Proc. Conf. Comput. Vis. Recognit., 2007, pp. 1–7, doi:10.1109/CVPR.2007.383105.

[11] B.Y.S. Li, L.F. Yeung, K.T. Ko, Indefinite kernel ridge regression and its application on QSAR modelling, Neurocomputing 158 (2015) 127–133, doi:10.1016/j.neucom.2015.01.060.

[12] P. Mohapatra, S. Chakravarty, P.K. Dash, Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system, Swarm. Evol. Comput. 28 (2016) 144–160, doi:10.1016/j.swevo.2016.02.002.

[13] S. Muthukrishnan, Data streams: algorithms and applications, Found. Trends Theor. Comput. Sci. 1 (2) (2005) 117–236, doi:10.1561/0400000002.

[14] W. Fan, A. Bifet, Mining big data, ACM SIGKDD Explor. Newsl. 14 (2) (2013) 1–5, doi:10.1145/2481244.2481246.

[15] K. Lan, D.-T. Wang, S. Fong, L.-S. Liu, K.K.L. Wong, N. Dey, A survey of data mining and deep learning in bioinformatics, J. Med. Syst. 42 (8) (2018), doi:10.1007/s10916-018-1003-9.

[16] J. Kivinen, A.J. Smola, R.C. Williamson, Online learning with kernels, in: Proc. 14th Int. Conf. Neural Inf. Process. Syst., 2001, pp. 785–792, doi:10.7551/mitpress/1120.003.0105.

[17] C. Scovel, D. Hush, I. Steinwart, J. Theiler, Radial kernels and their reproducing kernel hilbert spaces, J. Complex. 26 (6) (2010) 641–660, doi:10.1016/j.jco.2010.03.002.
[18] C.A. Micchelli, Y. Xu, H. Zhang, Universal kernels, J. Mach. Learn. Res. 7 (2006) 2651–2667.
[19] Z. Wang, K. Crammer, S. Vucetic, Breaking the curse of kernelization: budgeted stochastic gradient descent for large-scale SVM training, J. Mach. Learn. Res. 13 (2012) 3103–3131.
[20] C.R. Loader, Bandwidth selection: classical or plug-in? Ann. Stat. 27 (2) (1999) 415–438.
[21] G.C. Cawley, N.L. Talbot, Fast exact leave-one-out cross-validation of sparse least-squares support vector machines, Neural Netw. 17 (10) (2004) 1467–1475, doi:10.1016/j.neunet.2004.07.002.
[22] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Stat. Surv. 4 (2010) 40–79, doi:10.1214/09-SS054.
[23] R. Krauthgamer, J.R. Lee, Navigating nets: Simple algorithms for proximity search, in: Proc. 15th Annu. ACM-SIAM Symp. Discrete Algorithms, 2004, pp. 798–807.
[24] A. Beygelzimer, S. Kakade, J. Langford, Cover trees for nearest neighbor, in: Proc. 23th Int. Conf. Mach. Learn., 2006, pp. 97–104, doi:10.1145/1143844.1143857.
[25] T. Shinbrot, C. Grebogi, J. Wisdom, J.A. Yorke, Chaos in a double pendulum, Am. J. Phys. 60 (6) (2016) 491–499, doi:10.1119/1.16860.
[26] A. Marcelo Tusset, V. Piccirillo, A.M. Bueno, J. Manoel Balthazar, D. Sado, J.L.P. Felix, R.M.L. Brasil, Chaos control and sensitivity analysis of a double pendulum arm excited by an RLC circuit based nonlinear shaker, J. Vib. Control 22 (17) (2016) 3621–3637, doi:10.1177/1077546314564782.
[27] N. Rabin, R.R. Coifman, Heterogeneous datasets representation and learning using diffusion maps and Laplacian pyramids, in: Proc. 12th Int. Conf. Data Min., 2012, pp. 189–199, doi:10.1137/1.9781611972825.17.
[28] W. Leeb, Properties of Laplacian pyramids for extension and denoising, 2019, arXiv:1909.07974
[29] P.J. Burt, E.H. Adelson, The laplacian pyramid as a compact image code, IEEE Trans. commun. 31 (4) (1983) 532–540.
[30] G.R. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, M.I. Jordan, Learning the kernel matrix with semidefinite programming, J. Mach. Learn. Res. 5 (2004) 27–72.
[31] F.R. Bach, G.R. Lanckriet, M.I. Jordan, Multiple kernel learning, conic duality, and the SMO algorithm, in: Proc. 21th Int. Conf. Mach. Learn., 2004, pp. 41–48, doi:10.1145/1015330.1015424.
[32] S. Sonnenburg, G. Rätsch, C. Schäfer, B. Schölkopf, Large scale multiple kernel learning, J. Mach. Learn. Res. 7 (2006) 1531–1565.
[33] E.G. Băzăvan, F. Li, C. Sminchisescu, Fourier kernel learning, in: Eur. Conf. Comput. Vis., 2012, pp. 459–473, doi:10.1007/978-3-642-33709-3_33.
[34] A. Bermanis, A. Averbuch, R.R. Coifman, Multiscale data sampling and function extension, Appl. Comput. Harmon. Anal. 34 (1) (2013) 15–29, doi:10.1016/j.acha.2012.03.002.
[35] N. Rabin, D. Fishelov, Multi-scale kernels for nystrŋm based extension schemes, Appl. Math. Comput. 319 (2018) 165–177, doi:10.1016/j.amc.2017.02.025.
[36] W. Liao, M. Maggioni, S. Vigogna, Multiscale regression on unknown manifolds, Mathematics in Engineering 4 (4) (2022) 1–25, doi:10.3934/mine.2022028.
[37] H. Fan, Q. Song, S.B. Shrestha, Kernel online learning with adaptive kernel width, Neurocomputing 175 (2015) 233–242, doi:10.1016/j.neucom.2015.10.055.
[38] B. Chen, J. Liang, N. Zheng, J.C. Príncipe, Kernel least mean square with adaptive kernel size, Neurocomputing 191 (2016) 95–106, doi:10.1016/j.neucom.2016.01.004.
[39] J. Zhang, H. Ning, X. Jing, T. Tian, Online kernel learning with adaptive bandwidth by optimal control approach, IEEE Trans. Neural Netw. Learn. Syst. 32 (5) (2021) 1920–1934, doi:10.1109/TNNLS.2020.2995482.
[40] A. Graps, An introduction to wavelets, IEEE Comput. Sci. Eng. 2 (2) (1995) 50–61, doi:10.1109/99.388960.
[41] A.N. Akansu, W.A. Serdijn, I.W. Selesnick, Emerging applications of wavelets: a review, Phys. Commun. 3 (1) (2010) 1–18, doi:10.1016/j.phycom.2009.07.001.
[42] R.R. Coifman, M. Maggioni, Diffusion wavelets, Appl. Comput. Harmon. Anal. 21 (1) (2006) 53–94, doi:10.1016/j.acha.2006.04.004.
[43] M. Maggioni, H.N. Mhaskar, Diffusion polynomial frames on metric measure spaces, Appl. Comput. Harmon. Anal. 24 (3) (2008) 329–353, doi:10.1016/j.acha.2007.07.001.
[44] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Appl. Comput. Harmon. Anal. 30 (2) (2011) 129–150, doi:10.1016/j.acha.2010.04.005.
[45] A. Cloninger, H. Li, N. Saito, Natural graph wavelet packet dictionaries, J. Fourier Anal. Appl. 27 (3) (2021) 1–33, doi:10.1007/s00041-021-09832-3.
[46] E. De Vito, Z. Kereta, V. Naumova, L. Rosasco, S. Vigogna, Wavelet frames generated by a reproducing kernel, J. Fourier Anal. Appl. 27 (2) (2021) 1–39, doi:10.1007/s00041-021-09835-0.
[47] A. Rahimi, B. Recht, Random features for large-scale kernel machines, in: Adv. Neural Inf. Process Syst., volume 20, 2008, pp. 1177–1184.
[48] Q.V. Le, T. Sarlos, A. Smola, Fastfood-computing hilbert space expansions in loglinear time, in: Proc. 30th Int. Conf. Mach. Learn., volume 28, 2013, pp. 244–252.
[49] Z. Yang, A.J. Smola, L. Song, A.G. Wilson, A la carte | learning fast kernels, in: Proc. 18th Int. Conf. Artif. Intell. Stat., volume 38, 2015, pp. 1098–1106.
[50] J. Zhang, A. Cloninger, R. Saab, Sigma-delta and distributed noise-shaping quantization methods for random fourier features, 2021, arXiv:2106.02614
[51] C. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, in: Proc. 14th Annu. Conf. Neural Inf. Process Syst., volume 13, 2001, pp. 682–688.
[52] A.J. Smola, B. Schölkopf, Sparse greedy matrix approximation for machine learning, in: Proc. 1th Int. Conf. Mach. Learn., 2000, pp. 911–918.
[53] A. Cloninger, Prediction models for graph-linked data with localized regression, in: Proc. SPIE Int. Soc. Opt. Eng., volume 10394, 2017, doi:10.1117/12.2271840.
[54] S. Ma, R. Bassily, M. Belkin, The power of interpolation : Understanding the effectiveness of SGD, in: Proc. 35th Int. Conf. Mach. Learn., 2018, pp. 3331–3340.
[55] S. Ma, M. Belkin, Kernel machines that adapt to GPUs for effective large batch training, 2018, arXiv:1806.06144
[56] J. Picka, Statistical inference for disordered sphere packings, Stat. Surv. 6 (2012) 74–112, doi:10.1214/09-SS058.
[57] M. Hifi, R. M'Hallah, A literature review on circle and sphere packing problems: models and methodologies, Adv. Oper. Res. (2009), doi:10.1155/2009/150624.
[58] P. Flajolet, D. Gardy, L. Thimonier, Birthday paradox, coupon collectors, caching algorithms and self-organizing search, Discrete Appl. Math. 39 (3) (1992) 207–229, doi:10.1016/0166-218X(92)90177-C.
[59] G. Klambauer, Problems and propositions in analysis, Marcel Dekker, New York, 1979.
[60] H. Wendland, Scattered data approximation, Cambridge University Press, 2004, doi:10.1017/cbo9780511617539.
[61] D. Gómez, A more direct proof of gerschgorinós theorem, Mat: Enseñanza Univ. 14 (2) (2006) 119–122.
[62] T. Hofmann, B. Schölkopf, A.J. Smola, Kernel methods in machine learning, Ann. Stat. 36 (3) (2008) 1171–1220, doi:10.1214/009053607000000677.
[63] M. Aiserman, E.M. Braverman, L.I. Rozonoer, Theoretical foundations of the potential function method in pattern recognition, Avtomat. i Telemeh. 25 (6) (1964) 917–936.
[64] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proc. 5th Annu. Workshop. Comput. Learn. Theory, 1992, pp. 144–152, doi:10.1145/130385.130401.
[65] G.S. Kimeldorf, G. Wahba, A correspondence between bayesian estimation on stochastic processes and smoothing by splines, Ann. Math. Stat. 41 (2) (2011) 495–502, doi:10.1214/aoms/1177697089.
[66] B. Schölkopf, R. Herbrich, A.J. Smola, A generalized representer theorem, in: Int. Conf. Comput. Learn. Theory, 2001, pp. 416–426, doi:10.1007/3-540-44581-1_27.

Paper II

Improving inversion of model parameters from action potential recordings with kernel methods

Andreas Oslandsbotn, **Alexander Cloninger**
Nickolas Forsch

Submitted to Mathematical Biosciences and Engineering

(BioRxiv: doi.org/10.1101/2023.03.15.532862)

Improving inversion of model parameters from action potential recordings with kernel methods

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

Abstract

Current methods for solving inverse problems in cardiac electrophysiology are limited by their accuracy, scalability, practicality, or a combination of these. In this proof-of-concept study we demonstrate 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. Using synthetic training data from the 1977 Beeler-Reuter AP model of mammalian ventricular cardiomyocytes, we demonstrate our recently proposed boosted kernel ridge regression (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 exposed to parameter sensitivity issues than existing methods, such as standard KRR solvers and loss-minimization schemes based on nearest-neighbor heuristics.

1 Introduction

Measurement of ionic currents in cardiomyocytes can provide key insights into the mechanisms of dysfunctional cardiac electrical properties, with important applications such as cardiac antiarrhythmic drug development. 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 [1, 2].

Meanwhile, the dynamic cardiomyocyte membrane potential, commonly referred to as the action potential (AP), is accessible optically using live cell fluorescence microscopy and more directly via microelectrode arrays for certain *in vitro* model systems [3, 4, 5]. The AP is usually recorded over some time interval, and the corresponding time series is commonly called an AP trace. The AP trace and its characteristic shape are determined by the biophysical dynamics of these ionic membrane currents [6, 7]. Furthermore, numerous cardiac action potential models are developed to describe the relationship between the AP and the underlying ionic currents [8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. This makes it possible, at least in theory, to quantify ionic currents from observations of the AP through the use of an AP model. The challenge is that existing AP models are constructed to describe the mapping from ionic currents to the AP, and we are interested in the inverse relationship; namely, given an AP trace, we want the underlying currents.

[*] Department of Informatics, University of Oslo, Problemveien 11, 0313 Oslo, Norway

[†] Department of Data Science and Knowledge Discovery, Simula Research Laboratory, Kristian Augusts gate 23, 0164 Oslo, Norway

[‡] Department of Mathematics, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, USA

[§] Department of Computational Physiology, Simula Research Laboratory, Kristian Augusts gate 23, 0164 Oslo, Norway

We can formulate this problem mathematically by representing the ionic membrane currents as a vector $p = (p_i, \ldots p_d) \in \mathcal{P} \subseteq \mathbb{R}^d$ and the corresponding AP trace as generated by an unknown function $v = f(p)$. AP models construct a function $\widetilde{f}$ that approximates f, typically described as a system of ODEs or PDEs. The task of deriving ionic currents from AP trace measurements can then be defined as the inverse problem in Definition 1.1.

Definition 1.1 (The inverse problem)**.** *Given an AP trace w find the underlying ionic membrane currents $p_w = f^{-1}(w)$ using the AP model $\widetilde{f}$.*

The usual approach in the literature for solving this inverse problem is to define a loss function L on a set of AP features $\{\phi_i(v)\}_{i=1}^m$ (represented as a vector $\phi(v) = (\phi(v)_1, \ldots, \phi(v)_m) \in \mathbb{R}^m$) and seek to minimize this loss function using numerical optimization schemes. The AP features are designed to capture important biophysical properties of the AP traces, and their construction generally requires specialist domain knowledge. Commonly used optimization schemes are direct search methods (non-gradient-based), such as Particle swarm [18], Pattern-search, Nelder-Mead, genetic algorithms [19, 20] or combinations of these methods [21]. For example, Jæger et al. [22] used Nelder-Mead combined with the continuation method [23]. The strategy of these methods is to search iteratively in the parameter space $\mathcal{P}$, and for each candidate, $p \in \mathcal{P}$, solve the ODE model $\widetilde{f}(p)$ to compare with the target w through the loss $L(\phi(w), \phi(\widetilde{f}(p)))$.

The problem with these classical inversion strategies is their low scalability, which makes them inefficient in the face of large quantities of measured APs. 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 [24, 25] and, therefore, more expensive to compute. In addition to this, iterative optimization schemes are often vulnerable to local minima. This is problematic, especially in clinical applications where high reliability is essential, as wrong predictions can have severe consequences.

In addition to issues with scalability and reliability, the inversion of AP traces in cardiac electrophysiology faces the issue of parameter identifiability. This issue typically occurs when more than one parameter affects the AP so that their joint effect is significantly reduced or canceled [26]. This is especially problematic in large complex models, which include multiple inward and outward currents. A special case of this issue is when a parameter has low sensitivity, namely a small (hard to detect) effect on the AP [27]. The issue of parameter identifiability has been studied in several works [28, 26, 29, 27]

In general, successful AP inversion schemes should be able to satisfactorily address the following aspects:

1. *Scalability*, both in the sense of the number of parameters to estimate and in the sense of the quantity of data to parameterize.

2. *Reliability* over the parameter domain, both in the sense of accuracy and in the sense of variability in accuracy over the parameter domain

3. *Identifiability* and *sensitivity* issues with model parameters.

   (a) Sensitivity issues: Due to parameters that have a small (hard to detect) effect on the AP trace (such as the fast inward sodium (Na+) current in the 1977-Beeler-Reuter model)

   (b) Identifiability issues: Due to parameters whose corresponding effects (i.e., currents) are overlapping such that the combined effect on the action potential is hard to separate into two or more currents.

4. *Other considerations*: Analysis that is specific to studying AP dynamics under the effects of various drug compounds, such as cycle length and sampling rate.

Tveito et al. [30] introduced an alternative to traditional inversion strategies by optimizing over a pre-computed database of samples $\mathcal{D}_n = \{(p_i, v_i)\}_{i=1}^n$ using nearest neighbor heuristics [30]. 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 AP can reuse the same database. Consequently, the algorithm is significantly more scalable when estimating the parameters of several measured AP traces. The challenge with this approach is the large memory requirements of storing the data set. Furthermore, the accuracy is directly linked to the density of the pre-computed samples, a problem not present in the inversion schemes mentioned before. 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, reliability would be expected to vary across the parameter domain as a function of the distance to samples in $\mathcal{D}_n$.

In this study, we offer a different strategy, namely, to learn a model of the inverse map f^{-1} (the mapping from AP traces to the underlying ionic membrane currents). To achieve this, we propose to use the recently developed streaming multi-resolution adaptive kernel algorithm StreaMRAK [31], a kernel method based on kernel ridge regression (KRR). This is a natural choice since kernel methods such as StreaMRAK and other state-of-the-art KRR solvers such as FALKON [32] have already been proven as large-scale learning schemes [32, 33, 31].

Furthermore, kernel methods can generate highly non-linear features of the AP trace in a data-driven manner that can replace the tailored AP features used in traditional methods. The feature space generated by kernel methods allows the modeling of highly non-linear functions without the need for explicit assumptions on the shape and structure of the function; this gives promise for modeling highly involved relationships between AP traces and ionic currents. Moreover, KRR is a convex optimization problem, which rules out local minima.

The goal of this proof-of-concept study is to introduce kernel methods as a new approach to the inverse problem in cardiac cell parameter estimation. The focus is, in particular, to address the issues of scalability (1) and reliability (2) exhibited by existing methods. We also study how our new method performs with respect to the aforementioned parameter sensitivity issues (3a), i.e., parameters whose effect on the AP is difficult to detect. Analysis of identifiability issues due to currents that compound or negate each other will remain to be analyzed in a future study. Similarly, will other issues related to specific AP dynamics.

For a thorough analysis, the study is restricted to a selected subset of parameters in a well-studied model of cardiac cell electrophysiology, the Beeler-Reuter (1977) model of mammalian ventricular cardiomyocytes [34]. The BR model was the first ventricular cell AP model to be developed and included 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 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.

Of particular interest is the fast inward sodium (Na^+) current, which affects the slope and peak of depolarization. This current is well known to be difficult to characterize due to its short activation relative to the entire AP and the low degree of sensitivity of the AP to changes in this current [35]. We will study how the proposed inversion scheme handles a parameter sensitivity problem using the Na^+current.

To demonstrate our approach, we run our method and others on several synthetic experiments on data generated from the BR model. The contribution of our study is summarized in the following; see also Figure 1 for a comparison to existing methods.

- Our approach relies on a pre-computed database, similar to Tveito et al. [30], which allows better scalability compared to standard inversion schemes.

- By learning a model, instead of relying directly on the dataset, our proposed method reduces memory requirements relative to the scheme used in Tveito et al. [30], from $\mathcal{O}(n)$ to $\mathcal{O}(C\sqrt{n})$,

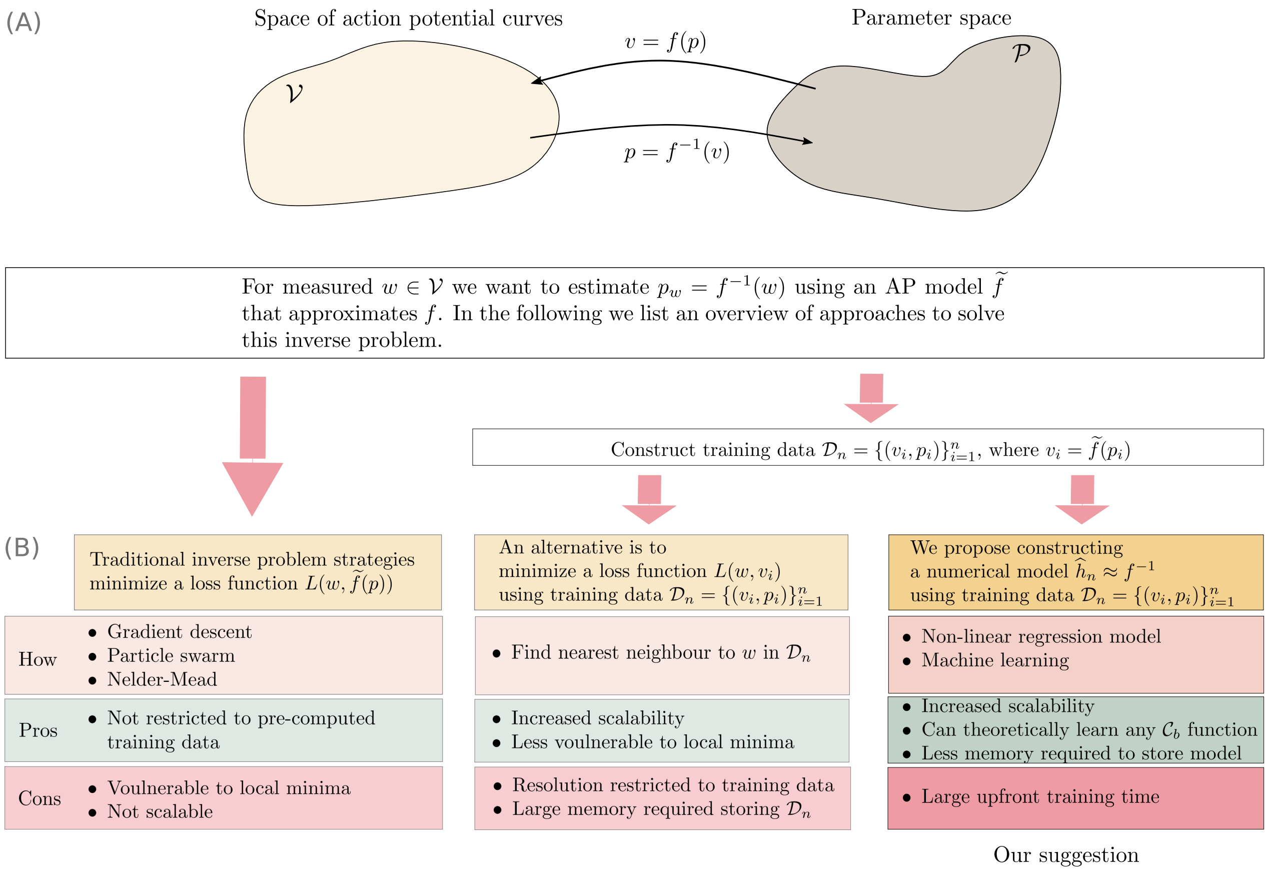

(A)
Space of action potential curves
Parameter space
$v = f(p)$
$\mathcal{V}$
$\mathcal{P}$
$p = f^{-1}(v)$
For measured $w \in \mathcal{V}$ we want to estimate $p_w = f^{-1}(w)$ using an AP model $\widetilde{f}$ that approximates f. In the following we list an overview of approaches to solve this inverse problem.
Construct training data $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$, where $v_i = \widetilde{f}(p_i)$
(B)
Traditional inverse problem strategies minimize a loss function $L(w, \widetilde{f}(p))$
An alternative is to minimize a loss function $L(w, v_i)$ using training data $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$
We propose constructing a numerical model $\widehat{h}_n \approx f^{-1}$ using training data $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$
How
• Gradient descent
• Particle swarm
• Nelder-Mead
• Find nearest neighbour to w in $\mathcal{D}_n$
• Non-linear regression model
• Machine learning
Pros
• Not restricted to pre-computed training data
• Increased scalability
• Less voulnerable to local minima
• Increased scalability
• Can theoretically learn any $\mathcal{C}_b$ function
• Less memory required to store model
Cons
• Voulnerable to local minima
• Not scalable
• Resolution restricted to training data
• Large memory required storing $\mathcal{D}_n$
• Large upfront training time
Our suggestion

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.

at the cost of an upfront computational training time. Here C is a constant independent of n.

- We demonstrate with synthetic experiments how StreaMRAK can estimate model parameters with increased reliability compared to alternative methods.

- For the analysis, we propose a geometric approach to identifiability analysis, which supplements the tool developed in Jæger et al. [27].

The current study provides a solid foundation with valuable insights for further analysis. In future work, we will focus on extending the analysis to more complex AP models with more parameters, states, and variables, such as Tusscher and Panfilov [9] and O'Hara et al. [11] or even models of cardiomyocytes derived from human induced pluripotent stem cells, such as Paci et al. [36].

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-vitro* 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-vitro* or generate them from a system of ODEs, we need to sample the traces at a finite time grid. We denote this time grid $[t_1, \dots, t_k, \dots t_T]$, 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 $p = (p_1, \dots, p_d)$ 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 $\mathcal{D}_n = \{(\widetilde{f}(p_i), p_i) : p_i \sim \text{Uni}(\mathcal{P}), \text{ for } \ i = 1, \dots n\}$. 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 parameters representing specific membrane currents. In this study, the parameters we consider $p = (p_1, \dots p_d)$ are scaling parameters that alter the amplitude of ionic 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. For a more detailed list of stimulus currents and other variables of the ODE implementation, we refer to Table S.3 in the Supplementary.

Table 1: Scaling parameters that alter the amplitude of ionic currents in the Beeler-Reuter model. The first column lists the parameter names. The second contains scaling constants, and the last column contains a description.

Parameter	Scaling	Description
g_{Na}	4.0×10^{-2} mS/mm^2	Sodium (Na^{2+}) ionic current component (mS/mm^2)
g_s	9.0×10^{-4} mS/mm^2	Slow inward ionic current component (mS/mm^2)
g_K	3.5×10^{-3} mS/mm^2	Time-independent potassium (K$^+$) current (mS/mm^2)
g_{x1}	1.9×10^{-3} mS/mm^2	Time-dependent outward current (mS/mm^2)

The ODEs that make up the Beeler-Reuter model are described mathematically in Halfar [37]. 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 $v_i = (v_{i1}, \ldots, v_{ik}, \ldots, v_{iT}) \in \mathcal{V}$ is the AP curve that corresponds to the parameter choice $p_i = (p_{i1}, \ldots, p_{ij}, \ldots, p_{id}) \in P$. 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 paced at 1 Hz with a sampling rate of 1 kHz, where only the first half of the pulse is included for the regression. Because the waveform is contained in the first 500 ms of the AP pulse, this reduces the problem's dimensionality without losing any information about the shape of the waveform. The time interval is chosen to be sufficiently large for all variations of AP traces over the domain $\mathcal{P}$ to re-polarize to the resting state.

Furthermore, the AP recording methods we are trying to simulate with our synthetic data do not obtain absolute measurements of the resting potential. Because of this, we align the resting potential for all traces in our experiments to represent the likely case where we do not have this information.

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 second column in Table 1. Under this scaling, the parameter $p = (1, \ldots, 1)$ 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 Choice of parameters for demonstrating parameter estimation capabilities

The goal of this work is to study the parameter estimation capability of our method compared to existing schemes with the purpose of creating a solid foundation for further application of the proposed method to more complex models. With this intent, we have chosen to use the fast inward Na^+ current and the slow inward Ca^{2+}-type current to demonstrate how the method can capture distinctly different changes to the AP trace with high reliability.

We note that the currents being studied here are commonly examined when investigating anti- and pro-arrhythmic effects of cardiac drugs [35] and are, therefore, natural choices in their own right. Furthermore, modulations of the fast Na^+ current are known to be difficult to detect due to it only being active during the fast upstroke of depolarization and considering the sampling constraints of modern measurement systems. Estimating the Na^+ current is, therefore, an interesting challenge for the inversion model when evaluating precision and reliability. Furthermore, the analysis is constrained to two model parameters to promote visualization and interpretation of the analysis. The authors refer the reader to Section 4.4 and 4.5 for more rationale on this choice.

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.

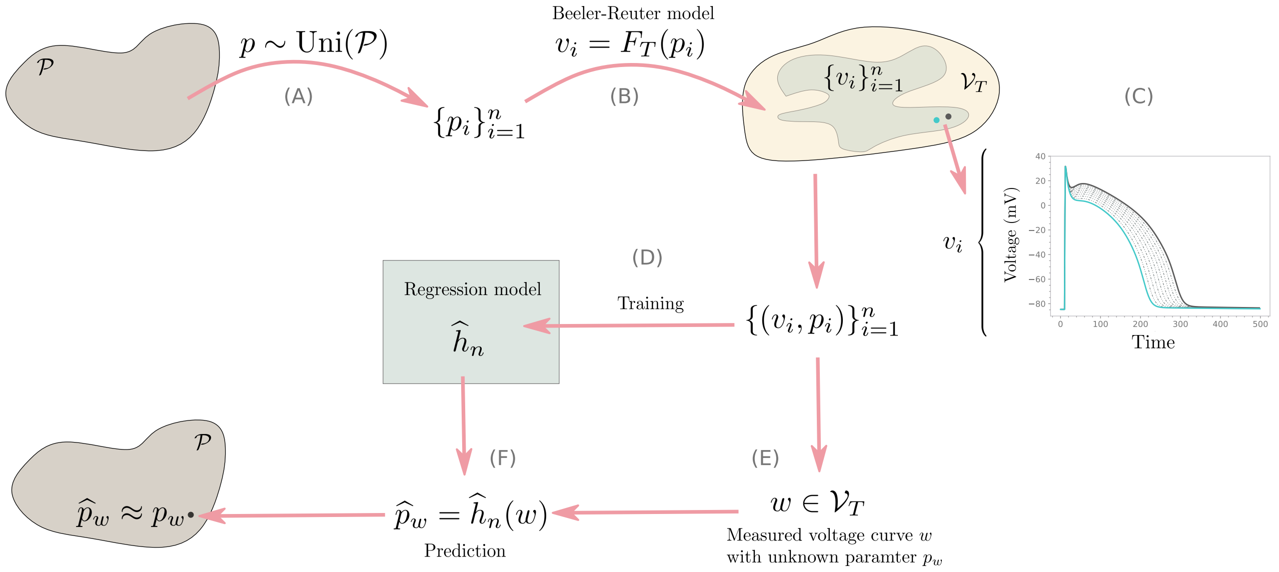


Figure 2: Illustration of our method

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 [31], 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, $j \in 1, \dots, d$ 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 [38], 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. StreaMRAK introduces a multi-resolution formulation of the model in (2.2). Details on this are given in Appendix 6.5.

We note that with the choice of the Gaussian kernel, the model used in Eq. (2.2) corresponds to interpolation with radial basis functions. Interpolation with radial basis functions is a common choice in machine learning and regression, for which the task is to learn the weights of the expansion. Numerous approaches exist for learning these coefficients, focusing on efficiency and the bias-variance tradeoff; an overview of different approaches can be found in [39]. The approach we use for this purpose is based on kernel regularized ridge regression (KRR) which is supported by a rich theoretical foundation that guarantees the method's reliability in terms of bounds on the bias and variance of the approximation. In the next section, we discuss the methods we use for finding the coefficients in more detail.

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 [31]. 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 [4]. In this regard, the time grid underlying the training data must be the same as for the AP traces w measured *in-vitro*. 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, by embedding them in the same vector space.

However, should the time grids differ initially, it is possible to align them by time interpolation or down sampling, which is possible because the AP traces are smooth functions of time. This way, one can avoid training the model again for different experimental recording systems that differ in sampling frequency.

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. Given a measured AP trace $w \in \mathcal{V}_T$ and a pre-computed training set $\mathcal{D}_n = \{(v_i, p_i)\}_{i=1}^n$ the methods finds

$$v_{opt} = \underset{v \in \{v_i\}_{i=1}^n}{\text{argmin}} \; L(w, v).$$

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 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) ϕ_j, such as AP duration. A complete list of APF and detailed descriptions can be found in S3.1 of Jæger et al. [22].

The motivation for finding v_{opt} is that this provides a lower bound on the prediction accuracy using the bounding box search used in Tveito et al. [30]. Clearly, for sufficiently large n, finding v_{opt} is computationally impractical, which is also why Tveito et al. [30] instead made use of an iterative search in bounding boxes (subsets) of $\mathcal{D}_n$. However, for our purposes, we are interested in accuracy and reliability comparisons, and as such, v_{opt} is a natural choice.

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 Jæger et al. [27], typical AP models have parameters that result in similar or overlapping changes, and the sensitivity of the AP to parameter perturbations 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 Jæger et al. [27], for determining parameter sensitivity and 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 [40], which offers a geometric understanding of the mapping between parameters and AP traces.

2.4.1 Spectral analysis

Jæger et al. [27] 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, 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 in Jæger et al. [27, 22].

2.4.2 Geometrical analysis

We propose to use Laplacian eigenmaps (LE) [40, 41] 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 $\sigma_1 \leq \ldots, \leq \sigma_n$, 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 Belkin and Niyogi [40].

The similarity to the spectral analysis proposed in Jæger et al. [27] 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 parameter estimation capabilities of the proposed kernel model StreaMRAK with four loss-minimization schemes similar to those used in Tveito et al. [30] and Jæger et al. [22]. 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.

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 comparing the reliability of the inversion schemes, we consider the accuracy and how it varies over physiologically relevant regions of the parameter space. This is because the application of the inversion scheme we are demonstrating is ultimately a tool for estimating drug effects from *in vitro* AP recordings. A comparison of the inversion schemes is made in light of what is desirable for drug effect estimation. 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 negative implications.

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 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 Finsberg [42]. 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

The experiments performed in this study were designed to rigorously compare the accuracy and variability of parameter estimation for the different methods over physiologically relevant regions.

The first experiment, covered in Section 3.1, gives an overview by inspecting the heterogeneity of the estimation errors over the parameter domain $\mathcal{P}$. Meanwhile, the two succeeding sections, Section 3.2 and 3.3, present experiments that inspect the accuracy as a function of specific directions (in $\mathcal{P}$) and distances away from the baseline biophysical cell at $p_{(0)}$.

Finally, Section 3.4 demonstrates the proposed inversion scheme as a tool for estimating drug effects using an example with simulated drugs. For this purpose, consider four hypothetical drugs $[A, B, C, D]$ whose perturbation directions θ are indicated in the right panel of 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 simulated drug.

The experiment conducted in Section 3.2 estimates the parameters of the AP traces along each of the concentric circles in the left panel of Figure 3. The purpose is to examine how the estimation accuracy varies with the direction away from $p_{(0)}$ in the parameter space. This experiment allows us to see how the different inversion schemes compare when more than one parameter is perturbed simultaneously.

Meanwhile, the experiment conducted in Section 3.3 examines how the accuracy varies with the distance from $p_{(0)}$ along the specific directions in parameter space. In particular, we consider the directions A, B, and C, as illustrated in the right panel in Figure 3, which can be related to the action of relevant drugs acting on the Na^{+}- and Ca^{2+}-type currents. The purpose is to compare the reliability of the inversion schemes, both in terms of estimation accuracy and 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 [43].

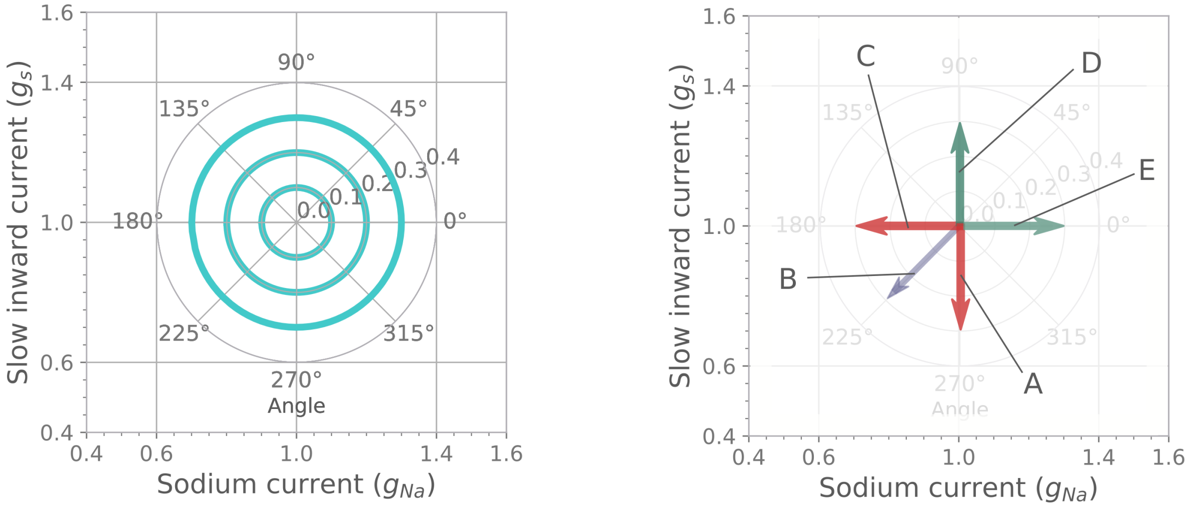


Figure 3: Direction and magnitude of perturbations in parameter space for simulated drug effects on the standard cardiac action potential. The left panel shows the three perturbation magnitudes $r = \{0.1, 0.2, 0.3\}$ for directions uniformly distributed on $[0°, 360°]$. The right panel indicates the 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.

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: Domain error

This experiment studies the heterogeneity of the estimation error over the parameter domain $\mathcal{P} = [0.2, 2]^2$ 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 4.

Panel (A)-(B) in figure 4 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

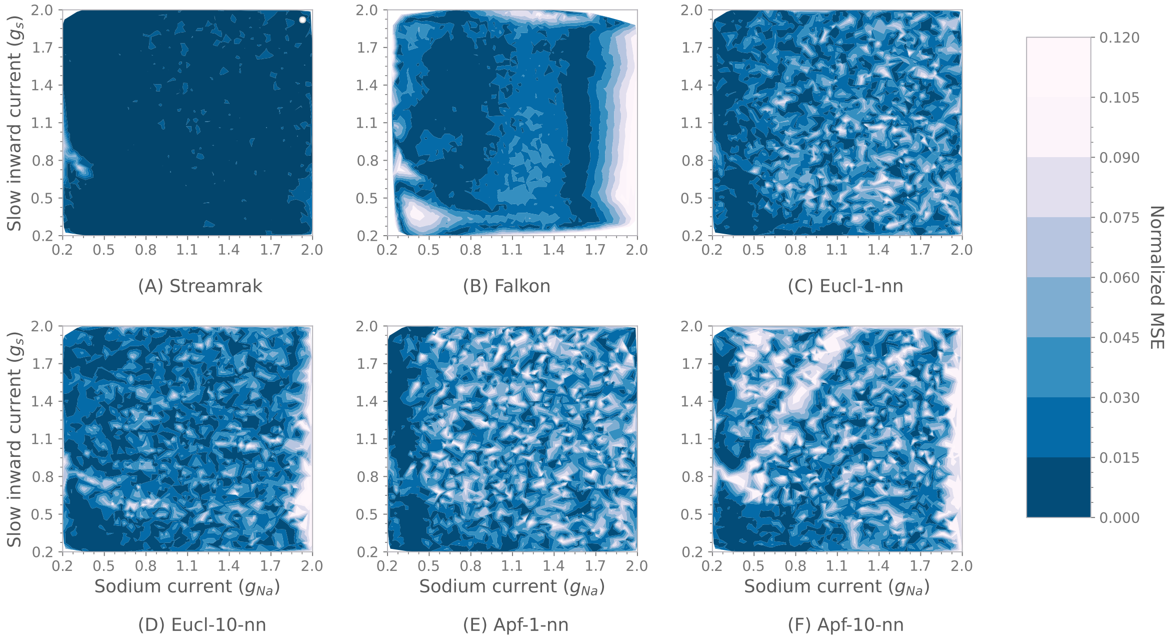


Figure 4: Experiment from Section 3.1. Distribution of estimation error over the domain $P = [0.2, 2]^2$.

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.

3.2 Experiment two: 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 $(g_{Na,i}, g_{s,i})$ along 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 5 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 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 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, note that for these, the regions with the best estimation accuracy are not along 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, for the loss-minimization algorithms, perturbations of g_{Na} affect the estimation accuracy of g_s. Meanwhile,

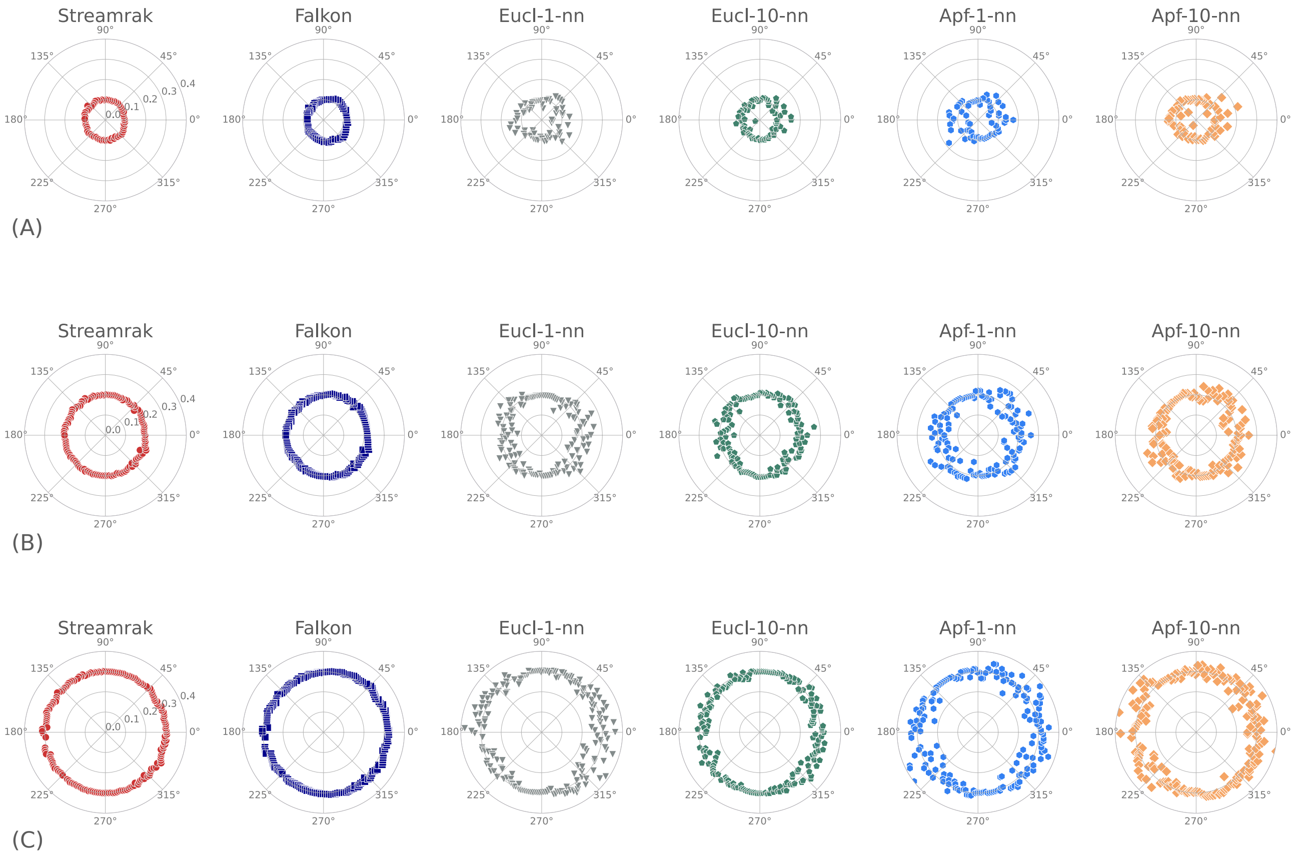


Figure 5: Estimation of perturbations with increasing magnitude for directions in interval $[0°, 360°]$. Here panel (A) corresponds to perturbation radius $r = 0.1$ from the unperturbed $p = (1.0, 1.0)$ (Smallest perturbation circle in Figure 3). Panel (B) corresponds to perturbation radius $r = 0.2$ (Middle perturbation circle in Figure 3). Panel (C) corresponds to perturbation radius $r = 0.3$ (Largest perturbation circle in Figure 3).

the regression models exhibit more confidence in identifying parameters. Notably, for StreaMRAK and FALKON, the estimation accuracy of g_s, g_{Na} does not depend on the perturbation direction and magnitude.

3.3 Experiment three: Accuracy as a function of perturbation magnitude

This experiment studies how prediction accuracy varies with perturbation magnitude away from the baseline biophysical cell at $p_{(0)}$. For this purpose, the directions A, B, and C introduced in Section 3.4 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. Figure 6 shows the estimation results.

Panel (A) in Figure 6 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. Table S.1 and Table S.2 in the supplementary quantify this by showing the three metrics (RMSE, Max abs. error, Standard deviation) over the interval $[0, 0.3]$.

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 Jæger et al. [22], have large fluctuations in estimation accuracy also for g_s. Again, using the three metrics (RMSE, Max abs. error, Standard deviation), the difference in reliability is quantified; see Table S.1 and Table S.2 in the supplementary.

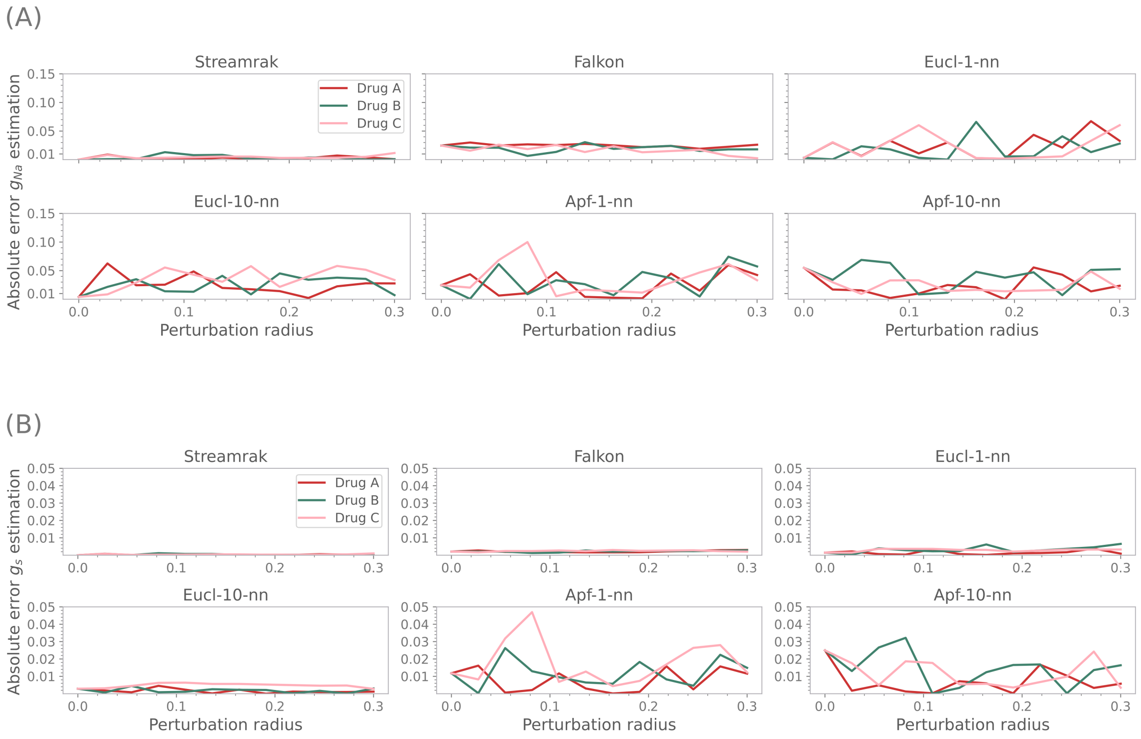


Figure 6: Experiment from Section 3.3. Figure showing parameter estimation accuracy, as measured by the absolute error, along the three selected directions A, B, and C as indicated in Figure 3. For each direction, 20 samples are evenly spaced on the interval $[0, 0.3]$. The red line is direction A, the 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.

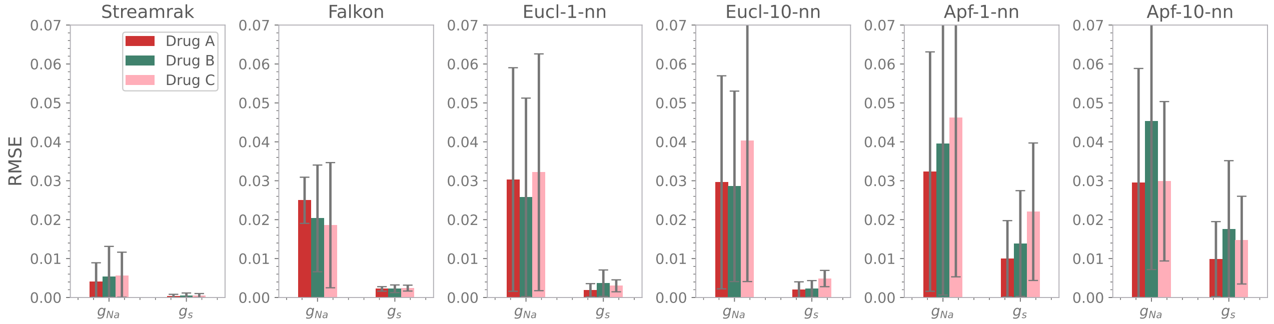


Figure 7: Figure showing the RMSE with error bars of the parameter estimations along three different perturbation directions on the interval $[0, 0.3]$. The perturbation directions correspond to the directions of Drugs A (red), B (green), and C (pink) defined in Figure 3.

To further support these observations, Figure 7 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 directions A, B, and C. The 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, the small standard deviation exhibited by StreaMRAK, and to some degree, FALKON, illustrates the higher consistency of StreaMRAK compared to the other methods.

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

g_s parameter.

3.4 Experiment four: Demonstration of the inversion scheme as a tool for drug effect estimation using an example with 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); [44, 45, 46, 47, 48, 49].

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.

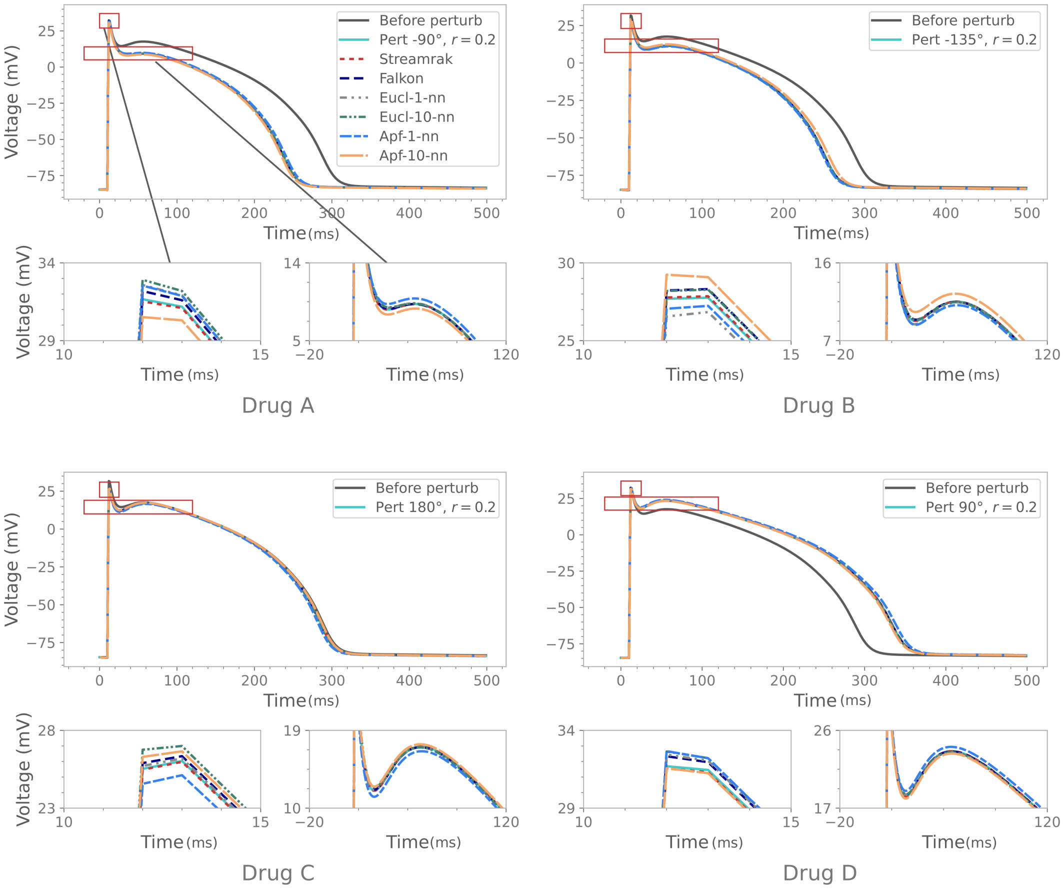


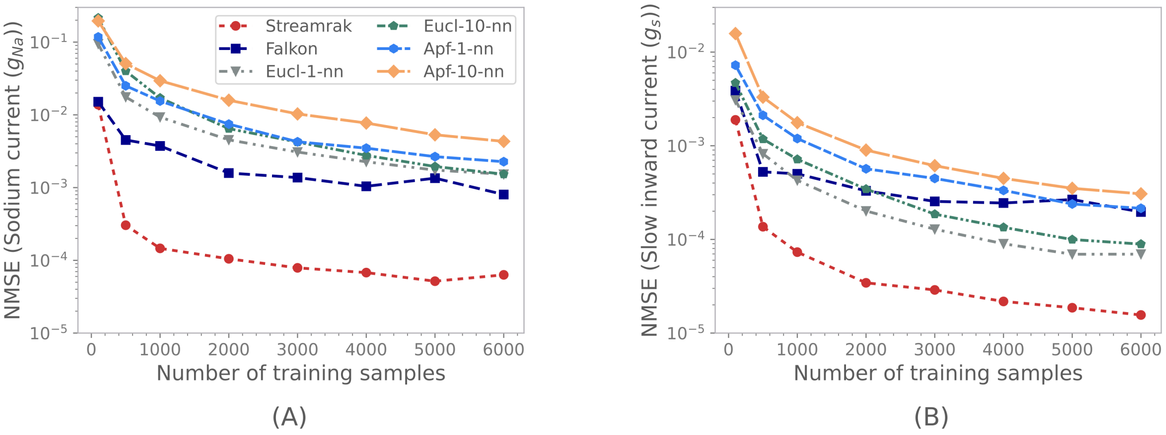
Figure 8: Comparison of how well the algorithms predict the drug effects of the drugs A, B, C, D from Figure 3. In each sub-figure, the upper panel shows the perturbation and the estimation from each algorithm. The unperturbed trace is included for reference. The lower panels highlight the first and the second ”peak” in the AP trace. The BR model is paced at 1 Hz with a sampling rate of 1 kHz. Only the first half of the pulse is included in the regression.

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 8 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 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 8. 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.

3.5 Performance analysis

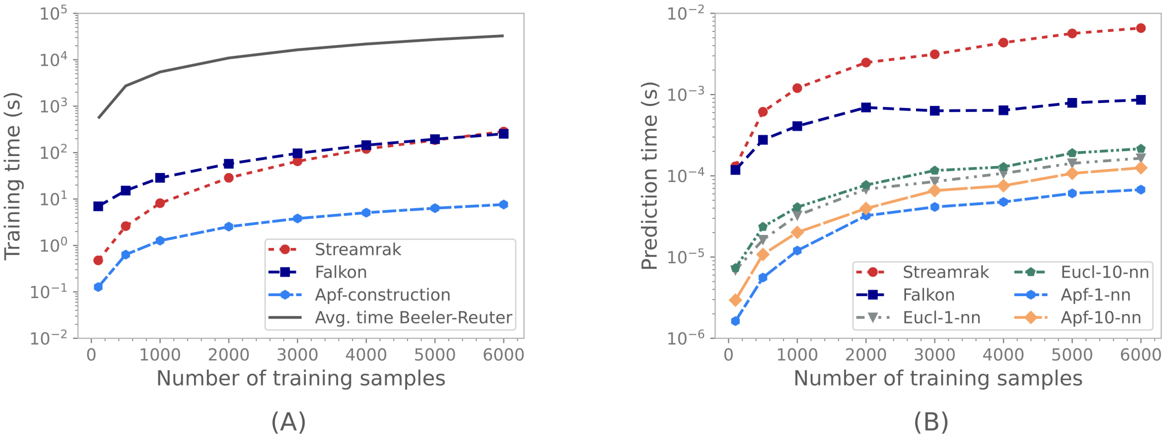
To compare the overall performance and scalability of the methods, their parameter estimation capability and their time complexity are analyzed as a function of the training data size. The MSE is taken over all test estimations for each parameter to quantify the parameter estimation capability. The test data is the same as in Section 3.1 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 $n \in [100, 500, 1000, \ldots, 6000]$.



(A)

(B)

Figure 9: 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 10: 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.

Consider the MSE over the test estimations for each parameter; the result is shown in Figure 9, 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 10 (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-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 9, 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 10 (A), and it is clear that it is several magnitudes higher than the training times.

Figure 10 (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

This study has 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 key findings are summarised in the following:

- Our experiments show that StreaMRAK estimates parameters with higher accuracy — roughly an order of magnitude higher — than both FALKON and the loss-minimization schemes.
- StreaMRAK, and to some degree FALKON, demonstrates a greater degree of reliability, both in terms of overall accuracy and consistency throughout the parameter domain.
- In particular, StreaMRAK demonstrates high reliability for the g_{Na} parameter, which the loss-minimization schemes struggle with.
- Our experiments show that StreaMRAK requires significantly fewer training examples to reach a high estimation accuracy than both FALKON and the loss-minimization schemes.

In the following, 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 supplement this discussion with a note on the identifiability and sensitivity of currents in the Beeler-Reuter model; see Section 4.3. In Section 4.4 and 4.5, we make some special remarks with respect to the choice of model, the subset of parameters that we analyze, and the generalization of the inversion scheme. Finally, we consider related work, limitations, and future work in Sections 4.6 and 4.7.

4.1 Accuracy and reliability of methods

The experiments show that StreaMRAK performed the best regarding the absolute prediction accuracy and the consistency of the estimate on repeated tests. In particular, it has a higher estimation accuracy — roughly an order of magnitude higher — than both FALKON and the loss-minimization schemes Eucl-1-nn, Eucl-10-nn, Apf-1-nn, and Apf-10-nn. Furthermore, the parameter estimations of both StreaMRAK and FALKON are significantly more consistent across the parameter domain as demonstrated by Figure 4. These are the aspects of parameter estimation from cardiac AP inversion we sought to improve.

These findings show the advantage of using StreaMRAK as a tool for parameter estimation in the context of drug development. This is because, when exploring drug mechanisms, for example in heart-on-chip systems, the reliability of parameter estimates is vital for making consistent and accurate estimates of 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. An inversion scheme with high reliability is, therefore, a sensible requirement when using it for the purpose of drug-effect discovery. This study demonstrates that StreaMRAK satisfies this requirement.

Inspection of the estimation accuracy for different parameter combinations shows how the kernel methods StreaMRAK and FALKON are reliable when learning from training data where more than one parameter is responsible for the modulations of the AP traces and also more reliable with respect to estimating different parameters; See figure 5. This demonstrates that StreaMRAK and FALKON are better suited than the other loss-minimization schemes for scaling to larger training tasks from more complex models with several parameters.

Furthermore, Figure 6 and 7 demonstrate how StreaMRAK is reliable with respect to parameter estimation along specific directions in parameter space. This is an important property for an AP inversion scheme when the goal is to use it as a tool for drug calibration during a dose-escalation study, since measuring the response under a range of doses is an essential part of drug characterization. Increasing the dose of specific drugs results in perturbations in the parameter space with increasing magnitude away from the baseline biophysical cell. High reliability as a given parameter varies is, therefore, a crucial factor for adequate calibration. This study demonstrates that StreaMRAK satisfies this requirement.

As a final note, we mention that for estimating drug effects for clinical applications, it is necessary with more complex models and training sets with significantly more parameters (higher dimensional training data). What we have demonstrated in this study is that StreaMRAK has more potential for extension to larger models compared to the other inversion schemes considered in this study. In Section 4.5.1 we discuss in more detail the scalability of StreaMRAK to larger learning tasks from more complex models.

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 [31], 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 10, this cost is significantly less than the training time of both StreaMRAK and FALKON.

Nevertheless, from panel (A) in Figure 9, 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 S.1 (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 10, 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 9.

In terms of parameter estimation time (prediction time), we see from Panel (B) in Figure 10 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 $\mathcal{O}(d \log n)$ [50] (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 four currents of interest: the rapid sodium, slow-inward, time-dependent-outward current, and time-independent outward current, whereof the first two are parameterized by respectively the g_{Na} and g_s parameters in this study. Using the spectral analysis from Jæger et al. [27], as described in Section 2.4.1, we can analyze the sensitivity and identifiability of these currents in the Beeler-Reuter model. Let $e = (1, 0, 0, 0)$ represent the sodium current, $e = (0, 1, 0)$ the slow inward current, $e = (0, 0, 1, 0)$ the time-dependent outward current, and $e = (0, 0, 0, 1)$ the time-independent outward current. From the analysis, we find the singular values $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4) = (707.8, 6.9, 0.18, 0.066)$ with corresponding singular vectors

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0.01 & 99.32 & 0.47 & -0.05 \\ 99.89 & -0.01 & 0.05 & -0.036 \\ -0.02 & 0.1 & -22.21 & -77.6 \\ -0.078 & -0.57 & 77.27 & -22.31 \end{pmatrix},$$

where the singular vectors are multiplied by a factor of 10^2 to enhance readability.

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 and time-independent outward currents both project along v_3 and v_4. Jæger et al. [27] 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 sensitivity 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.

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

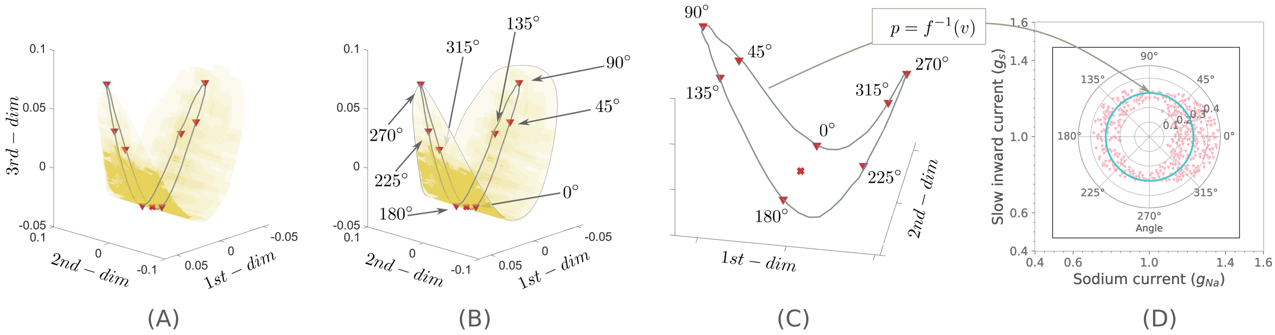


Figure 11: Panel (A) shows the 3-dimensional projected embedding of a ball in the space of action potential curves V (Embedding using the Laplacian Eigenmaps algorithm [40]). The grey ellipse in (A) corresponds to a circle of radius $r = 0.1$ centered at $p = (1, 1)$ in the parameter space. Panel (B) 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.

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 8, where we see that StreaMRAK is better at detecting critical features of the AP traces. For instance, consider the following example. A key identifier of Na$^+$ current inhibitors (such as quinidine) is the peak of depolarization. This drug effect can be simulated by perturbations along both C and B in Figure 3, and we see from Figure 8 that in both cases, 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 distinguish differences in this peak.

4.4 A note on the choice of parameters and model for the study

A limitation of using the 1977 Beeler-Reuter model is that it does not incorporate the presence of more specific inward and outward currents that are present in contemporary models and would introduce challenges with identifiability due to overlapping effects. These are challenges that warrant further investigation in order to extend our proposed methods to more clinically relevant scenarios; However, the focus of this study has been to compare StreaMRAK to existing schemes with respect to reliability across the parameter domain. Meanwhile, the analysis of identifiability issues that arise from overlapping current effects is left for a separate study. Because of this, the use of the 1977 Beeler-Reuter model is a natural choice for the purpose of this study.

This study did not include parameter estimations for the time-independent potassium K$^+$ current and the time-dependent outward current. These parameters are interesting from a drug screening

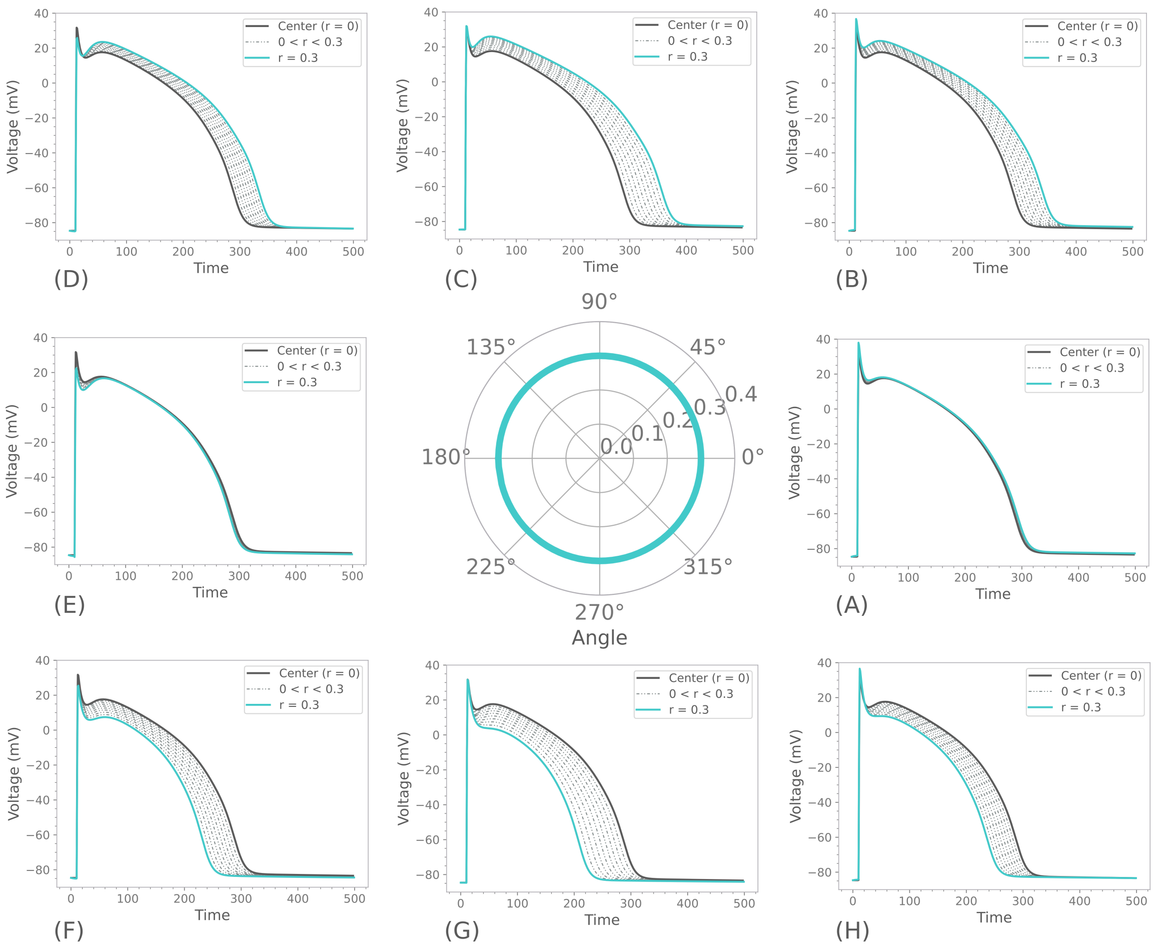


Figure 12: Illustration of how the voltage curves change as a function of direction and radius away from the center $p = (1, 1)$. The middle panel is a polar representation of the direction and radial distance away from $p = (1, 1)$. Each panel (A)-(H) shows 10 curves equally spaced on the interval [0, 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. The BR model is paced at 1 Hz with a sampling rate of 1 kHz. Only the first half of the pulse is included in the regression.

perspective and carry important clinical implications. However, for the purpose of this study, we have focused on rigorous analysis of a subset of parameters which enabled an interpretable and thorough comparison of different inversion schemes across the parameter domain. For demonstrating StreaMRAK as a tool for estimating drug effects with clinical utility, it will be necessary to consider more elaborate AP models with significantly more parameters, complexity, and challenges, such as Tusscher and Panfilov [9] and O'Hara et al. [11].

4.5 Generalization of the inversion scheme

We discuss the scalability of the inversion scheme to larger models and the generalization to more complex AP models. We also offer our thoughts on how the method will perform on AP traces recorded at multiple pacing rates and different time resolutions.

4.5.1 Scalability to larger models

The scalability of the kernel methods StreaMRAK and FALKON, to larger data sets with higher dimensionality (more parameters) has already been demonstrated in several studies [32, 51, 31]. Furthermore, because the learning algorithms are non-parametric, they do not rely on specific information about the AP trace and AP model at hand. As such, they should also work on training data generated by more complex AP models.

We note that as the number of parameters in the training data increase, one needs more training samples to maintain the estimation accuracy. This problem is known as the *curse of dimensionality* [52], and is a well-known reality of learning that also applies to the loss-minimization schemes Eucl-1-nn, Eucl-10-nn, Apf-1-nn, and Apf-10-nn. What we have shown in this study is that StreaMRAK requires significantly fewer training samples to achieve high accuracy for a given number of parameters. Therefore, StreaMRAK will be the better option when considering more complex models with more parameters.

4.5.2 Differences in AP recordings

For AP traces recorded with different recording systems, we distinguish between two cases: The first is AP traces with different pacing rates, which is relevant when studying, for example, anti- and pro-arrhythmic drug effects. The second is AP traces recorded at different sampling frequencies, which is often the case for different experimental setups and recording systems. For example, microelectrode arrays typically sample at frequencies in the order of $10 - 20$ kHz [5, 53], while live cell fluorescence microscopy sample at significantly lower frequencies $<$ 1kHz.

Regarding the first case. The pacing rate can affect channel kinetics and, therefore, the relationship between the AP trace and the underlying parameters. This means the inverse map f^{-1} we are approximating will differ depending on the pacing rate. As such, a model trained on a set of AP traces with the same pacing rate (as done in this study) is only representative of current parameters for AP traces with this pacing rate (or very similar pacing rates). We note that this would also be the case for any other learning scheme since changing the pacing rate alters the target function to learn.

Because of the importance of pacing rate with respect to detecting antiarrhythmic effects, future studies should consider the performance of StreaMRAK in this regard. The extension of StreaMRAK to parameter estimation from multiple pacing rates can be done straightforwardly. The first step is to include more than one pulse of the AP trace; In Cairns et al. [20], this was done by including two pulses, which should be sufficient. To allow the traces to be compared, the two-pulse AP traces must be embedded in the same vector space, as explained in Section 2.2.4. The next step is to train the model on these two-pulse AP traces using a selected set of relevant pacing rates $\mathcal{S}$. The performance of the inversion scheme should then be explored with respect to parameter estimations of AP traces with pacing rates both in $\mathcal{S}$ and outside of $\mathcal{S}$.

For clinical applications involving *in-vitro* measurements of AP traces, the relevant pacing rates are determined by the practitioner in the lab, which measures the AP traces at pacing rates relevant to the problem that is studied. To invert these traces, the model should be trained at the same set of pacing rates. For purely *in-silico* experiments, the pacing rates used for training the model should, in a similar manner, be relevant to the problem that is studied.

Regarding the second case. As noted in Section 2.2.4, a requirement for the proposed method is that the sampling frequency of the training data and the AP traces measured *in-vitro* must be the same. Otherwise, the AP trace would not be comparable. However, should the time grids differ due to different sampling frequencies, it is possible to align them by time interpolation or downsampling, which is possible because the AP traces are smooth functions of time. This way, one can avoid training the model again due to differences in sampling frequency. Furthermore, in this study, we have demonstrated the method on synthetic AP traces with a sampling rate of 1 kHz. However, nothing prevents the method from being trained on any other reasonable sampling rate.

4.6 Complementary studies

This study has demonstrated StreaMRAK as a computationally efficient AP trace inversion scheme with greater reliability than alternative schemes that have been previously applied for this purpose [30]. As such, our contribution is a piece in a larger effort to develop efficient drug-effect-estimation pipelines based on high-throughput, optical measurements of AP traces.

Tveito et al. [30] demonstrate, using synthetic and experimental data, 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 mature ventricular cells. This scheme has implications for 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 inversion step. In particular, Jæger et al. [22] introduces a new computational scheme for inverting the AP model based on a continuation-based optimization algorithm [23], which searches for the optimal parameter of an AP trace by moving iteratively from a known initial parameter. Meanwhile, Jæger et al. [54] show 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, Gong and Sobie [55] developed a cross-cell type regression model for predicting responses in human adult ventricular cardiomyocytes from measurements in hiPSC-CMs. Meanwhile, Paci et al. [56] combined *in silico* simulation trials with optical measurments of AP and Ca^{2+} traces in hiPSC-CMs to estimate drug effects. Similarly, Passini et al. [57] estimated drug effects by combining *in silico* simulation trials 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. Lancaster and Sobie [58] 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. Aghasafari et al. [59] 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.

4.7 Limitations and future work

We have demonstrated the parameter estimation capability of StreaMRAK on the 1977 Beeler-Reuter AP model of mammalian ventricular cardiomyocytes [34]. 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 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 [60]. Furthermore, with more complex models come more involved identifiability problems; sensitivity problems due to subtle (hard to detect) changes to the AP, such as observed in this study with the g_{Na} parameter; And identifiability problems due to parameters with overlapping effects on the AP that either significantly reduce or cancel each other out [26].

For the extension to more complex AP models, it will be necessary with an in-depth study of the performance of StreaMRAK in the context of the second of these identifiability problems. In particular, one should aim at isolating a subset of parameters from a more complex model for which this problem occurs, similar to what has been done in this study with respect to scalability, reliability, and g_{Na}.

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. This is because reducing the number of training points reduces the computational burden of solving the system of ODEs in the AP model and the memory requirement of storing the samples. This reduction is significant due to the expense of solving the ODE compared to the training time, as seen from Figure 10. 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. Future work should aim to demonstrate this model in the drug-effect-identification pipeline developed in Tveito et al. [30]; Jæger et al. [22] and to extend the demonstration to AP models of immature human induced pluripotent stem cells (hiPSC-CMs), such as Paci et al. [36, 61]. The generalization to AP models of hiPSC-CMs is straightforward, as StreaMRAK is a non-parametric learning scheme that does not rely on model-specific input about the training data.

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 [1, 2]. Because of this, validation of the inversion of AP models of adult human cardiomyocytes will mainly be done using live cell fluorescence microscopy [3, 4]. 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 Conclusion

In this study, we have introduced StreaMRAK as a new tool for inverting cardiac AP traces using a subset of parameters from the 1977 Beeler-Reuter model. We have demonstrated that StreaMRAK is a scalable method that offers greater reliability than existing inversion schemes, especially for AP model parameters with effects that are hard to detect. This systematic study offers a foundation for applying StreaMRAK to larger and more complex AP models with clinical value.

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

[2] Chris Chambers, Ian Witton, Cathryn Adams, Luke Marrington, and Juha Kammonen. High-throughput screening of $\mathrm{Na}_V1.$ 7 modulators using a giga-seal automated patch clamp instrument. *Assay and drug development technologies*, 14(2):93–108, 2016.

[3] 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.

[4] 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.

[5] Mike Clements and Nick Thomas. High-throughput multi-parameter profiling of electrophysiological drug effects in human embryonic stem cell derived cardiomyocytes using multi-electrode arrays. *Toxicological Sciences*, 140(2):445–461, August 2014.

[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] Fulong Chen, Angdi Chu, Xuefei Yang, Yao Lei, and Jizheng Chu. Identification of the parameters of the beeler–reuter ionic equation with a partially perturbed particle swarm optimization. *IEEE Transactions on Biomedical Engineering*, 59(12):3412–3421, 2012.

[19] Z Syed, E Vigmond, Stanley Nattel, and LJ Leon. Atrial cell action potential parameter fitting using genetic algorithms. *Medical and Biological Engineering and Computing*, 43:561–571, 2005.

[20] Darby I Cairns, Flavio H Fenton, and EM Cherry. Efficient parameterization of cardiac action potential models using a genetic algorithm. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(9):093922, 2017.

[21] Axel Loewe, Mathias Wilhelms, Jochen Schmid, Mathias J. Krause, Fathima Fischer, Dierk Thomas, Eberhard P. Scholz, Olaf Dössel, and Gunnar Seemann. Parameter estimation of ion current formulations requires hybrid optimization approach to be both accurate and reliable. *Frontiers in Bioengineering and Biotechnology*, 3, 2016.

[22] 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.

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

[24] 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.

[25] 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.

[26] Martin Fink and Denis Noble. Markov models for ion channels: versatility versus identifiability and speed. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1896):2161–2179, 2009.

[27] 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.

[28] Socrates Dokos and Nigel H. Lovell. Parameter estimation in cardiac ionic models. *Progress in Biophysics and Molecular Biology*, 85(2):407–431, 2004.

[29] Maria T Mora, Jose M Ferrero, Lucia Romero, and Beatriz Trenor. Sensitivity analysis revealing the effect of modulating ionic mechanisms on calcium dynamics in simulated human heart failure. *PloS one*, 12(11):e0187739, 2017.

[30] 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.

[31] 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.

[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] Giacomo Meanti, Luigi Carratino, Lorenzo Rosasco, and Alessandro Rudi. Kernel methods through the roof: handling billions of points efficiently. *Advances in Neural Information Processing Systems*, 33:14410–14422, 2020.

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

[35] 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.

[36] 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.

[37] 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.

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

[39] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

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

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

[42] 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.

[43] 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`.

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

[45] 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.

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

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

[48] 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.

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

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

[51] Giacomo Meanti, Luigi Carratino, Lorenzo Rosasco, and Alessandro Rudi. Kernel methods through the roof: Handling billions of points efficiently. In *Advances in Neural Information Processing Systems*, volume 33, pages 14410–14422. Curran Associates, Inc., 2020.

[52] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *Computational Intelligence and Bioinspired Systems: 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10, 2005. Proceedings 8*, pages 758–770. Springer, 2005.

[53] Raha M. Dastgheyb, Seung-Wan Yoo, and Norman J. Haughey. Meanalyzer - a spike train analysis tool for multi electrode arrays. *Neuroinformatics*, 18(1):163–179, 2020.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

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

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

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

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

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

[67] 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.

[68] 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.

[69] 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 $\mathsf{x}_0 \in X$ if the jacobian $J_f(\mathsf{x}_0)$ is invertible. In other words, if the determinant $\det J_f(\mathsf{x}_0) \neq 0$.

Theorem 6.1 (The inverse function theorem). *(See for instance Krantz et al. [62]) 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 $\mathsf{x}_0 \in X$ and let $J(\mathsf{x}_0)$ be the Jacobian at x_0. If $J_f(\mathsf{x}_0)$ is invertible (i.e. $\det J_f(\mathsf{x}_0) \neq 0$), then there exists an open neighborhood $\mathcal{N}(\mathsf{x}_0)$ such that the inverse f^{-1} of $f : \mathcal{N}(\mathsf{x}_0) \to f(\mathcal{N}(\mathsf{x}_0))$ exists and $J_{f^{-1}}(f(\mathsf{x}_0)) = (J_f(\mathsf{x}_0))^{-1}$.*

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 $\mathsf{x} \in X$. Then the Jacobian matrix of f is defined as*

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

Definition 6.3 (Positive semi-definite matrix). *Let $\mathbf{A} \subset \mathbb{R}^{n\times n}$ be a symmetric matrix. If $\mathsf{x}^\top \mathbf{A}\mathsf{x} \geq 0$ for all $\mathsf{x} \in \mathbb{R}^n$ 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 $\mathbf{A}_{ij} = f(\mathsf{x}_i - \mathsf{x}_j)$ 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 $\mathsf{x}' \in \Gamma_\varepsilon$ we have $\varepsilon < d(\mathsf{x}', \mathsf{x}) < \varepsilon + \delta$ for all $\mathsf{x} \in \Gamma_\varepsilon$ such that $\mathsf{x} \neq \mathsf{x}'$. 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 Singer and Coifman [63] for a proof.

6.5 Kernel methods

The use of kernel methods for supervised learning has a strong theoretical foundation [64, 65, 66]. 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" [67, 68], 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

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

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 [69] 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 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 Rudi et al. [32] and Oslandsbotn et al. [31].

6.5.2 StreaMRAK

The streaming multi-resolution adaptive kernel algorithm StreaMRAK introduced in Oslandsbotn et al. [31], 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 $\mathbf{d}^{(l)} = (d_i^{(l)}, \dots, d_n^{(l)}) \in \mathbb{R}^n$ from the previous level. The residuals are defined as

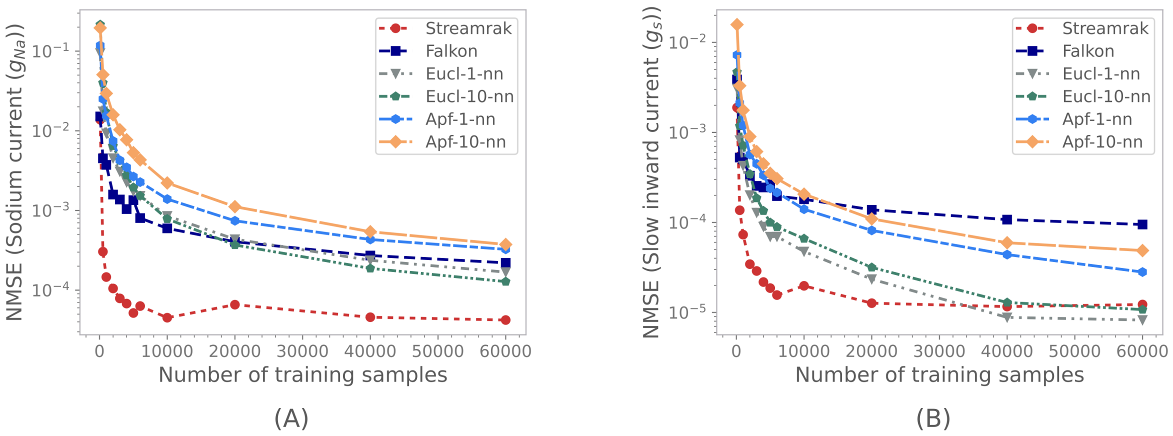
$$d_i^{(l)} = \begin{cases} y_i & l = 1 \\ y_i - h^{(l-1)}(\mathbf{x}_i) & l > 1. \end{cases}$$

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 [31].

Supplementary

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.



(A)

(B)

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)

Paper III

Effective resistance in metric spaces

Robi Bhattacharjee, **Alexander Cloninger**
Yoav Freund, **Andreas Oslandsbotn**

Submitted to Journal of Machine Learning Research

(Co-first-authors: Robi Bhattacharjee and Andreas Oslandsbotn)

(arXiv: doi.org/10.48550/arXiv.2306.15649)

Effective resistance in metric spaces

Robi Bhattacharjee[*] Alexander Cloninger[†] Yoav Freund[‡] Andreas Oslandsbotn[§¶]

Abstract

Effective resistance (ER) is an attractive way to interrogate the structure of graphs. It is an alternative to computing the eigenvectors of the graph Laplacian.

One attractive application of ER is to point clouds, i.e. graphs whose vertices correspond to IID samples from a distribution over a metric space. Unfortunately, it was shown that the ER between any two points converges to a trivial quantity that holds no information about the graph's structure as the size of the sample increases to infinity.

In this study, we show that this trivial solution can be circumvented by considering a region-based ER between pairs of *small regions* rather than pairs of *points* and by *scaling the edge weights* appropriately with respect to the underlying density in each region. By keeping the regions fixed, we show analytically that the region-based ER converges to a non-trivial limit as the number of points increases to infinity. Namely the ER on a metric space. We support our theoretical findings with numerical experiments.

1 Introduction

A fundamental task of data science is to model the structure of point clouds embedded in a high-dimensional ambient space. A common approach to this task is to represent the data as a graph where data points are considered vertices, and neighborhood information on the point cloud is encoded in edges connecting the vertices. The edges are often weighted based on the relative distance between points and are usually restricted to local neighborhoods. For simplicity of the introduction, we assume that an edge connects any two vertices whose distance is at most $\gamma > 0$. [1]

Measuring the lengths of paths on such graphs is a key component of many methods that seek to characterize point clouds. In the following, we compare two of the most popular graph metrics used for this purpose:

- **Shortest path** defines the distance between two vertices as the minimal number of edges that must be traversed to get from one vertex to the other. This is the most straightforward and intuitive measure of distance and corresponds to the geodesic distance when the point cloud lies on a differentiable manifold. However, the shortest path is sensitive to noise and the subtraction or addition of single points. It is, therefore, unreliable in the context of random point clouds.
- **Effective resistance (ER)**, also called *commute time* is significantly more stable than the shortest path distance as it considers all possible paths between two points instead of only the shortest.

[*] Department of Informatics, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, United States

[†] Department of Mathematics, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, United States

[‡] Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, United States

[§] Department of Informatics, University of Oslo, Problemveien 7, 0315 Oslo, Norway

[¶] Simula Research Laboratory, Kristian Augusts gate 23, 0164 Oslo, Norway

[1] More general definitions of edge weights will be given later.

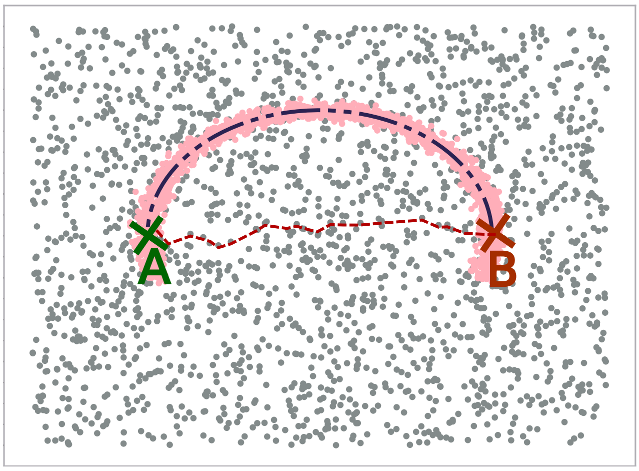


Figure 1: Comparison between ER distance and shortest path distance on a graph constructed on a point cloud. The point cloud consists of a high-density half-moon over a noisy background. The ER distance is the blue dotted line following the half-moon arch. The shortest path distance is the red dotted line across the center of the half-moon.

> Specifically, the *hitting time* of vertex B from vertex A is the expected number of edges traversed by a random walk starting at A and ending at B. The *commute time* is the sum of the hitting time from A to B and from B back to A. The *effective resistance* is a metric defined using analogy to resistance networks and is equal to the commute time up to a constant.

Figure 1 shows the difference between the shortest path and the ER distance on a point cloud shaped as a high-density half-moon over a noisy background. The figure illustrates how the shortest path distance would follow the noisy background, while the ER would follow the half-moon arch.

ER has been used in a number of applications in machine learning on topics such as graph sparsification [1], online learning [2], detecting community structure [3] and dimensionality reduction [4]. It has also been used in several other fields ranging from bioinformatics [5] to electronics [6, 7, 8, 9]

A long-standing issue with using ER for point clouds is that the ER distance between points converges to a trivial limit as the number of sampled points increases. This problem has been described in a long line of works [10, 11, 12, 13, 14]. The problem is that in the large graph limit, the ER between two points A and B depends only on the degree (number of neighbors) of A and B. This means that ER distances are not useful for the analysis of the shape of point clouds.

The trivial limit can be described as follows:

Problem 1 (Von-Luxburg limit)**.** *The ER between two nodes i, j in a graph converges as the number of nodes in the graph increases to $1/d_i + 1/d_j$, where d_i, d_j are the degrees of respectively node i and j.*

While the statement of the problem is asymptotic, Von-Luxburg et al. [14] shows that the asymptotic behavior begins already for moderately sized graphs, with the number of nodes in the order of 1000. Moreover, for increasing dimension of the embedding space, this convergence is even faster.

This study aims to develop a new formulation of ER that does not suffer from Problem 1 and can thereby be used to characterize large point cclouds.

The key idea is to consider the RE between small regions, rather than individual points. The number of points in a region scales linearly with the total number of points, which avoids the collapse of the RE in the limit. While this is in principle a simple transformation, some care is required to prove that it works as desired.

Our contributions can be summarized as follows:

- We alleviate the problem described by Von-Luxburg et al. [13] by introducing the concept of a region-based ER between a source and sink region, combined with an appropriate scaling as the sample size grows.

- We prove the existence and convergence of region-based ER to a non-trivial limit.
- We prove that region-based ER is a distance metric. In particular, region-based ER satisfies the triangle inequality.
- We support our theoretical findings with several numerical experiments.

The remainder of this paper is organized as follows. In Section 2, we introduce the concept of a resistor graph, the standard definition of ER, and the definition of ER between sets. In Section 3, we introduce the concept of a metric graph and extend the concept of effective resistance to metric spaces. In Section 4, we show how a finite sample region-based ER converges, with the appropriate scaling, to the limit object region-based ER on a metric space. We provide further results for the region-based ER in Section 5, where we show that it corresponds to the ER between sets and is a distance metric. Meanwhile, Section 6 describes a strategy to control the computational complexity of the ER calculation, using an ε-cover combined with a suitable scaling of the graph weights. Finally, section 7 provide several numerical experiments supporting our theoretical findings.

1.1 Related work

In our work, we show that our region-based ER can be thought of as ER between sets. An extension of the ER to ER between disjoint sets was introduced in Song et al. [15] for application on signed graphs. Song et al. [15] showed that ER between sets is a convex function of the edge weights. Furthermore, it was established that effective resistance between disjoint sets is monotonically increasing w.r.t. decreasing set size. Our contribution is to show that ER between sets can be generalized to ER over a metric space. We prove convergence to a non-trivial limit as the graph size increases and give numerical evidence that this limit is meaningful. Furthermore, we prove the triangle inequality and show that region-based ER is a distance metric.

2 Resistor Graphs

In this section, we review several key ideas and concepts about *resistor graphs* and the effective resistance. Finally, we introduce the notion of effective resistance between sets.

We start by formulating our setting. Let $G = (X, W)$ be an un-directed, weighted graph with nodes $X = \{x_1, \dots, x_n\}$, and edge weights $W_{i,j}$ and let $L = D - W$ be the graph Laplacian, where D is the degree matrix $D_{ii} = \sum_j W_{ij}$.

The graph can be thought of as an electrical network where each edge (x_i, x_j) has a non-negative *resistance* $R(x_i, x_j) = 1/W_{ij}$. With further analogy to electrical circuits, we can interpret a function $v : X \to \mathbb{R}$ on the graph's vertices as a voltage $v(x_i)$ and assign to each edge a signed *current* $J_{i,j} = -J_{j,i}$, which can be related to the resistance and voltages through Ohm's law. The relation can be written as

$$v(x_i) - v(x_j) = R(x_i, x_j) J_{i,j} \quad \text{or alternatively} \quad J_{i,j} = W_{ij}(v(x_i) - v(x_j)). \tag{2.1}$$

Meanwhile, from Kirchhoff's law, the sum of currents entering a node i must be zero, namely

$$\sum_{j \sim i} J_{i,j} = J_{ext,i}, \tag{2.2}$$

where $J_{ext,i}$ is an external current that can be either a source, a sink, or zero if the node is unconstrained (no external source applied). Combining these laws, we have that

$$(Lv)_i = \sum_{j \sim i} W_{ij}(v(x_i) - v(x_j)) = J_{ext,i} \tag{2.3}$$

Furthermore, since energy transfer in an electrical circuit is voltage times charge. We can define the *energy* of the voltage v as

$$E(v) \doteq \sum_{x_i,x_j \in X} W_{i,j}(v(x_i) - v(x_j))^2 = v^T L v \tag{2.4}$$

2.1 Effective resistance

The ER is a measure for calculating distances on graphs [16], which considers all possible paths between two nodes, as opposed to the shortest-path metric. As such, the ER captures the graph's structure more carefully, which can be advantageous in many applications. Using the analogy to electrical circuits, effective resistance can be defined in two equivalent ways based on $R(x_i, x_j) = \Delta V/J$, where ΔV is the voltage difference and J the total current flowing between the two nodes. In the **voltage difference formulation**, the current flow is constrained to unity, and the resistance is given in terms of the voltage difference between the nodes. In the **current flow formulation** the voltage difference is constrained to unity, and the resistance is given in terms of the inverse of the total current flow between the nodes. The following definitions formalize these approaches.

Definition 2.1 (Effective resistance (voltage difference formulation)). *The effective resistance $R(x_i, x_j)$ between two nodes $x_i, x_j \in X$ in a graph is the voltage difference between the nodes when a current of one ampere is injected between the source node x_i and extracted from the sink node x_j.*

Definition 2.2 (Effective resistance (current flow formulation)). *The effective resistance $R(x_i, x_j)$ between two nodes $x_i, x_j \in X$ in a graph is the inverse of the current flow between the nodes with the boundary conditions $v(x_i) = 1$ and $v(x_j) = 0$.*

From Definitions 2.1 and 2.2, we know that $R(x_i, x_j)$ can be found from the voltage or current in the system when the other is appropriately constrained. However, we still need a way to explicitly calculate these quantities. Several approaches exist for calculating the ER, and a summary of different formulations can be found in Theorem 4.2 in Jørgensen and Erin [17]. However, in this work we restrict ourselves to the two formulations that follow most naturally from Definitions 2.1 and 2.2 respectively.

Proposition 2.3 (Voltage difference formulation). *The effective resistance between nodes x_i, x_j corresponds to $R(x_i, x_j) = v(x_i) - v(x_j) = E_{min}$, where v is the function that minimizes the energy*

$$\min_v \quad \sum_{x_i,x_j \in X} W_{i,j}(v(x_i) - v(x_j))^2$$

$$\textit{Subject to} \quad (Lv)_i = 1, \quad (Lv)_j = -1, \quad (Lv)_i = 0, \forall i \in X \backslash \{x_i, x_j\}$$

Proof. See Theorem 4.2, Jørgensen and Erin [17]. □

Proposition 2.4 (Current flow formulation). *The effective resistance between nodes x_i, x_j corresponds to $R(x_i, x_j) = 1/J_{tot}$, where*

$$J_{tot} = \sum_{j \in X} W_{ij}(v(i) - v(j))$$

and v is the function that minimizes the Dirichlet energy

$$\min_v \quad \sum_{x_i,x_j \in X} W_{i,j}(v(x_i) - v(x_j))^2$$

$$\textit{Subject to} \quad v(x_i) = 1, \quad v(x_j) = 0$$

Proof. See Theorem 4.2, in Jørgensen and Erin [17]. □

We note that the standard formulation of effective resistance

$$R(x_i, x_j) = (e_i - e_j)^\top L^\dagger (e_i - e_j)$$

can easily be derived from $R(x_i, x_j) = v(x_i) - v(x_j) = (e_i - e_j)^\top v$ and the constraints $Lv = e_i - e_j$ in Proposition 2.3, which gives $v = L^\dagger(e_i - e_j)$. This relation shows how ER is an euclidean distance matrix [18]. Furthermore, it show how the distance between nodes using ER, can be computed without explicitly calculating the voltage functions v. Instead, it suffices to solve for $L^\dagger$, the pseudo-inverse of L. Methods have also been proposed for distributed computation [19].

Lemma 2.5. *Effective resistance satisfies the triangle inequality and is a distance metric on graphs*

Proof. See e.g. Jørgensen and Erin [17], Ghosh et al. [18] or Klein and Randić[16]. □

Remark 2.6. *We note that the* ***voltage difference formulation*** *corresponds to a Poisson problem since the constraints on the current are applied as an inhomogenous term on the Laplace equation $Lv = e_1 - e_2$. Similarly, the* ***current flow formulation*** *corresponds to a Dirichlet problem since it solves a homogenous Laplace equation with the constraints applied as boundary conditions.*

2.2 Effective Resistance Between Sets

In this study, we propose considering the ER between small regions rather than pairs of points to achieve a non-trivial limit in the metric graph setting. Our first step in this regard is to extend the concept of effective resistance to effective resistance between sets. Whereby sets, we mean subsets of the graph nodes X. The effective resistance between sets was defined in Definition 2 [15], in terms of the Schur complement of the Laplacian. For the analysis in this paper, we consider the current flow interpretation from Song et al. [15] and use this to write down an equivalent formulation that generalizes the current flow formulation from Proposition 2.4 to the effective resistance between sets. We consider the sets $X_a, X_b \subset X$ and $X_c = X\backslash(X_a \cup X_b \cup X_z)$ and let $R^s(X_a, X_b)$, defined in Definition 2.7, denote the effective resistance between two sets. Proposition 2.8 tells us how we can explicitly calculate $R^s(X_a, X_b)$.

Definition 2.7 (Effective resistance between sets (current flow formulation))**.** *The effective resistance $R^s(X_a, X_b)$ between two non-empty disjoint sets $X_a, X_b \subset X$ in a graph is the inverse of the current flow between the two subsets with the boundary conditions $v(x_i) = 1$, $\forall i \in X_a$ and $v(x_i) = 0$, $\forall i \in X_b$.*

Proposition 2.8 (Current flow formulation on sets)**.** *The effective resistance between the non empty disjoint subsets $X_s, X_g \subset X$ corresponds to $R^s(X_s, X_g) = 1/J_{tot}$, where*

$$J_{tot} = \sum_{i \in X_s} \sum_{j \in X} W_{ij}(v(i) - v(j))$$

and v is the function that minimizes the energy

$$\begin{aligned} \min_v \quad & \sum_{x_i, x_j \in X} W_{i,j}(v(x_i) - v(x_j))^2 \\ \textit{Subject to} \quad & v(x_i) = 1 \, \forall i \in X_s, \quad v(x_i) = 0, \, \forall i \in X_g \end{aligned}$$

Proof. Definition 2 and Theorem 1 in Song et al. [15]. □

3 Effective Resistance over metric spaces

Our goal is to extend the concept of effective resistance to metric spaces. To this end, we start by defining a particular type of graph, namely graphs constructed from samples drawn from a distribution over a metric space. Let (M, d) be a compact metric space and μ, a probability measure over M.

Let $k : M \times M \to [0, 1]$ be a kernel function that defines what it means for two points to be ”near” each other. Commonly used kernel functions are:

- The radial kernel: $k_r(x, y) = \mathbb{1}(d(x, y) \leq r)$ where $r > 0$ is some fixed radius.

- The Gaussian kernel: $k_\sigma(x, y) = \exp(\frac{-d(x,y)^2}{2\sigma^2})$, where $\sigma > 0$ is the fixed temperature parameter.

Next, let $M_s \subseteq M$ and $M_g \subseteq M$ be two disjoint measurable subsets of M. As before, we let $k : M \times M \to [0, 1]$ be a kernel function.

To define the effective resistance between M_s and M_g, we begin by defining the energy-minimizing voltage induced by μ and k over these sets.

Definition 3.1. *Let $M_s, M_g \subset M$ be measurable disjoint subsets, k a kernel function, and μ a probability measure over M. Let V_{M_s,M_g} be the set of all measurable functions $v : M \to [0, 1]$ with $v(x) = 1$ for all $x \in M_s$ and $v(x) = 0$ for all $x \in M_g$. For any such v, define its energy as*

$$E(v) = \int_M \int_M k(x, y)(v(x) - v(y))^2 d\mu(x)d\mu(y).$$

Then we say that $v^* \in V_{M_s,M_g}$ is an energy minimizing voltage if it is a global minimum of $E(v)$, meaning that

$$v^* = \operatorname*{argmin}_{v \in V_{M_s,M_g}} E(v).$$

Observe that Definition 3.1 does not necessarily imply a unique energy minimizing voltage induced by M_s and M_g. However, under certain technical conditions on k, μ, and M, we can show that uniqueness holds, which will be crucial for defining the effective resistance between M_s and M_g.

3.1 Technical conditions of M, μ, and k

We begin with the notion of a normalized kernel, which integrates to 1 over any fixed point $x \in M$.

Definition 3.2. *The normalized kernel of k, denoted $\hat{k} : M \times M \to R$, is defined as*

$$\hat{k}(x, y) = \frac{k(x, y)}{\int_M k(x, z)d\mu(z)}.$$

A normalized kernel can be thought of as the natural analog of a degree normalized weight matrix. We can easily verify that $\int \hat{k}(x, y)d\mu(y) = 1$ for all x.

Next, we generalize the notion of *adjacency* to the metric setting.

Definition 3.3. *Let $A \subseteq M$ a measurable set and $\alpha > 0$. Define $C_\alpha(A)$ as the set of all x such that $\int_A \hat{k}(x, y)d\mu(y) > \alpha$.*

$C_\alpha(A)$ can be thought of as a continuous generalization of the set of "neighbors" of A. In the finite setting $C(A)$ would comprise of vertices that are adjcaent to some vertex in A. The parameter α provides a lower bound on the degree of connectivity.

Next, we continue this to develop an analogous generalization of vertices that are path connected to A. To do so, we first define the convolution operation over kernel similarity functions.

Definition 3.4. *Let p, q be kernel functions $p : M \times M \to [0, 1]$, $q : M \times M \to [0, 1]$. Then their convolution is the function $p \circ q : M \times M \to [0, 1]$ defined by*

$$(p \circ q)(x, y) = \int_M p(x, z)q(z, y)d\mu(z).$$

We let $p^{(i)}$ denote $p \circ p \circ \ldots p$ repeated i times.

The key idea is that the kernel $\hat{k}^{(i)}$ corresponds to paths of length i. As a result, we now define C^i_α accordingly.

Definition 3.5. *Let $A \subseteq M$ a measurable set. Then $C_\alpha^i(A)$ denotes the set of all x for which $\int_A \hat{k}^{(i)}(x,y)d\mu(y) \geq \alpha$.*

We will now restrict our interest to metric spaces M for which $M = C_\alpha^m(A)$ for some fixed integer $m > 0$, and real number $\alpha > 0$. This condition essentially means that the graphs we consider are both path connected and have bounded diameter, and the parameter α implies that this connection has a degree of robustness. We note that these assumptions are extremely mild: for example, any compact manifold using a radial or Gaussian kernel can be easily shown to satisfy them.

3.2 Defining the effective resistance

We now show that under the previously discussed technical conditions, there exists a unique energy minimizing voltage with respect to M_s, M_g, k, and μ.

Proposition 3.6. *Let k be a kernel, M, a metric space, and μ a measure over M. Let M_s and M_g be measurable disjoint subsets of M. Suppose there exists $\alpha > 0$, $m > 0$ such that $M = C_\alpha^m(M_g \cup M_s)$ (Definition 3.5). Then for any measurable disjoint sets M_s and M_g, there exists a unique energy minimizing voltage v^* (Definition 3.1) over M with M_s as its source and M_g as its sink.*

Given a unique energy-minimizing voltage, we can then define the effective resistance between M_s and M_g as the inverse of the induced current between them by v^*.

Definition 3.7 (Effective resistance on metric space)**.** *Let M_s, M_g be non-empty measurable disjoint subsets of M such that they have a unique energy minimizing voltage, v^*, with respect to measure μ and kernel k. Then their effective resistance is defined as $R^\mu(M_s, M_g) = 1/J_{tot}$, where*

$$J_{tot} = \int_{M_s} \int_M k(x,y)(v^*(x) - v^*(y))d\mu(x)d\mu(y).$$

4 Convergence towards the effective resistance

In this section, we show how the effective resistance between two subsets of metric space M, M_s, and M_g, can be approximated by computing the effective resistance of a graph induced by points sampled from the probability measure, μ, over M. To this end, let $X_n = \{x_1, x_2, \ldots, x_n\} \sim \mu$ be a set of points sampled from data distribution μ over M. Using one of the kernels defined above, a weighted graph can be constructed on the samples by assigning weights $W_{ij} \propto k(x_i, x_j)$ between each pair of points $x_i, x_j \in X_n$.

Definition 4.1 (Metric resistor graph)**.** *Let X_n a sample as defined above and $k : M \times M \to [0,1]$ be a kernel similarity function. Then the* ***metric resistor graph****, $G_n = (X_n, W)$, is the weighted graph with edge weights $W_{ij} = \frac{k(x_i,x_j)}{n^2}$.*

In the next section, we take a closer look at the scaling factor $1/n^2$.

4.1 Scaling

Our goal is to construct a definition of the effective resistance that converges towards a non-trivial solution as the number of sampled points, n, goes towards infinity. To achieve this, it is necessary that the edge weights W_{ij} scale appropriately with the number of samples. It is natural to demand that the physical properties of the graph, embodied by the resistance, current, and voltage, should remain relatively stable as n increases. Thus, it is crucial to understand how the edge resistances should scale with the number of points sampled.

Intuition To this end, consider two small regions T and T' such that $k(x,x') > 0$ for all $x \in T$ and $x' \in T'$. For simplicity, let $k(x,x')$ be constant, which means each edge has equal resistance $R = 1/k(x,x')$. We aim to keep the resistance between these two regions constant as the number of points changes. For a fixed X_n, on average there are m points $x \in S_n = \{x \in X_n : x \in T\}$ and m points $x' \in S'_n = \{x \in X_n : x \in T'\}$. This results in m^2 edges between T and T'. This means the total resistance between these regions is R/m^2, given that these edges are connected in parallel.

The issue here is that, once we move to a denser sample X_{2n}, there will be, on average, $2m$ points in S_{2n} and S'_{2n} respectively. This will create a net resistance $\frac{1}{4}R/m^2$, which means the resistance between these physical regions T, T' decreases and will go to 0 as n goes to infinity. We illustrate this construction in Figure 2.

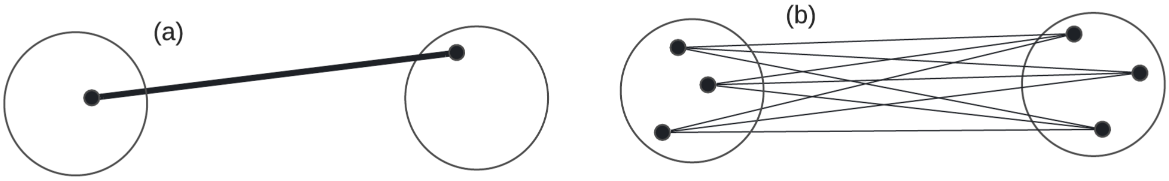


Figure 2: Example of resistance scaling. **(a)** Number of edges connecting T and T' for $m = 1$ point sampled from each region. **(b)** Number of edges connecting T and T' for $m = 3$ points sampled from each region. Notice that the number of edges between these regions is m^2.

Point-wise scaling In order to overcome this issue of decreasing net resistance, we introduce a point-wise scaling $\gamma_{ij} = 1/n^2$ for all node pairs i, j, to compensate for the m^2 factor. Here $m = pn$ where p is the probability that a sample drawn from X_n falls in region T. In other words, with the point-wise scaling, we have the net resistance $\frac{1}{4}R/pp'$, where p, p' are constant with respect to n.

Having defined the proper scaling, we now define the random object, v_n^*, which is the energy-minimizing voltage induced by a metric graph constructed from an i.i.d sample of n points from M.

Definition 4.2 (Energy minimizing voltage over metric graph). *Let M_s, M_g be disjoint measurable sets in M. For $X_n \sim \mu^n$, let $v_{X_n} : X_n \to [0,1]$ be defined as the energy minimizing voltage over the metric graph G_n where the weights W_{ij} are defined by $W_{ij} = \frac{1}{n^2}k(x_i, x_j)$. Then $v_n^* : M \to [0,1]$ denotes the function*

$$v_n^*(x) = \begin{cases} 1 & x \in M_s \\ 0 & x \in M_g \\ \frac{\sum_{x_i \in X_n} k(x,x_i) v_{X_n}(x_i)}{\sum_{x_i \in X_n} k(x,x_i)} & x \in M \setminus (M_s \cup M_g) \end{cases}.$$

Note that v_n^* represents a random variable over functions $M \to [0,1]$ with randomness induced by the randomness of X_n.

We similarly define the region-based ER, $R_n(M_s, M_g)$ as follows.

Definition 4.3 (Region-based effective resistance). *Let M_s, M_g be disjoint measurable sets in M. For $X_n \sim \mu^n$, let v_n^* be defined as the energy minimizing voltage over the metric graph, G_n where the weights W_{ij} are defined by $W_{ij} = \frac{1}{n^2}k(x_i, x_j)$. We then define the region-based ER $R_n(M_s, M_g) = 1/J_{tot}$ where J_{tot} is as defined in Proposition 2.8 with the voltage v_n^*. Note that $R_n(M_s, M_g)$ corresponds to a random variable, denoted R_{X_n}, induced by the randomness of X_n.*

4.2 Convergence analysis

We now show that the finite sample energy minimizing voltage, v_n^*, and the finite sample region-based ER, $R_n(M_s, M_g)$, converge towards the limit objects v^* and $R^\mu(M_s, M_g)$.

Theorem 4.4. *Let M, k, and μ satisfy that there exists $m > 0$ and $\alpha > 0$ such that $M = C_\alpha^m(M_g \cup M_s)$. Let M_s, M_g be disjoint measurable subsets of M, and let $v_n^*, v^*, R_n(M_s M_g)$, and $R^\mu(M_s, M_g)$ be as defined in Definitions 4.2, 3.1, 4.3, and 3.7. Then the following hold:*

1. *For any $x \in M$, the sequence $v_1^*(x), v_2^*(x), \ldots$ converges to $v^*(x)$ in probability.*
2. *The sequence $R_1(M_s, M_g), R_2(M_s, M_g), \ldots$ converges to $R^\mu(M_s, M_g)$ in probability.*

A proof can be found in Appendix B.

5 Properties of the effective resistance between sets

We have established that the finite sample region-based ER $R_n(M_s, M_g)$ from Definition 4.3 converges to the ER on a metric space $R^\mu(M_s, M_g)$, defined in Definition 3.7. In this section, we establish that $R_n(M_s, M_g)$ is a distance metric. In particular, we prove that it satisfies the triangle inequality.

Consider a metric space M and a sample $X_n \sim \mu^n(M)$. For two disjoint measurable subsets $M_s, M_g \subset M$ we can denote the corresponding subsets of X_n as $X_s = \{x \in X_n : x \in M_s\}$ and $X_g = \{x \in X_n : x \in M_g\}$. The region-based ER $R_n(X_s, X_g)$ over a finite sample can be thought of as the ER between sets, namely $R^s(X_s, X_g)$ as defined in Section 2.2. For the analysis in this section, this is the interpretation we will take.

In the following, we will introduce the subscript G on $R_G^s(X_a, X_b)$ to indicate the graph over which the ER is calculated. From Lemma E.3 it follows that $R_G^s(X_a, X_b) = R_{G_{ab}}^s(a, b) = (e_1 - e_2)^\top L_{G_{ab}}^\dagger (e_1 - e_2)$ where $L_{G_{ab}}$ is the Laplacian on the reduced graph G_{ab} defined in Definition E.1. Symmetry and positive semi-definiteness of $R_G^s(X_a, X_b)$ follows therefore from the classical result on the reduced graph G_{ab} [17, 18, 16].

Meanwhile, the triangle inequality is established by Theorem 5.1.

Theorem 5.1. *(Triangle inequality for effective resistance between sets) Consider a graph $G = (X, W)$ and the non-empty disjoint subsets $X_a, X_b, X_z \in X$. Let $R^s(X_p, X_q)$ be the ER between sets X_p, X_q for $p, q \in \{a, b, z\}$. We then have*

$$R^s(X_a, X_b) \leq R^s(X_a, X_z) + R^s(X_z, X_b)$$

Proof. Appendix E.2. □

Having established the triangle inequality along with symmetry and positive semi-definiteness, it follows that the effective resistance between disjoint sets is a distance metric.

6 Computational considerations

In practice, calculating the effective resistance in the large graph limit is not computationally viable. This is because the computational cost directly increases with n, the number of sampled points. For large values of n, it can be relatively expensive to compute the effective resistance – especially if we desire to do so for many pairs of regions M_s, M_g. To control the computational complexity of the calculation, we suggest building the graph on a partitioning of the data which we call an α-cover $\mathcal{C}$ to remove the dependency on n. Furthermore, in order to incorporate information about the density, we combine the α-cover with a suitable scaling of the graph weights.

6.1 Alpha-cover

Let (M, d) be a metric space and conisder an α-cover $\mathcal{C}$ as defined in Definition 6.1. The α-cover is closely related to the concept of doubling dimension $\mathsf{ddim}(M)$ as defined in Definition A.1, which is a measure of the intrinsic dimension of M. In fact, if $\mathcal{B}(\eta)$ is the smallest ball such that $M \subseteq \mathcal{B}(\eta)$ and $\alpha = 2^{-\ell}\eta$, then it follows that $|\mathcal{C}| = 2^{\ell \mathsf{ddim}(M)}$. This means that the size of the α-cover depends only on the resolution, span of the data, and the intrinsic dimension of M, while it is independent of n.

Definition 6.1 (α-cover)**.** *Let (M, d) be a metric space. A subset $\mathcal{C} \subseteq M$ is an α-cover of M if it satisfies the following two conditions.*

1. *(Packing property): All points $y_1, y_2 \in \mathcal{C}$ are such that $d(y_1, y_2) \geq \alpha$*
2. *(Covering property): For all points $x \in M$ there exists $y \in \mathcal{C}$ such that $d(x, y) \leq \alpha$*

To construct the α-cover, a popular algorithm is the cover-tree algorithm, see e.g. Beygelzimer et al. [20] and Oslandsbotn et al. [21] (which is the algorithm we use in this paper).

Region-wise scaling In order to incorporate information about the underlying distribution $\mu(M)$, we combine the α-cover with a region-wise scaling as an alternative to the scaling suggested in Section 4.1. This is important because, as mentioned in the introduction, one of the desirable properties of ER is that it can capture cluster structures in graphs, which in turn means information about the underlying data distribution $\mu(M)$. Because this information is lost with the α-cover, due to the uniform partitioning, a scaling that captures the density is necessary.

To formally define this scaling, we first introduce the concept of a Voronoi cell

Definition 6.2 (Voronoi cell)**.** *Let $\mathcal{C}$ be an α-cover as defined in Definition 6.1. We define the Voronoi cell associated with point $x_i \in \mathcal{C}$ as*

$$\mathcal{V}_i = \{x \in M : d(x_i, x) \leq d(x_j, x) \, \forall x_i, x_j \in \mathcal{C}\}$$

We can then define the scaling γ_i for a given sample $X_n \sim \mu^n(M)$.

Definition 6.3 (Region-wise scaling)**.** *Let X_n be a sample as defined above and $\mathcal{C}$ the associated α-cover. With each center, $x_i \in \mathcal{C}$, we associate a scaling constant*

$$\gamma_i = \frac{|\{x \in X_n : x \in \mathcal{V}_i\}|}{n}$$

The interpretation of γ_i is that it is the empirical local probability density associated with each Voronoi cell in the α-cover. It can easily be estimated by counting the number of samples $x \in X_n : x \in \mathcal{V}_i$, divided by the total number of samples n.

Remark 6.4. *We note that for the α-cover approach, we can not use the point-wise scaling introduced in Section 4.1. To see this, consider the discussion in Section 4.1, which showed that the net resistance between two regions T and T' is $\frac{1}{4}R/m^2$. For data sampled from μ, it follows that $m \propto n$. However, for the α-cover, we have instead $m \propto |\mathcal{C}|$, and the size of the epsilon cover $|\mathcal{C}|$ depends only on the doubling dimension of the data and not on n.*

6.2 Effective resistance on α-cover

Our idea is to construct a graph on the α-cover, instead of directly on X_n, and then define the ER on this graph instead. We call this new graph the Cover resistor graph.

Definition 6.5 (Cover resistor graph)**.** *Let $\mathcal{C}$ be an α-cover, and let X_n a sample as defined above and $k : M \times M \to [0, 1]$ be a kernel similarity function. Let γ_i, γ_j be the region-wise scaling weights defined in Definition 6.3. Then the **cover resistor graph**, $G_n^{\mathcal{C}} = (\mathcal{C}, W)$, is the weighted graph with edge weights $W_{ij} = \gamma_i \gamma_j k(x_i, x_j)$.*

The cover resistor graph essentially uses the nodes of the cover as vertices and constructs weights based on the relative *weights* of each cover node.

We now show that computing effective resistance over the cover resistor graphs converges to the same limit object that using the sampled metric graphs in the previous section does. To do so, we begin by defining analogs of v_n^* and $R_n(M_s, M_g)$.

Definition 6.6 (Energy minimizing voltage over α-cover). *Let $\mathcal{C}$ be an α-cover, and $G_n^{\mathcal{C}}$ be its cover resistor graph constructed from sample X_n, source and sink regions M_s and M_g, and kernel function k. Let $v_{X_n,\mathcal{C}} : \mathcal{C} \to [0,1]$ denote the energy minimzing voltage function over $G_n^{\mathcal{C}}$. Then we let $v_n^{\mathcal{C}} : M \to [0,1]$ be defined as*

$$v_n^{\mathcal{C}}(x) = \begin{cases} 1 & x \in M_s \\ 0 & x \in M_g \\ \frac{\sum_{c_i \in \mathcal{C}} k(x,c_i) v_{X_n}(x_i)}{\sum_{c_i \in \mathcal{C}} k(x,c_i)} & x \in M \setminus (M_s \cup M_g) \end{cases}.$$

Note that $v_n^{\mathcal{C}}$ represents a random variable over functions $M \to [0,1]$ with randomness induced by the randomness of X_n.

Definition 6.7 (Region-based ER on α-cover). *Let $\mathcal{C}$ be an α-cover, let $v_n^{\mathcal{C}}$ be the associated energy minimizing voltage and let $G_n^{\mathcal{C}}$ be the cover resistor graph from Definition 6.5 constructed from sample $X_n \sim \mu^n$. Consider the source and sink regions M_s and M_g. We define the region-based ER on the α-cover as $R_n^{\mathcal{C}}(M_s, M_g) = 1/J_{tot}$ where J_{tot} is as defined in Proposition 2.8 but now with respect the sets $\mathcal{C} \cap M_s, \mathcal{C} \cap M_g$, the graph $G_n^{\mathcal{C}}$ and the voltage $v_n^{\mathcal{C}}$. Note that $R_n^{\mathcal{C}}(M_s, M_g)$ denotes the random variable, $R_{X_n}^{\mathcal{C}}$, induced by the randomness of X_n.*

We now show that similarly to v_n^* and $R_n(M_s, M_g)$, for sufficiently small values of α, $v_n^{\mathcal{C}}$ and $R_n^{\mathcal{C}}(M_s, M_g)$ converge to the same quantities.

Theorem 6.8. *Let M be a metric space, k a kernal similarity function, and μ be a measure over M. Let M_s and M_g be two disjoint measurable subsets of M such that $M = C_\beta^m(M_s \cup M_g)$ for some $m, \beta > 0$. Then there exists a function $\Delta : \mathbb{R}^+ \to \mathbb{R}^+$ such that the following properties hold. First, $\lim_{\alpha \to 0^+} \Delta(\alpha) = 0$. Second, for any $\alpha > 0$ and any α-cover of M, $\mathcal{C}$, the following two conditions hold.*

1. *For any $x \in M$, the sequence $v_1^{\mathcal{C}}(x), v_2^{\mathcal{C}}(x), \ldots$ converges in probability, and satisfies that*
$$|\lim_{n\to\infty} v_n^{\mathcal{C}}(x) - v^*(x)| < \Delta(\alpha).$$

2. *The sequence $R_1^{\mathcal{C}}(M_s, M_g), R_2^{\mathcal{C}}(M_s, M_g), \ldots$ converges in probability and satisfies that*
$$|\lim_{n\to\infty} R_n^{\mathcal{C}}(M_s, M_g) - R^\mu(M_s, M_g)| < \Delta(\alpha).$$

This result implies that for a sufficiently small value of α, we can essentially replace our sample X_n with any α-cover, $\mathcal{C}$, of M.

6.3 A note on the advantages of using an α-cover

The advantages of defining the ER on an α-cover are two-fold:

1. With the α-cover, the size of the graph and, therefore, the computational complexity of computing the ER can be controlled independently of n.

2. The approximation of the density can be refined in a continuous manner by updating the local probabilities using more samples from $\mu(M)$, without the need to change the size of the graph.

For example, with $\alpha = 2^{-l}\eta$, it follows that $|\mathcal{C}|$ and, therefore, the graph size grows as $\mathcal{O}((\eta/\alpha)^{\mathsf{ddim}(M)})$. This means that the size of the α-cover depends only on the resolution we want α, the span of the data η, and the doubling dimension $\mathsf{ddim}(M)$ of M (Definition A.1).

We note that the α-cover, with region-wise scaling, satisfies the criteria for a streaming algorithm [22], allowing for continuous refinement of the graph weights, without increasing the size of the graph. The problem is that calculating the effective resistance between sets requires solving the Schur complement with respect to each set, which is computationally expensive; see Section A.1.

Remark 6.9. *We note that the use of α-cover and region-wise scaling is not restricted to the region-based ER proposed in this paper. It can also be used for computing the standard ER.*

7 Experiments

In this section, we demonstrate the region-based effective resistance (region-based ER) in the large graph limit and show that it converges to a meaningful limit. We divide the section into three sets of experiments. In Section 7.1, we replicate some of the experiments conducted in Von Luxburg et al. [13] and show that the region-based ER does not suffer from the trivial limit issue that standard ER suffer from. In Section 7.2 we take this a step further and demonstrate the convergence of region-based ER to a meaningful limit. Finally, in Section 7.3, we demonstrate a computationally efficient way to extend the calculations to large graphs in a controlled manner.

We note that to calculate the region-based ER, we utilize the Schur complement of the graph Laplacian with respect to these sets; see Appendix A.1 for more details.

Setting: Throughout this section, we consider standard ER and region-based ER on data sets $X_n \sim \mu^n(M)$, sampled from a distribution over a metric space (M, d). For the standard ER we consider a resistor graph as described in Section 2. For the region-based ER we use a metric graph as defined in Definition 4.1. The standard ER between two nodes x_i, x_j in the graph is denoted $R_{ij} \coloneqq R(x_i, x_j)$. For the region-based ER, we introduce the notation $R^s_{ij} \coloneqq R_n(X_i, X_j)$. We use a radial kernel to determine the sets X_j, X_i associated with x_i, x_j. Namely $X_i = \{x \in X_n : \mathbb{1}(d(x, x_i) \leq r)\}$ and similarly for X_j. Here r_s is the source radius (which varies for each experiment).

7.1 Region-based ER does not converge to the Von-Luxburg limit

In this section, we replicate some of the experiments conducted in Von Luxburg et al. [13]. For the experiments, we calculate both the standard ER and the region-based ER proposed in this paper. We show that where the standard definition of ER converges to a trivial limit as shown by Von Luxburg et al. [13], the region-based ER does not.

We are interested in the convergence of the region-based ER compared to the standard ER as the number of samples used to construct the graph increases. We construct the graph on a sample X_n. Let $\eta_{ij} \coloneqq 1/D_i + 1/D_j$ be the Von-Luxburg limit for the standard ER; similarly, let $\eta^s_{ij} \coloneqq 1/D^s_i + 1/D^s_j$ be the Von-Luxburg limit for the region-based ER. Here D_i, D_j are the degrees of nodes i, j, respectively. Similarly, D^s_i, D^s_j are the degrees associated with the sets X_i, X_j, see Appendix E.1 and Lemma E.2 for more details. We then consider the max and mean of the relative deviation from the Von-Luxburg limit, namely:

$$\max_{ij} |R_{ij} - \eta_{ij}|/R_{ij} \quad \text{and} \quad \hat{\mathbb{E}}[|R_{ij} - \eta_{ij}|/R_{ij}]. \tag{7.1}$$

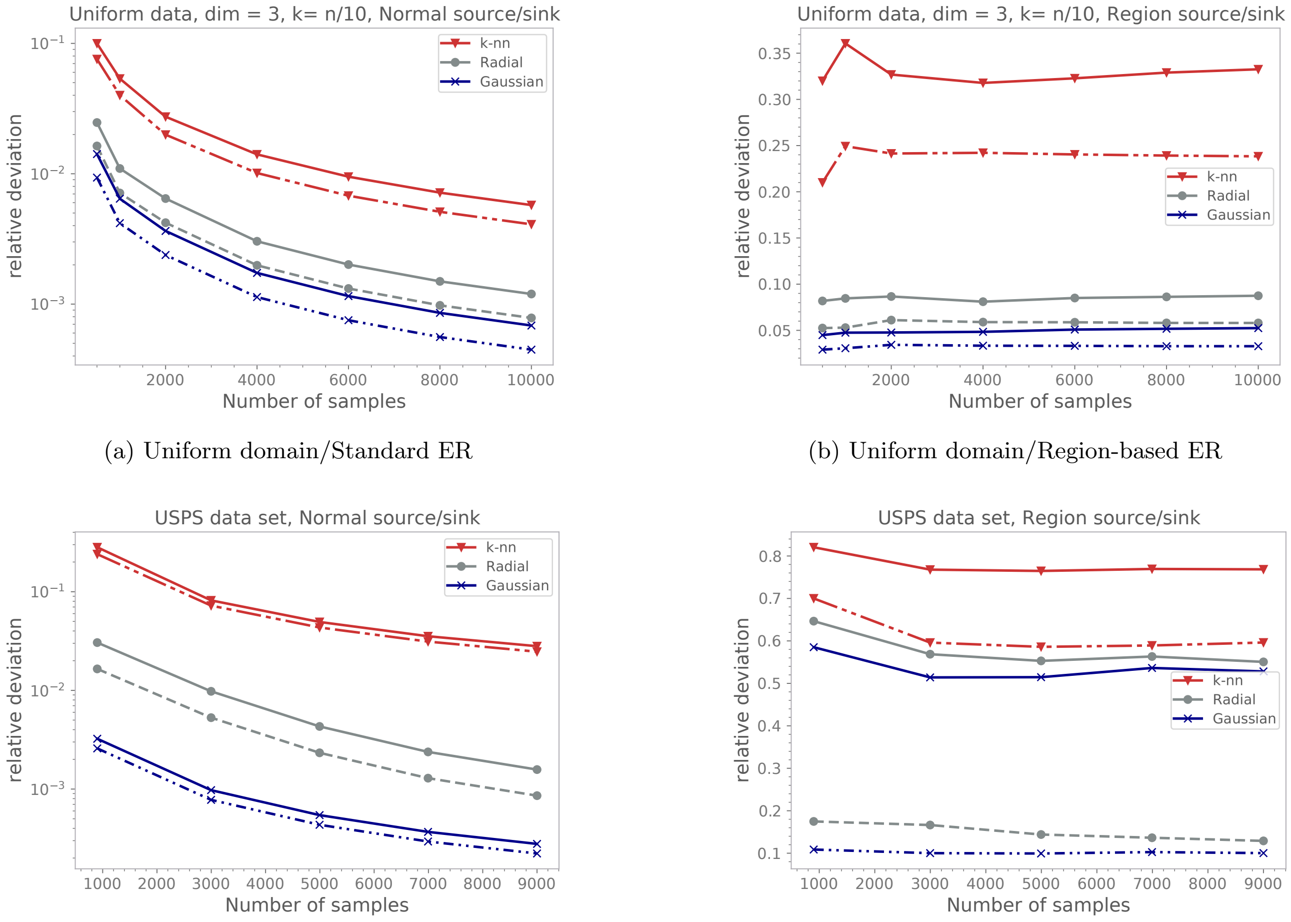
and

$$\max_{ij} |R^s_{ij} - \eta^s_{ij}|/R^s_{ij} \quad \text{and} \quad \hat{\mathbb{E}}[|R^s_{ij} - \eta^s_{ij}|/R^s_{ij}]. \tag{7.2}$$

We consider the convergence of the relative deviation from the Von-Luxburg limit on two data sets that are similar to those studied in [13]. These are:

- ◇ Uniform 3-dim domain
- ◇ USPS data-set of handwritten digits (roughly 9200 samples in 256 dimensions)

On each data set, we build a graph using the kernels outlined in section 3 and also include the nearest neighbor kernel, $\Gamma(x, y) = \mathbb{1}(y \in \mathcal{N}_\kappa(x)) \vee (x \in \mathcal{N}_\kappa(y))$ to better replicate the corresponding experiments in Von Luxburg et al. [13]. Similarly to these experiments, we also select the radius of the radial kernel and the bandwidth of the Gaussian kernel to be the maximal k-nn distance in the data. Note that for the USPS data set, we let $k = 100$, and for the uniform data set, we let $k = n/100$ where n is the number of samples. We let the source radius be the maximal 20-nn distance in the data.



(a) Uniform domain/Standard ER

(b) Uniform domain/Region-based ER

(c) USPS data set/Standard ER

(d) USPS data set/Region-based ER

Figure 3: This figure demonstrates that standard ER converges to the Von-Luxburg limit, while region-based ER does not (which is a good thing). Here solid lines show maximal relative deviations, and dashed lines show mean relative deviations. The x-axis shows the number of samples used to construct the graph. The y-axis is the relative deviation of the effective resistance from the Von-Luxburg limit.

Uniform domain The results for the uniform domain are shown in Figure 3a-3b. The figure shows the max and mean relative deviation from the Von-Luxburg limit for the standard ER (Figure 3a) and the region-based ER (Figure 3b). We see that the standard ER converges quickly to the Von-Luxburg limit, which corresponds to the results in Von Luxburg et al. [13]. Meanwhile, we see that for the region-based ER, convergence to the Von-Luxburg limit does not occur, which confirms our theoretical results.

USPS data set A similar set of experiments is shown for the USPS data set in Figure 3c-3d. We observe the same behavior as for the uniform domain. The standard ER converges quickly to the Von-Luxburg limit, while the region-based ER does not converge to this limit. Again, this confirms our theoretical findings with respect to the region-based ER.

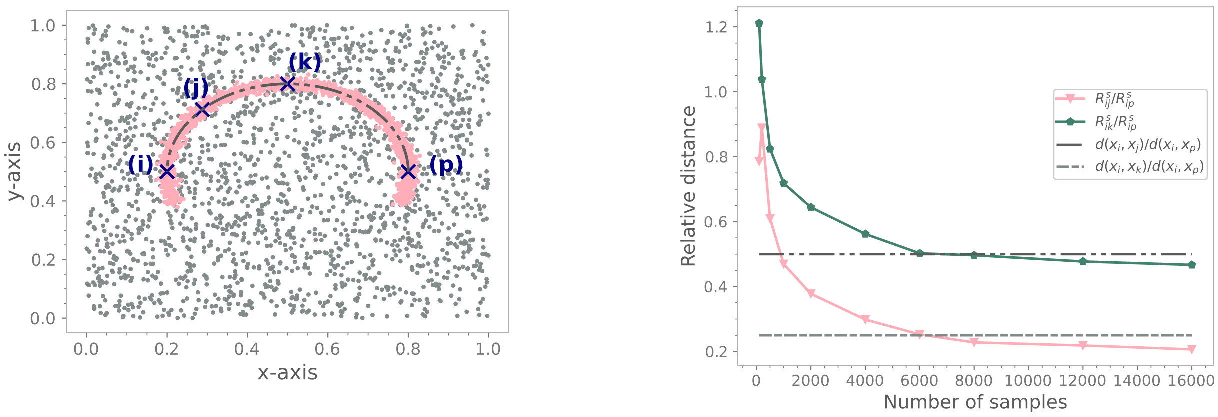
7.2 Meaningful limit

In the previous section, we saw that the region-based ER does not converge to the Von-Luxburg limit as was the case for the standard ER. However, it remains to be shown that the effective resistance under this new definition converges towards a meaningful limit. In this section, we take a closer look at this by considering the following two experiments

- ◇ Convergence to a meaningful limit on a half-moon of increasing density
- ◇ Meaningful ordering of points on a Swiss roll.

Half-moon experiment We consider a data distribution that consists of a background $[0,1]^2$ with low density (10000 samples) and a half-moon with increasing density (samples in the interval $[100,\dots,16000]$). This point cloud is illustrated in Figure 4a.

The purpose of this experiment is to demonstrate that the region-based ER converges to the "geodesic distance" along the half-moon (see the grey dotted line in Figure 4a), as the number of samples on the half-moon increases. This is a meaningful limit since the ER-based distance should consider all paths, and as the density of the half-moon becomes increasingly dominant, the distance should converge to the distance of paths along this curve.



(a)

(b)

Figure 4: The region-based ER between points on the half-moon converges to the distance along the half-moon. (This is what we want to see). (a) High-density half-moon (pink) over low-density background (grey). The points we consider are labeled i, j, k, and p. (b) Grey dotted lines shows Γ_{ijp} and Γ_{ikp} (See Eq. (7.3)). Pink line shows R^s_{ij}/R^s_{ip} and Green line shows R^s_{ik}/R^s_{ip}.

The half-moon we consider is sampled from a circle segment with radius $t = 0.3$ and angle $\theta \in [-20, 200]^\circ$ with a Gaussian distribution $\mathcal{N}((t,\theta), 0.01)$ for each θ. Along the half-moon, we consider four points x_i, x_j, x_k, and x_p; placed respectively at $\theta_i = 0^\circ$, $\theta_j = 45^\circ$, $\theta_k = 90^\circ$, and $\theta_p = 180^\circ$. We construct a graph on the point cloud using a radial kernel with radius $r = 0.08$. For the source and sink regions, we use a source radius $r_s = 0.05$ centered on each source-sink point, respectively.

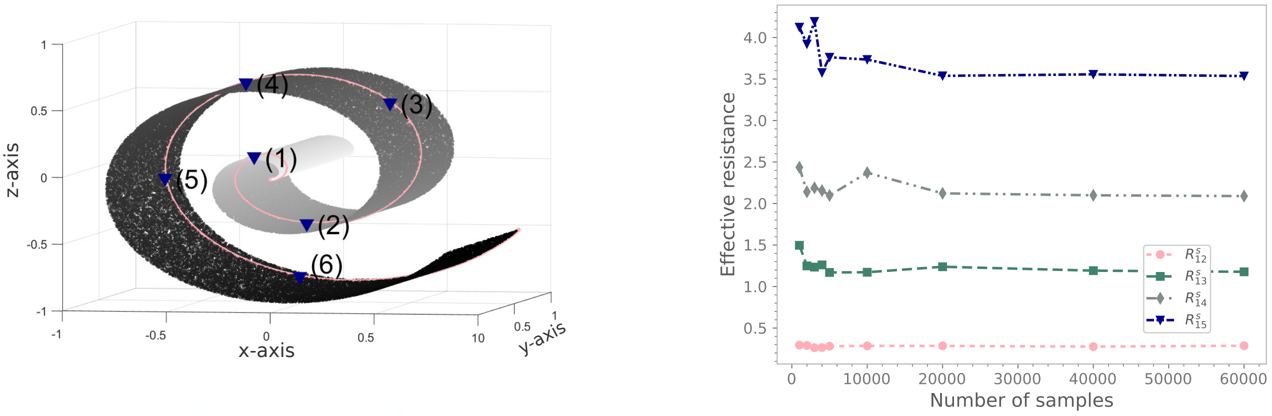
Let $d(\cdot,\cdot)$ denote the distance along the half-moon arch as illustrated in Figure 4a by the grey dotted line. In order to compare the region-based ER distance with $d(\cdot,\cdot)$ we need to consider these distances relative to some reference distance. Because of this, we introduce the distance between points i and p as the reference. We then have

$$\Gamma_{ijp} := d(x_i, x_j)/d(x_i, x_p) = 0.25 \quad \text{and} \quad \Gamma_{ikp} := d(x_i, x_k)/d(x_i, x_p) = 0.5 \tag{7.3}$$

Figure 4b shows the region-based ER ratios R^s_{ij}/R^s_{ip} and R^s_{ik}/R^s_{ip} as the number of points sampled from the half moon increases. The ratios R^s_{ij}/R^s_{ip} and R^s_{ik}/R^s_{ip} converges to values close to Γ_{ijp} and Γ_{ikp} respectively. This is expected because, as the density on the half-moon increases, the effective resistance, which considers all possible paths, should be increasingly dominated by the paths along the half-moon. Note that the convergence is not expected to coincide exactly with Γ_{ijp} and Γ_{ikp} due to the background and the width of the half-moon.

Remark 7.1. *We note that had the region-based ER converged to the Von-Luxburg limit, we would not have observed the convergence in Figure 4b. This is because the Von-Luxburg limit only depends on the* *degrees of the respective sets, which for the half-moon would be the same for all points i, j, k, p. Therefore, one would expect R^s_{ij}/R^s_{ip} and R^s_{ik}/R^s_{ip} to converge to 1 if this was the case.*

Swiss roll experiment We consider a data distribution shaped as a Swill roll and compare the relative distance between five points along the Swiss roll surface as indicated in Figure 5a. We consider the source radius $r_s = 0.1$ centered at the point and use a radial kernel with radius $r = 0.2$ to construct the graph.



(a)

(b)

Figure 5: The ordering of the lines in (b) is meaningful (which is a good thing). Experiment demonstrating that the region-based ER distance gives a meaningful ordering of the distance between points and maintains this as the number of samples used to construct the graph increases. (a) Data distribution. Blue triangles indicate the location of the points $1, \dots, 6$. (b) The y-axis shows the region-based ER scaled by a factor of 10^{-5}. The x-axis shows the number of samples used to construct the graph. The pink line corresponds to the region-based ER R^s_{12} between $1, 2$. Similarly, for R^s_{13} (green), R^s_{14} (gray), and R^s_{15} (blue).

The region-based ER R^s_{1i} between point 1 and respectively points $i \in [2, 3, 4, 5]$ is shown in Figure 5b. Using R^s_{1i} as a measure of distance between the points, we see that region-based ER gives a natural ordering of the distance from 1 to the other points, which is maintained as the number of samples increases. Namely, that $R^s_{12} < \cdots < R^s_{15}$.

Remark 7.2. *We note that had the region-based ER converged to the Von-Luxburg limit, this ordering would not have been maintained for the same reason as discussed in the half-moon experiment; See remark 7.1.*

7.3 Example on the benefit of using an α-cover graph

We include a simple experiment to demonstrate the use of an α-cover and region-wise scaling, which was introduced in Section 6.1. We consider data sampled from a non-uniform density $\mu([0, 1])$ consisting of two regions of high density separated by a region of low density (See Figure 6). We consider five points $\mathcal{J} = \{1, \dots, 5\}$ marked by blue triangles in Figure 6 and calculate the region-based ER R^s_{1j} between 1 and points $j \in \mathcal{J} \backslash \{1\}$. In the experiment, the source radius is $r_s = 0.1$, and we use a radial kernel with radius $r = 0.1$.

We calculate the region-based ER using two different graphs;

(A) Graph built on an α-cover with $\alpha = 2/3 \times 3^{-6}$ (1122 centers) and with region-wise scaling

(B) Graph built with samples directly from $\mu([0, 1])$ and with point-wise scaling.

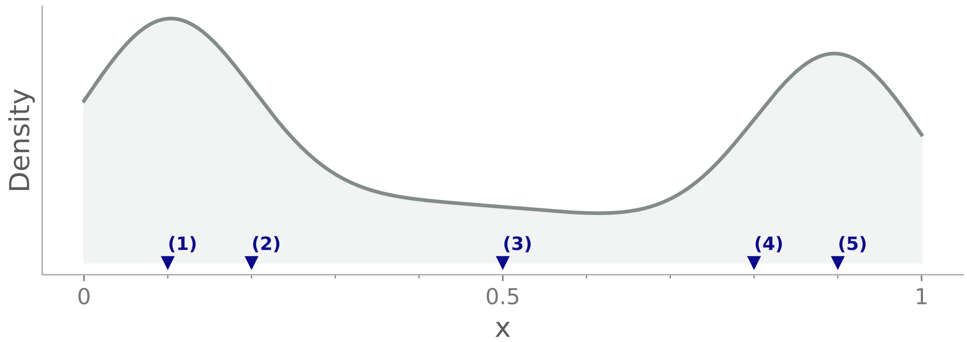
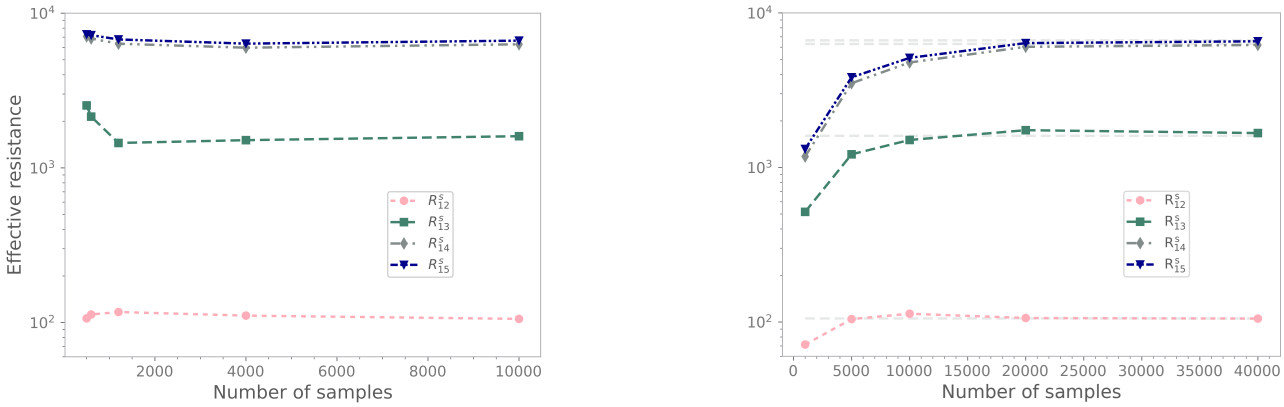


Figure 6: Non-uniform data distribution $\mu([0,1])$. The y-axis shows the density of the data distribution $\mu([0,1])$. The x-axis shows the support. The points we consider are marked by the blue triangles labeled $(1), \ldots, (5)$.

We note that in order to construct the α-cover, we use the cover-tree algorithm [20, 21].

In the experiment, the region-based ER R^s_{1j} between point 1 and points $j \in [2,3,4,5]$ are calculated for both the α-cover graph with p_i scaling (A) and the density graph (B). Figure 7 shows that the region-based ER on the graph (A) converges to the same limit as the region-based ER on graph (B) when increasing the number of samples used to estimate the local probabilities p_i.

This is what we wanted to show because it means that region-based ER can be calculated in two ways, either using a graph constructed on an α-cover with appropriate scaling or by using a graph constructed directly on samples from $\mu([0,1])$. The benefit of constructing the graph on the α-cover is that the graph size will be independent of the number of samples. At the same time, accuracy can still be increased by improving the estimates of the local probabilities p_i. This has the desirable property that it satisfies the condition of a streaming algorithm; See Section 6 for more details.



(a) Graph built on samples from density

(b) Graph built on α-cover with region-wise scaling

Figure 7: Demonstration of region-based ER on α-cover. We see that the ER in (b) converges to the asymptotics of the ER in (a) (weak gray lines). (This is a good thing). The dotted lines corresponds to region-based ER R^s_{1j} between point 1 and points $j \in [2,3,4,5]$ respectively. The pink line corresponds to R^s_{12}, the green line to R^s_{13}, the grey line to R^s_{14} and the blue line to R^s_{15}. The y-axis shows the region-based ER, while the x-axis is different for each sub-figure. (a) x-axis shows the number of samples used to construct the graph. (b) x-axis shows the number of samples used to estimate the local probabilities p_i for scaling of the α-cover graph.

Furthermore, using an α-cover graph with region-wise scaling, a smaller graph can be used to calculate the region-based ER, provided sufficient samples are used to estimate the local probabilities p_i. Since estimating these local probabilities is cheap, time is saved. Table 1 illustrates this; the table has to be seen in relation to the convergence results in Figure 7.

Table 1: Example of the time saved by calculating ER on an α-cover graph (graph A) instead of a graph constructed directly on the samples from the distribution (graph B). The first column shows the graphs used to calculate the ER; These are Graphs (A) and (B), described earlier. The second column is the number of samples used (The number in parenthesis is the size of the α-cover). The last column is the time to calculate the region-based ER using the Schur complement on the two graph types (The time in parenthesis is the time to estimate the local probabilities p_i for the α-cover).

Graph type	Number of samples	Time (s)
Graph (A)	21122 (1122)	0.04s + (0.07s)
Graph (B)	4000	4.6s

Remark 7.3. *An additional note is that also for this experiment, using the region-based ER as a measure of distance gives a meaningful ordering, namely,* $R^s_{12} < R^s_{13} < R^s_{14} < R^s_{15}$. *Moreover, as more samples are used to estimate the local probabilities, the distance between the samples increases. Due to the shape of the density, it makes sense that the distance between points should be larger when the density is taken into account. Especially points separated by the region of low density, which is also what we observe.*

Acknowledgments

AC was partially funded by NSF DMS 1819222 and 2012266, and a gift from Intel research. YF is funded by the NIH grant NINDS (PHS) U19NS107466 Reverse Engineering the Brain Stem Circuits that Govern Exploratory Behavior. RB is funded by NSF under CNS 1804829. AO is funded by Simula Research Laboratory.

References

[1] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[2] Mark Herbster and Massimiliano Pontil. Prediction on a graph with a perceptron. *Advances in neural information processing systems*, 19, 2006.

[3] Teng Zhang and Changjiang Bu. Detecting community structure in complex networks via resistance distance. *Physica A: Statistical Mechanics and its Applications*, 526:120782, 2019.

[4] Jihun Ham, Daniel D Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, page 47, 2004.

[5] Stefan Forcey and Drew Scalzo. Phylogenetic networks as circuits with resistance distance. *Frontiers in Genetics*, 11:1177, 2020.

[6] Sotharith Tauch, William Liu, and Russel Pears. Measuring cascade effects in interdependent networks by using effective graph resistance. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 683–688, 2015.

[7] Yakup Koç, Martijn Warnier, Piet Van Mieghem, Robert E. Kooij, and Frances M.T. Brazier. The impact of the topology on cascading failures in a power grid model. *Physica A: Statistical Mechanics and its Applications*, 402:169–179, 2014.

[8] Xiangrong Wang, Yakup Koç, Robert E. Kooij, and Piet Van Mieghem. A network approach for power grid robustness against cascading failures. In *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 208–214, 2015.

[9] Guido Cavraro and Vassilis Kekatos. Graph algorithms for topology identification using power grid probing. *IEEE Control Systems Letters*, 2(4):689–694, 2018.

[10] László Lovász. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2(1-46):4, 1993.

[11] Stephen P Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Mixing times for random walks on geometric random graphs. In *ALENEX/ANALCO*, pages 240–249, 2005.

[12] Chen Avin and Gunes Ercal. On the cover time and mixing time of random geometric graphs. *Theoretical Computer Science*, 380(1-2):2–22, 2007.

[13] Ulrike Luxburg, Agnes Radl, and Matthias Hein. Getting lost in space: Large sample analysis of the resistance distance. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[14] Ulrike Von Luxburg, Agnes Radl, and Matthias Hein. Hitting and commute times in large random neighborhood graphs. *The Journal of Machine Learning Research*, 15(1):1751–1798, 2014.

[15] Yue Song, David J Hill, and Tao Liu. On extension of effective resistance with application to graph laplacian definiteness and power network stability. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(11):4415–4428, 2019.

[16] Douglas J Klein and Milan Randić. Resistance distance. *Journal of mathematical chemistry*, 12(1):81–95, 1993.

[17] Palle ET Jorgensen and PJ Pearse Erin. Operator theory and analysis of infinite networks. *arXiv preprint arXiv:0806.3881*, 3, 2008.

[18] Arpita Ghosh, Stephen Boyd, and Amin Saberi. Minimizing effective resistance of a graph. *SIAM Review*, 50(1):37–66, 2008.

[19] Iqra Altaf Gillani and Amitabha Bagchi. A queueing network-based distributed laplacian solver for directed graphs. *Information Processing Letters*, 166:106040, 2021.

[20] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd international conference on Machine learning*, pages 97–104, 2006.

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

[22] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends® in Theoretical Computer Science*, 1(2):117–236, 2005.

[23] Ittai Abraham, Yair Bartal, and Ofer Neimany. Advances in metric embedding theory. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 271–286, 2006.

A Definitions

Definition A.1 (The doubling dimension). *(Adapted from [23])*
*Let (M, d) be a metric space. The **doubling constant** of M is the minimal κ required to cover a ball $B_r(x)$ by κ balls of radius $r/2$, for all $x \in M$ and for all $r > 0$. The **doubling dimension** of X is defined as $\mathsf{ddim}(M) = \log_2(\kappa)$.*

A.1 Schur complement formulation of effective resistance between sets

[15] offered an explicit expression for the effective resistance between sets X_a, X_b in terms of the Schur complement. We restate this expression here, as we will use it when calculating the effective resistance for our experiments. Let L/L_{cc} be the Schur complement, Definition A.2, of the Laplacian L with respect to the block L_{cc}, where L_{cc} is the block corresponding to nodes in the set $X_c = X \backslash (X_a \cup X_b)$. The effective resistance between the sets X_a, X_b can then be defined as

$$R^s(X_a, X_b) = (e_{X_a}^\top (L/L_{cc}) e_{X_a})^{-1} \tag{A.1}$$

where e_{X_a} is the vector with all ones for $i \in X_a$ and zero otherwise.

Definition A.2 (Schur complement). *Consider the block partition of the matrix $M = \begin{bmatrix} A & B \\ C, & D \end{bmatrix}$, where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{m \times m}$. Provided D is non-singular, we define the Schur complement of M as $M/D = A - BD^{-1}C$.*

B Proof of Proposition 3.6

B.1 Properties of Contractions

In this section, we develop several tools that we will use throughout our entire proofs section. Let Z denote a space. We will be interested in functions, $v : Z \to \mathbb{R}$, as well as operators $A : v \mapsto (Av : Z \to \mathbb{R})$ that take functions to functions. For our purposes, Z will often be M, the underlying metric space, but will sometimes also be a finite sample from M.

We begin by defining the ℓ_∞-norm, which will play a key role in our analysis.

Definition B.1. *Let $v : Z \to \mathbb{R}$ be a map. Then $||v||_\infty$ denotes the ℓ_∞ norm of v, and is defined as $||v||_\infty = \sup_{z \in Z} |v(z)|$.*

We can also define the ℓ_∞-norm of an operator.

Definition B.2. *Let A be an operator, meaning it maps functions $(Z \to \mathbb{R})$ to other functions. Then*

$$||A||_\infty = \sup_{||v||_\infty > 0, v: Z \to \mathbb{R}} \frac{||Av||_\infty}{||v||_\infty}.$$

We will be especially interested in *contractions*, which are operators with infinity norm strictly less than 1. We are also interested in contractions combined with translations. We call such operators nice.

Definition B.3. *An operator, T, is **nice**, if there exists a contraction A and a map b such that $Tv = Av + b$.*

We are now prepared for our first useful result:

Lemma B.4. *Let T be a **nice** operator. Then there is a unique map $v : Z \to \mathbb{R}$ such that $Tv(z) = v(z)$ for all $z \in Z$.*

Proof. Let $Tv = Av + b$. Define $u_0 = b$ and $u_i = Tu_{i-1}$ for $i \geq 1$. Let $||A||_\infty = \rho$ where $0 \leq \rho < 1$ because T is nice. Observe that for $i \geq 1$,

$$\begin{aligned}||u_{i+1} - u_i||_\infty &= ||(Au_i(x) + b) - (Au_{i-1} + b)||_\infty \\ &= \sup_{x \in M} ||A(u_i - u_{i-1})||_\infty \\ &\leq \rho ||u_i - u_{i-1}||_\infty .\end{aligned}$$

It follows that $u_0, u_1, \ldots$ is a Cauchy sequence under the ℓ_∞ metric. Since the set of all functions on the reals is closed, it follows that $u_0, u_1, \ldots,$ converges to some u, which must satisfy $Tu = u$.

To show uniqueness, we can simply bound the infinity distance between any two fixed points to see that this distance is at most ρ times itself. Since $\rho < 1$, it follows that the two fixed points must be the same function u. □

We prove one addition lemma about nice operators.

Lemma B.5. *Let T be a nice operator with $Tv = Av + b$ where $||A||_\infty = \rho$. Suppose that function v satisfies $||v - Tv||_\infty < \epsilon$. If u denotes the unique fixed point of T, then*

$$||u - v||_\infty < \frac{\epsilon}{1 - \rho}.$$

Proof. By using the same argument as the previous lemma, we see that the sequence $v, Tv, T^2v, \ldots$ must converge to u. Since $||v - Tv||_\infty < \epsilon$, it follows that $||T^i v - T^{i+1}|| < \rho^i \epsilon$. Summing the infinite geometric sequence gives us the desired result. □

B.2 Proving the existence of the energy-minimizing voltage (the limit object): Proposition 3.6

We begin by defining an operator that characterizes our desired limit object, v^*, (defined in Definition 3.1). In Lemma B.7 - B.9, we prove the existence and uniqueness of v^*. In Lemma B.10, we then prove that v^* is the energy-minimizing voltage from Definition 3.1.

Definition B.6. *Let k be a kernel and $\hat{k}$ be the normalized version (Definition 3.2). Then A_* is the operator defined as follows. If $v : M \to \mathbb{R}$ is a measurable function, then $A_* v$ is the function $M \to \mathbb{R}$ defined by*

$$(A_* v)(x) = \int_M v^*(y) \hat{k}(x, y) \mathbb{1}(y \in M \setminus (M_g \cup M_s)) d\mu(y).$$

We also let $b_* : M \to \mathbb{R}$ denote the fixed function defined as

$$b_*(x) = \int_M \hat{k}(x, y) \mathbb{1}(y \in M_s) d\mu(y).$$

Together, we let T_* be the operator

$$T_* v = A_* v + b_*.$$

Our main idea will be to show the following two statements holds:

1. There exists $m > 0$ such that T_*^m is contractive (Definition B.3).

2. v^* is a fixed point of T_*.

Together with Lemma B.4, this will prove the existence and uniqueness of v^*. We start by proving the first claim, which relies on the following technical Lemma that characterizes iterative powers of A_*.

Lemma B.7. *Let $v : M \to \mathbb{R}$ be a measurable map and $x \in M$ be a point. Then for all $i \geq 1$,*

$$|(A_*^i v)(x)| \leq \int_M |v(y)| \hat{k}^{(i)}(x, y) \mathbb{1}\left(y \in M \setminus (M_g \cup M_s)\right).$$

Proof. We proceed by induction on i. In the base case, $i = 1$, and we have

$$\begin{aligned}|(A_* v)(x)| &= \left|\int_M v(y)\hat{k}(x,y)\mathbb{1}\left(y \in M \setminus (M_g \cup M_s)\right) d\mu(y)\right| \\ &\leq \int_M |v(y)|\hat{k}(x,y)\mathbb{1}\left(y \in M \setminus (M_g \cup M_s)\right) d\mu(y) \\ &\leq \int_M |v(y)|\hat{k}(x,y) d\mu(y).\end{aligned}$$

For the inductive step, let $i > 1$ and assume that the claim holds for $i - 1$. Then

$$\begin{aligned}|(A_*^i v)(x)| &= |(A_*(A_*^{(i-1)}v))(x)| \\ &= \left|\int_M (A_*^{(i-1)}v)(y)\hat{k}(x,y)\mathbb{1}\left(y \in M \setminus (M_g \cup M_s)\right) d\mu(y)\right| \\ &\leq \int_M \left|(A_*^{(i-1)}v)(y)\right| \hat{k}(x,y)\mathbb{1}\left(y \in M \setminus (M_g \cup M_s)\right) d\mu(y) \\ &\leq \int_M \left|(A_*^{(i-1)}v)(y)\right| \hat{k}(x,y) d\mu(y) \\ &\leq \int_M \left(\int_M |v(z)|\hat{k}^{(i-1)}(y,z)\mathbb{1}\left(z \in M \setminus (M_g \cup M_s)\right) d\mu(z)\right) \hat{k}(x,y) d\mu(y) \\ &= \int_M \left(\int_M \hat{k}(x,y)\hat{k}^{(i-1)}(y,z) d\mu(y)\right) |v(z)|\mathbb{1}\left(z \in M \setminus (M_g \cup M_s)\right) d\mu(z) \\ &= \int_M |v(z)|\hat{k}^{(i)}(x,z)\mathbb{1}\left(z \in M \setminus (M_g \cup M_s)\right) d\mu(z),\end{aligned}$$

as desired. Here the inequalities hold by

1. Moving the absolute value into the expectation.
2. bounding the indicator function by 1.
3. Applying the inductive hypothesis.
4. Applying Fubini's theorem to switch the order.
5. Applying convolution (Definition 3.4).

□

We now show that there exists a power of A_* that is a contraction.

Lemma B.8. *Let α be as in the statement of Proposition 3.6. Then for all maps $v : M \to \mathbb{R}$,*

$$\sup_{x \in M} |(A_*^m v)(x)| \leq (1 - \alpha) \sup_{x \in M} |v(x)|.$$

Proof. Fix $x \in M$. Then by applying Lemma B.7,

$$\begin{aligned}|(A_*^m v)(x)| &\leq \int_M |v(y)|\hat{k}^{(m)}(x,y)\mathbb{1}\left(y \in M \setminus (M_g \cup M_s)\right) d\mu(y) \\ &\leq \sup_{x \in M} |v(x)| \int_M \hat{k}^{(m)}(x,y)\mathbb{1}\left(y \in M \setminus (M_g \cup M_s)\right) d\mu(y).\end{aligned}$$

Because $\hat{k}$ is normalized, $\hat{k}^{(m)}$ is as well, meaning that $\int_M \hat{k}^{(m)}(x,y)d\mu(y) = 1$. However, we also have by assumption (Proposition 3.6) that $M = C_\alpha^m(M_g \cup M_s)$, which implies that

$$\int_M \hat{k}^{(m)}(x,y)\mathbb{1}(y \in M_g \cup M_s)d\mu(y) \geq \alpha.$$

Substituting this, it follows that

$$
\begin{aligned}
|(A_*^m v)(x)| &\leq \sup_{x\in M} |v(x)| \int_M \hat{k}^{(m)}(x,y)\mathbb{1}\left(y \in M \setminus (M_g \cup M_s)\right) d\mu(y). \\
&\leq (1-\alpha) \sup_{x\in M} |v(x)|
\end{aligned}
$$

□

We are now prepared to show that there exists a unique fixed point of T_*.

Lemma B.9. *There exists a unique function $v^* : M \to [0,1]$ such that $T_* v^* = v^*$.*

Proof. Let m be as in Lemma B.8. Observe that the operator $T_*^m v$ can be written as

$$T_*^m : v \mapsto A_*^m v + b_*(x) + (A_* b_*)(x) + \cdots + (A_*^{m-1} b_*)(x).$$

By Lemma B.8, A_*^m is a contraction and therefore T_*^m is a nice (Definition B.3) operator. It follows by Lemma B.4 that T_*^m has a unique fixed point which we denote as v^*.

Any fixed point of T_* is a fixed point of T_*^m, and therefore it suffices to show that v^* is a fixed point of T_* – uniqueness will follow from the unqiueness of v^* w.r.t. T_*^m. To do so, observe that

$$T_*^m(T_* v^*) = T_*^m(T_* v^*) = T_*(T_*^m v^*) = T_*(v^*).$$

Thus $T_* v^*$ is a fixed point of T_*^m, meaning that is must equal v^* by the uniqueness of v^*. Thus $T_* v^* = v^*$, as desired. □

We now show that v^* is the energy-minimizing voltage of the energy defined in Definition 3.1.

Lemma B.10. *Let $M_s, M_g \subset M$ be measurable disjoint subsets. Let V_{M_s,M_g} be the set of all measurable functions $v : M \to [0,1]$ with $v(x) = 1$ for all $x \in M_s$ and $v(x) = 0$ for all $x \in M_g$. The minimizer of the energy $E(v)$ defined in Definition 3.1, is the function $v \in V_{M_s,M_g}$ that satisfies $v^* = T_* v^*$.*

Proof. To show that v^* is a minimizer of $E(v)$ from Definition 3.1, it is sufficient to show that the following two statements hold.

1. If $v^* = T_* v^*$ then $E(v^*) < E(u)$ for all $u \in V_{M_s,M_g}$, $u \neq v^*$
2. If $E(v^* + u) \geq E(v^*)$ for all $u \in V_{M_s,M_g}$ then $v^* = T_* v^*$

Let $\varepsilon > 0$ and $v \in V_{M_s,M_g}$. Consider $u = v^* + \varepsilon v \in V_{M_s,M_g}$ and the energy from Definition 3.1 evalauted at u namely

$$
\begin{aligned}
E(u) = E(v^* + \varepsilon v) &= \int_M \int_M \hat{k}(x,y)[(v^*(x) + \varepsilon v(x)) - (v^*(y) + \varepsilon v(y))]^2 d\mu(x) d\mu(y) \\
&= \int_M \int_M \hat{k}(x,y)(v^*(x) - v^*(y))^2 d\mu(x) d\mu(y) \\
&- 2\varepsilon \int_M \int_M \hat{k}(x,y)(v^*(x) - v^*(y))(v(x) - v(y)) d\mu(x) d\mu(y) \\
&+ \varepsilon^2 \int_M \int_M \hat{k}(x,y)(v(x) - v(y))^2 d\mu(x) d\mu(y) \\
&= E(v^*) + \varepsilon^2 E(v) - 4\varepsilon E(v^*, v).
\end{aligned}
$$

Here $E(v^*, v) \coloneqq \frac{1}{2} \int_M \int_M \hat{k}(x,y)(v^*(x) - v^*(y))(v(x) - v(y)) d\mu(x) d\mu(y)$. We then have

$$E(v^*) = E(v^* + \varepsilon v) - \varepsilon^2 E(v) + 4\varepsilon E(v^*, v) \tag{B.1}$$

For the first statement, if $v = T_* v^*$ it follows from Lemma B.11 that $E(v^*, v) = 0$. Consequently, with $\varepsilon = 1$,

$$E(v^*) = E(v^* + \varepsilon v) - E(v).$$

Since $E(v^* + \varepsilon v) > 0$ and $E(v) > 0$ it follows that $E(v^*) < E(v^* + \varepsilon v)$. Because u was arbitrary, the first statement holds.

For the second statement, if $E(v^*+\varepsilon v) \geq E(v^*)$ then from Eq. (B.1) we require $4\varepsilon E(v^*, v) - \varepsilon^2 E(v) \leq 0$. In other words, $\varepsilon E(v) \geq 4E(v^*, v)$ for any $\varepsilon > 0$. Since ε can be arbitrarily small, this means we need $E(v^*, v) = 0$. From Lemma B.11 it follows that if $E(v^*, v) = 0$ for all $v \in V_{M_s,M_g}$ then $v^* = Tv^*$. Consequently, the second statement holds. □

Lemma B.11. *Consider the bilinear energy* $E(v^*, v) = \frac{1}{2}\int_M \int_M \hat{k}(x,y)(v^*(x)-v^*(y))(v(x)-v(y))d\mu(x)d\mu(y)$ *defined in lemma B.10.*

1. *If* $v^* = T_* v^*$ *then* $E(v^*, v) = 0$ *for all* $v \in V_{M_s,M_g}$
2. *If* $E(v^*, v) = 0$ *for all* $v \in V_{M_s,M_g}$ *then* $v^* = T_* v^*$

Proof. From Lemma B.12 it follows that

$$E(v^*, v) = \int_M v(x)[v^*(x) - (Tv^*)(x)]d\mu(x).$$

Consequently, if $v^* = T_* v^*$, then $E(v^*, v) = \int_M v(x)[v^*(x) - v^*(x)]d\mu(x) = 0$ for all $v \in V_{M_s,M_g}$. Now, consider the second statement. If $E(v^*, v) = 0$ for all $v \in V_{M_s,M_g}$ then we require $v^* - Tv^* = 0$. Consequently, $v^* = Tv^*$. □

Lemma B.12. *The bilinear energy form*

$$E(v^*, v) = \frac{1}{2}\int_M \int_M \hat{k}(x,y)(v^*(x) - v^*(y))(v(x) - v(y))d\mu(x)d\mu(y)$$

can be written as

$$E(v^*, v) = \int_M v(x)[v^*(x) - (Tv^*)(x)]d\mu(x).$$

Proof. We have

$$\begin{aligned} E(v^*, v) &= \frac{1}{2}\int_M \int_M \hat{k}(x,y)v^*(x)v(x)d\mu(x)d\mu(y) + \frac{1}{2}\int_M \int_M \hat{k}(x,y)v^*(y)v(y)d\mu(x)d\mu(y) \\ &- \frac{1}{2}\int_M \int_M \hat{k}(x,y)v^*(x)v(y)d\mu(x)d\mu(y) - \frac{1}{2}\int_M \int_M \hat{k}(x,y)v^*(y)v(x)d\mu(x)d\mu(y) \\ &= \underbrace{\int_M \int_M \hat{k}(x,y)v^*(x)v(x)d\mu(x)d\mu(y)}_{I_1} - \underbrace{\int_M \int_M \hat{k}(x,y)v^*(y)v(x)d\mu(x)d\mu(y)}_{I_2} \end{aligned}$$

For I_1 we have

$$\begin{aligned} \int_M \int_M \hat{k}(x,y)v^*(x)v(x)d\mu(x)d\mu(y) &= \int_M \left(\int_M \hat{k}(x,y)d\mu(y)\right)v^*(x)v(x)d\mu(x) \\ &= \int_M v^*(x)v(x)d\mu(x). \end{aligned}$$

In the last step, we used that $\int_M \hat{k}(x,y)d\mu(y) = 1$, since from Definition 3.2 we have $\hat{k}(x,y) = k(x,y)/\int_M k(x,z)d\mu(z)$. For I_2 we have

$$\begin{aligned} \int_M \int_M \hat{k}(x,y)v^*(y)v(x)d\mu(x)d\mu(y) &= \int_M v(x)\left(\int_M \hat{k}(x,y)v^*(y)d\mu(y)\right)d\mu(x) \\ &= \int_M v(x)(T_* v^*)(x)d\mu(x) \end{aligned}$$

It follows that

$$E(v^*, v) = \int_M [v^*(x)v(x) - v(x)(Tv^*)(x)]d\mu(x) = \int_M v(x)[v^*(x) - (Tv^*)(x)]d\mu(x)$$

□

C Proofs from Analysis Section

In this section, our goal will be to show that effective resistances computed over metric graphs that are sampled from M will converge towards the desired limit object. We begin with some technical results for approximating integrals with sums over finite samples from μ.

C.1 Bounding integrals of functions

We begin by stating Hoeffding's bound for a map $f : M \to [0,1]$, $\int_M f(y)d\mu(y)$ can be approximated with the sum $\frac{1}{n}\sum_{i=1}^n f(x_i)$.

Lemma C.1 (Hoeffding)**.** *Let $S_n \sim \mu^n$ be n i.i.d draws with $S_n = \{x_1, \ldots, x_n\}$, and let $f : M \to [0,1]$ be a map. Then,*

$$\Pr\left[\left|\int_M f(y)d\mu(y) - \frac{1}{n}\sum_{i=1}^n f(x_i)\right| > \epsilon\right] \leq 2\exp\left(-2n\epsilon^2\right).$$

Next, we bound the quotient of two integrals.

Lemma C.2. *Let $S_n \sim \mu^n$ be n i.i.d draws with $S_n = \{x_1, \ldots, x_n\}$, and let $f : M \to [0,1]$ and $g : M \to [0,1]$ be two maps. Suppose that $\int_M g(y)d\mu(y) \geq G$. Then*

$$\Pr\left[\left|\frac{\int_M f(y)d\mu(y)}{\int_M g(y)d\mu(y)} - \frac{\frac{1}{n}\sum_{i=1}^n f(x_i)}{\frac{1}{n}\sum_{i=1}^n g(x_i)}\right| > \epsilon\right] \leq 4\exp\left(-8nG^2\epsilon^2\right).$$

Proof. It suffices to bound the probability that

$$\left|\int_M f(y)d\mu(y) - \frac{1}{n}\sum_{i=1}^n f(x_i)\right| < \frac{\epsilon}{4}$$

and

$$\left|\int_M g(y)d\mu(y) - \frac{1}{n}\sum_{i=1}^n g(x_i)\right| < \frac{G\epsilon}{4}$$

as these two equations imply the desired bound by straightforward algebra. The probability that both of these occur can be bounded by a union bound after applying Hoeffding's inequality twice. □

C.2 Showing that A_n^m is a contraction

We define the *finite* analogs of A_* and b_* (Definition B.6).

Definition C.3. *Let $S_n \sim \mu^n$ be a sample of n points, $x_1, \ldots, x_n$. Let $v : S_n \to \mathbb{R}$ be a map. Then the operators A_n and the map b_n are defined as*

$$(A_n v)(x) = \frac{\sum_{i=1}^n k(x, x_i)v(x_i)\mathbb{1}\left(x_i \in M \setminus (M_0 \cup M_1)\right)}{\sum_{i=1}^n k(x, x_i)},$$

and

$$b_n(x) = \frac{\sum_{i=1}^n k(x, x_i)\mathbb{1}(x_i \in M_1)}{\sum_{i=1}^n k(x, x_i)}.$$

Our goal in this section is to show that A_n^m is also a contraction, where m is as defined in the statement of Theorem 4.4.

We first define the parameter K as follows.

Definition C.4. *Let K be the inverse of the minimal degree in M. That is, let*

$$K = \max_{x \in M} \frac{1}{\int_M k(x,y) d\mu(y)}.$$

This is well defined because M is compact and because $\int_M k(x,y)d\mu(y)$ is both continuous and larger than 0 by assumption. Because k has range in $[0,1]$, it follows that

$$\max_{x \in M} \hat{k}(x) \leq K \tag{C.1}$$

where $\hat{k}$ is the normalized kernel defined in Definition 3.2.

We now prove a technical lemma that will assist us in understanding the structure of G_n, the metric graph built over a finite sample $S_n \sim \mu^n$.

Lemma C.5. *Let $P \subseteq M$ be a measurable set, and let $i > 1$. Let $p > 0$ and suppose that for all $x \in M$,*

$$\int_M \hat{k}^{(i)}(x,y) \mathbb{1}(y \in P) d\mu(y) \geq p.$$

Then there exists a measurable subset $Q \subseteq M$ such that the following two properties hold:

- *For all $x \in M$, $\int_M \hat{k}^{(i-1)}(x,y)\mathbb{1}(y \in Q)d\mu(y) \geq \frac{p}{2K-p}$, with K as defined above.*
- *For all $x \in Q$, $\int \hat{k}(x,y)\mathbb{1}(y \in P)d\mu(y) \geq \frac{p}{2}$.*

Proof. Observe that by the definition of convolution and Fubini's theorem,

$$\begin{aligned}
\int_M \hat{k}^{(i)}(x,y)\mathbb{1}(y \in P)d\mu(y) &= \int_M (\hat{k}^{(i-1)} \circ \hat{k})(x,y)\mathbb{1}(y \in P)d\mu(y) \\
&= \int_M \left(\int_M \hat{k}^{(i-1)}(x,z)\hat{k}(z,y)d\mu(z) \right) \mathbb{1}(y \in P)d\mu(y) \\
&= \int_M \hat{k}^{(i-1)}(x,z) \left(\int_M \hat{k}(z,y)\mathbb{1}(y \in P)d\mu(y) \right) d\mu(z).
\end{aligned}$$

Define $f(z) = \int_M \hat{k}(z,y)\mathbb{1}(y \in P)d\mu(y)$. Since $\hat{k} \leq K$, it follows that f has range $[0,K]$. We now define $Q = \{z : f(z) \geq \frac{p}{2}\}$, and claim that this suffices. By definition, the second property trivially holds. To see that the first property holds, let $q = \int_M \hat{k}^{(i-1)}(x,y)\mathbb{1}(y \in Q)d\mu(y)$. Then,

$$\begin{aligned}
p &\leq \int_M \hat{k}^{(i)}(x,y)\mathbb{1}(y \in P)d\mu(y) \\
&= \int_M \hat{k}^{(i-1)}(x,z) \left(\int_M \hat{k}(z,y)\mathbb{1}(y \in P)d\mu(y) \right) d\mu(z) \\
&= \int_M \hat{k}^{(i-1)}(x,z)f(z)d\mu(z) \\
&= \int_Q \hat{k}^{(i-1)}(x,z)f(z)d\mu(z) + \int_{M \setminus Q} \hat{k}^{(i-1)}(x,z)f(z)d\mu(z) \\
&\leq Kq + \left(\frac{p}{2}\right)(1-q).
\end{aligned}$$

Rearranging this gives the desired inequality. □

Recall that m and α are defined in the statement of Theorem 4.4 such that $M = C_\alpha^m(M_1 \cup M_0)$. We have the following lemma which gives additional structure to M.

Lemma C.6. *There exists sets $P_0, P_1, P_2, \dots, P_m \subseteq M$ with $P_0 = M$ and $P_m = M_0 \cup M_1$ such that the following holds. There exists positive reals $p_1, p_2, \dots p_m$ such that for all $1 \leq i \leq m$, for all $x \in P_{i-1}$,*

$$\int_M \hat{k}(x,y)\mathbb{1}(y \in P_i)d\mu(y) \geq p_i.$$

Proof. The idea is simple: we apply Lemma C.5 m times starting with $P_m = M_0 \cup M_1$. To establish the base case, recall that $M = C_\alpha^m(M_1 \cup M_0)$, which implies

$$\int_M \hat{k}^{(m)}(x,y)\mathbb{1}(y \in P_m)d\mu(y) \geq \alpha$$

for $\alpha > 0$.

Define $q_m = \alpha$. We will now show how to recursively construct P_i, q_{i-1} and p_i from q_i. Suppose that for all $x \in M$,

$$\int_M \hat{k}^{(i)}(x,y)\mathbb{1}(y \in P_i)d\mu(y) \geq q_i. \tag{C.2}$$

Then applying Lemma C.5, we let $P_{i-1} = Q$, $p_i = \frac{q_i}{2}$, and $q_{i-1} = \frac{q_i}{2k - q_i}$. It is easy to see by Lemma C.5 that doing so preserves Equation C.2 and that p_i satisfies the desired equation.

Finally, in the case that $i = 1$, we can simply let $p_1 = q_1$, as we no longer require Lemma C.5 since $\hat{k}^{(1)} = \hat{k}$. This completes the proof. □

We are now prepared to show that A_n^m is indeed a contraction over S_n.

Lemma C.7. *There exists an absolute constant $p > 0$ independent of n and S_n such that the following holds.*

$$\lim_{n \to \infty} \Pr_{S_n \sim \mu^n}[||A_n^m||_\infty \leq 1 - p] = 1.$$

Proof. Fix $P_0, \dots P_m$ and $p_0, p_1, \dots, p_m$ as in Lemma C.6. Let E denote the event that for all $1 \leq i \leq n$ and $1 \leq j \leq m$ such that $x_i \in P_{j-1}$,

$$\frac{\frac{1}{n}\sum_{t=1}^n k(x_i, x_t)\mathbb{1}(x_t \in P_j)}{\frac{1}{n}\sum_{t=1}^n k(x_i, x_t)} \geq \frac{p_i}{2}.$$

The key observation is that by applying Lemma C.2 to all mn pairs of points along with a union bound, there exists an absolute constant C for which $\Pr[E] \geq 1 - nm\exp(-nC)$. Thus for n sufficiently large, the probability of E converges to 1. This is a direct result of the fact that by Lemma C.6,

$$\int_M \hat{k}(x_i, y)\mathbb{1}(y \in P_j)d\mu(y) \geq p_j,$$

whenever $x_i \in P_{j-1}$. Note that although there is a technical independence issue as the function $x_t \to k(x_i, x_t)$ has a dependence on x_i which is in S_n, this can be resolved by observing that for any function f, for n sufficiently large, $\left|\frac{1}{n}\sum f(x_j) - \frac{1}{n-1}\sum_{j \neq i} f(x_j)\right|$ is small.

Finally, given that the event E occurs, we can bound $||A_n^m||_\infty$ as follows. For any $v : S_n \to \mathbb{R}$, we have from Definition C.3,

$$\begin{aligned}
|A_n^v(x)| &\leq \sum_{t_1=1}^n \frac{k(x, x_{t_1})}{\sum_{i=1}^n k(x, x_i)} \sum_{t_2=1}^n \frac{k(x_{t_1}, x_{t_2})}{\sum_{i=1}^n k(x_{t_2}, x_i)} \cdots \sum_{t_m=1}^n \frac{k(x_{t_{m-1}}, x_{t_m})}{\sum_{i=1}^n k(x_{t_m}, x_i)} \mathbb{1}(x_{t_m} \in M \setminus (M_1 \cup M_0) \\
&= 1 - \sum_{t_1=1}^n \frac{k(x, x_{t_1})}{\sum_{i=1}^n k(x, x_i)} \sum_{t_2=1}^n \frac{k(x_{t_1}, x_{t_2})}{\sum_{i=1}^n k(x_{t_2}, x_i)} \cdots \sum_{t_m=1}^n \frac{k(x_{t_{m-1}}, x_{t_m})}{\sum_{i=1}^n k(x_{t_m}, x_i)} \mathbb{1}(x_{t_m} \in M_1 \cup M_0) \\
&\leq 1 - \sum_{x_{t_1} \in P_1} \frac{k(x, x_{t_1})}{\sum_{i=1}^n k(x, x_i)} \sum_{x_{t_2} \in P_2} \frac{k(x_{t_1}, x_{t_2})}{\sum_{i=1}^n k(x_{t_2}, x_i)} \cdots \sum_{x_{t_m} \in P_m} \frac{k(x_{t_{m-1}}, x_{t_m})}{\sum_{i=1}^n k(x_{t_m}, x_i)} \mathbb{1}(x_{t_m} \in M_1 \cup M_0) \\
&\leq 1 - \frac{\prod_{i=1}^m p_i}{2^m}.
\end{aligned}$$

Thus selecting $p = \frac{\prod_{i=1}^{m} p_i}{2^m}$ suffices, as desired. □

As a quick remark, although the factor of contraction shown here is *extremely* close to 1, this analysis is just considering worst-case situations in which M has a very complicated structure. In most actual cases, the factor of contraction is far far smaller.

C.3 Finishing the proof

We are now prepared to compare v^* with v_n^*. They key idea is to restrict v^* to S_n and then compare it with $A_n^m v^*$. We begin with the following lemma.

Lemma C.8. *Let $u : M \to [0, 1]$ be a measurable function, $x \in M$ be a point, and $\epsilon > 0$ be a real number. Then there exists a constant C such that*

$$\Pr_{S \sim \mu^n} \left[\left| (A_* u + b_*)(x) - (A_n u + b_n)(x) \right| > \epsilon \right] < 4 \exp(-C n \epsilon^2).$$

Proof. We will examine $|A_* u - A_n u|$ and $|b_* - b_n|$ separately and then apply the triangle inequality. Let $M' = M \setminus (M_1 \cup M_0)$. Then,

$$\left| (A_* u)(x) - (A_n u)(x) \right| = \left| \frac{\int_M' k(x, y) u(y) d\mu(y)}{\int_M k(x, y) d\mu(y)} - \frac{\frac{1}{n} \sum_{x_i \in M'} k(x, x_i) u(x_i)}{\frac{1}{n} \sum_{x_i \in M'} k(x, x_i)} \right|.$$

However, by Lemma C.2, this quantity is at most ϵ with probability $2 \exp(-O(n \epsilon^2))$. We can apply a similar argument for $b_* - b_n$ which completes the proof. □

Lemma C.9. *Restrict v^* and b_* (definition B.6) to S_n. Then $||v^* - (A_n v^* + b_n)||_\infty \to 0$ and $||b_* - b_n||_\infty \to 0$ in probability. Here again, the infinity norms are taken over S_n, not M.*

Proof. Simply apply Lemma C.8 for all $x = x_i$ and $u = v^*$ and then apply a union bound. While there are technically independence issues, this is solved by observing that x_i is independent of all other elements of S_n, and the averages taken over S_n *ignoring* x_i are barely different from those including x_i. □

Lemma C.10. *Let $c_n = b_n + A_n b_n + A_n^2 b_n + \dots A_n^{m-1} b_n$. Then $||v^* - (A_n^m v^* + c_n)||_\infty \to 0$ in probability.*

Proof. The key idea is that A_n is at most an averaging operator, and consequently $||A_n|| \leq 1$. Intuitively, applying A_n to a map $v : S_n \to \mathbb{R}$ cannot increase $\max_{x \in S_n} |v(x)|$. Let $T_n v = A_n v + b_n$. This implies that for all u and v,

$$||u - v||_\infty \geq ||T_n u - T_n v||.$$

Applying this m times along with the triangle inequality, we see that

$$|v^* - T_n^m v^*| \leq (m - 1)|v^* - T_n v^*|.$$

However, since the latter goes to 0 in probability, it follows that the former does as well. All the remains to see is that $T_n^m v$ precisely equals $A_n^m v^* c_n$, as desired. □

We can finally prove Theorem 4.4.

Proof. By the previous Lemma, for any $\epsilon > 0$ and $\delta > 0$ there exists n such that with probability $1 - \delta$, $||v^* - (A_n^m v^* + c_n)||_\infty < \epsilon$. Since A_n^m is a contraction, it follows by Lemma B.5 that v^* has distance at most $\frac{\epsilon}{p}$ from v_n^*, the fixed point of $v \mapsto A_n^m v + c_n$. Since p is fixed, it follows that $|v_n^* - v_n|_\infty \to 0$. Finally, for an arbitrary point $x \in M$, simply applying Lemma C.2 one last time over the definition of $v_n^*(x)$ implies the desired result. □

D Proof of Convergence using α-covers

Here, our goal is to prove Theorem 6.8. To do so, we impose an additional technical restriction on our kernel k.

Definition D.1. *Let μ be a measure over metric space M, and let $k : M \times M \to [0,1]$ be a kernel. Let $V \subset M \times M$ denote the set of all (x, y) for which k is continuous at (x, y), and let $\mu \times \mu$ denote the product measure over $M \times M$. Then k is said to be μ-**continuous** if*

$$(\mu \times \mu)\left((M \times M) \setminus V\right) = 0.$$

It is easy to verify that for most common measures over $\mathbb{R}^d$ that the Gaussian and Radial kernels are μ-continuous.

We now assume that k is a μ-continuous kernel, and moreover that

Next, we define an analog to the quantities A_n and b_n with respect to our α-cover $\mathcal{C}$.

Definition D.2. *Let $S \sim \mu^n$ be an i.i.d sample of data, and let $\mathcal{C} = \{c_1, \ldots, c_m\}$ be an α-cover. Let $\gamma_{i,n}$ denote the fraction of points from S that lie in the cell corresponding to c_i. Let $v : \mathcal{C} \to \mathbb{R}$ be a map. Then the operators $A_{\mathcal{C},n}$ and the map $b_{\mathcal{C},n}$ are defined as*

$$(A_{\mathcal{C},n}v)(x) = \frac{\sum_{i=1}^m k(x, c_i)v(c_i)\gamma_{i,n}\mathbb{1}\left(c_i \in M \setminus (M_0 \cup M_1)\right)}{\sum_{i=1}^m \gamma_{i,n}k(x, c_i)},$$

and

$$b_{\mathcal{C},n}(x) = \frac{\sum_{i=1}^m \gamma_{i,n}k(x, c_i)\mathbb{1}(c_i \in M_1)}{\sum_{i=1}^m \gamma_{i,n}k(x, c_i)}.$$

The goal is to show that computing the voltage function v using α-covers converges towards the same limit object as computing v using a direct sample. To achieve this, the idea is to show that the operators, $A_{\mathcal{C},n}$ and A_n (along with the functions $b_{\mathcal{C},n}, b_n$) behave similarly.

Lemma D.3. *Let $\mathcal{C}$ be any α-cover. There exists a function Δ_1 such that the following two things hold. First, for any $v : M \to [0,1]$, and any $x \in M$,*

$$|(A_{\mathcal{C},n}v)(x) - (A_n v)(x)| < \Delta_1(\alpha), \quad |b_{\mathcal{C},n}(x) - b_n(x)| < \Delta_1(\alpha).$$

Second, $\lim_{\alpha \to 0} \Delta_1(\alpha) = 0$.

Proof. This directly follows from the fact that k is μ continuous. Since the diameter of each cell is at most α, and since the support of M is compact, we see that the maximum deviation made in a single entry in the corresponding matrices for $A_{\mathcal{C},n}$ and A_n is bounded by some continuous function of α. □

Lemma D.4. *There exists a function $\Delta : \mathbb{R}^+ \to \mathbb{R}^+$ such that the following hold. First, for any ϵ, δ and α-cover, $\mathcal{C}$, there exists N such that for all $n \geq N$, with probability at least $1 - \delta$ over $S \sim \mu^n$,*

$$|v_n(x) - v_n^{\mathcal{C}}(x)| \leq \Delta(\alpha) + \epsilon.$$

Second, $\lim_{\alpha \to 0} \Delta(\alpha) = 0$.

Proof. Let m, p, N be such that for all $n \geq N$, $||A_n^m||_\infty \leq 1 - p$ with probability at least $1 - \delta$ over $S \sim \mu^n$. Such N must exist by Lemma C.7. It follows that with probability at least $1 - \delta$, T_n^m is a *nice* operator (Definition B.3).

By the definitions of v_n and $v_n^{\mathcal{C}}$, we have that $T_n^m v_n = v_n$ and $T_{n,\mathcal{C}}^m v_n^{\mathcal{C}} = v_n^{\mathcal{C}}$, where $T_{n,\mathcal{C}}(v) = A_{n,\mathcal{C}}(v) + b_{n,\mathcal{C}}$. It follows by applying the previous lemma m times that

$$|T_n^m(v_n^{\mathcal{C}}) - v_n^{\mathcal{C}}| = |T_n^m(v_n^{\mathcal{C}}) - T_{n,\mathcal{C}}^m(v_n^{\mathcal{C}})| \leq O(m\Delta_1(\alpha)).$$

Since T_n^m is nice with norm at most $1-p$, it follows by Lemma B.5 that

$$|v_n^{\mathcal{C}} - v_n| \leq O\left(m\frac{\Delta_1(\alpha)}{p}\right).$$

Since m and p can be chosen to be fixed and since $\lim_{\alpha\to\infty} \Delta_1(\alpha) \to 0$ from Lemma D.3, it follows that $\Delta(\alpha) \mapsto m\frac{\Delta_1(\alpha)}{p}$ satisfies both properties above which completes the proof. □

Lemma D.5. *The sequence $v_i^{\mathcal{C}}(x)$ converges in probability.*

Proof. Let the operator $A_{\mathcal{C}}$ and the map $b_{\mathcal{C}}$ be defined as

$$(A_{\mathcal{C}}v)(x) = \frac{\sum_{i=1}^m k(x,c_i)v(c_i)\gamma_i \mathbb{1}\left(c_i \in M \setminus (M_0 \cup M_1)\right)}{\sum_{i=1}^m \gamma_i k(x,c_i)},$$

and

$$b_{\mathcal{C}}(x) = \frac{\sum_{i=1}^m \gamma_i k(x,c_i)\mathbb{1}(c_i \in M_1)}{\sum_{i=1}^m \gamma_i k(x,c_i)}.$$

Here, we let γ_i denote the probability mass under μ of the cell corresponding to c_i (Note the difference with $\gamma_{i,n}$ defined in Definition D.2). Then by applying an proof analogous to that in Lemma C.8, we have that

$$\Pr_{S\sim\mu^n}\left[|(A_{\mathcal{C}}v + b_{\mathcal{C}})(x) - (A_{n,\mathcal{C}}u + b_{n,\mathcal{C}}) > \epsilon\right] < 4\exp\left(-Cn\epsilon^2\right).$$

Applying an argument analogous to that in Lemma C.9 finishes the proof. □

Finally, Theorem 6.8 is a direct consequence of the previous two lemmas.

E Properties of the region-based effective resistance

E.1 The reduced graph

[15] offer an interpretation of the effective resistance between sets X_a, X_b on a graph G as the effective resistance between two aggregated nodes a, b on a reduced graph G_{ab}. We extend the concept of a reduced graph to more than two nodes by considering the reduced graph corresponding to the sets X_a, X_b, X_z.

We define $P_{abz} = \begin{bmatrix}\mathbb{1}_n(X_a) & \mathbb{1}_n(X_b) & \mathbb{1}_n(X_z) & I_c\end{bmatrix} \in \mathbb{R}^{n\times(c+3)}$, where $\mathbb{1}_n(X_p) \in \mathbb{R}^n$ is the indicator vector

$$\mathbb{1}_n(X_p) = \begin{cases}1, & \forall i \in X_p \\ 0, & \text{otherwise}\end{cases} \tag{E.1}$$

and $I_c \in \mathbb{R}^{n\times c}$ is the matrix collecting all basis vectors e_i for $i \in X_c$.

Definition E.1. *(Reduced graph) We define the reduced graph of G, with respect to the non-empty disjoint subsets $X_a, X_b, X_z \subset X$, to be the graph G_{abz} with Laplacian $L_{G_{abz}} := P_{abz}^\top L_G P_{abz}$.*

The relation between the reduced graph G_{abz} and G is given by Lemma E.2. We note that the interpretation offered by Lemma E.2 will be used in our experiments presented in Section 7, where the degree of the source and sink sets will be taken as the degree of the corresponding aggregated nodes.

Lemma E.2. *The reduced graph G_{abz} corresponds to reducing the nodes in G contained in each of the subsets X_p for $p \in \{a,b,z\}$ into corresponding aggregated nodes a, b, and z that satisfy the following properties for each aggregated node.*

- *The degree D_p of each aggregated node p is $D_p = \sum_{i\in X_p} D_i - \sum_{i\in X_p}\sum_{j\in X_p, j\neq i} W_{ij}$*
- *All external edges from each of the sets X_p to the complement $X\backslash X_p$ are preserved such that*

- *The edge weights of each aggregated node p to a node $i \in X_c$ are given by $W_{pi} = \sum_{j \in X_p} W_{ji}$*
- *The edge weights of each aggregated node p to another aggregated node $q \neq p$ is $\sum_{j \in X_p} \sum_{j \in X_q} W_{ji}$*

Proof. We write $L_G P_{abz}$ in block form $L_G P_{abz} = \begin{bmatrix} L\mathbb{1}_n(X_a) & L\mathbb{1}_n(X_b) & L\mathbb{1}_n(X_z) & LI_c \end{bmatrix} \in \mathbb{R}^{n \times (c+3)}$. This gives

$$L_{G_{abz}} = P_{abz}^\top L P_{abz} = \left[\begin{array}{ccc|c} L_{aa} & L_{ab} & L_{az} & L_{ac} \\ L_{ba} & L_{bb} & L_{bz} & L_{bc} \\ L_{za} & L_{zb} & L_{zz} & L_{zc} \\ \hline L_{ca} & L_{cb} & L_{cz} & L_{cc} \end{array}\right]$$

where $L_{pq} = \sum_{i \in X_p} \sum_{j \in X_q} (L_G)_{ij}$ for $p, q \in \{a, b, z\}$, $L_{cq}(i) = \sum_{j \in X_q} (L_G)_{ij}$ for $i = 1, \cdots c$ and $L_{cc} \in \mathbb{R}^{c \times c}$ is the Laplacian on X_c. Now $(L_G)_{ij} = -W_{ij}$ for $i \neq j$, where W_{ij} is the edge weight between node i, j, and $L_{ij} = D_i$ for $i = j$, where D_i is the degree of node i. The interpretation of the blocks follows. □

Lemma E.3. *The effective resistance between the nodes $p, q \in \{a, b, z\}$ in the graph G_{abz} corresponds to the effective resistance between the corresponding non-empty disjoint sets X_p, X_q in the graph G such that $R_{G_{abz}}(p, q) = R^s_G(X_p, X_q)$.*

Proof. From Theorem E.5, Definition E.1 and Lemma E.2 it follows that

$$R^s_G(X_a, X_b) = (e_1 - e_2)^\top (P_{abz}^\top L_G P_{abz})^\dagger (e_1 - e_2) = (e_1 - e_2)^\top (L_{G_{abz}})^\dagger (e_1 - e_2) = R_{G_{abz}}(p, q).$$

□

E.2 Proof of Theorem 5.1 Triangle inequality for ER between sets

Consider the setting in Section 2.2. The effective resistance between sets can be calculated using the voltage difference formulation of Proposition 2.3, by extending the source and sink constraints to the subsets. We require $\sum_{i \in X_s} J_i = 1$ and similarly $\sum_{i \in X_g} J_i = -1$. With $J_i = (Lv)_i$ from Eq. (2.3) we have $\sum_{i \in X_s} (Lv)_i = 1$ and $\sum_{i \in X_g} (Lv)_i = -1$. Proposition E.4 defines the new optimization problem we need to solve.

Proposition E.4 (Voltage difference formulation on sets)**.** *The effective resistance between the non-empty disjoint subsets $X_s, X_g \subset X$ corresponds to $R^s(X_s, X_g) = E(\nu)$, where ν is the voltage that minimizes the energy*

$$\begin{aligned} \min_v \quad & \sum_{x_i, x_j \in X} W_{i,j}(v(x_i) - v(x_j))^2 \\ \textit{Subject to} \quad & \sum_{i \in X_s} (Lv)_i = 1, \quad \sum_{i \in X_g} (Lv)_i = -1, \quad (Lv)_i = 0, \forall i \in X_c \end{aligned}$$

Proof. Theorem 2 in [15]. □

The next theorem extend the result in Theorem 2, Song et al. [15] to an additional subset $X_z \subset X$. Where X_z is non-empty and disjoint from X_a, X_b.

Theorem E.5. *The effective resistance from Proposition E.4 can be written as*

$$R^s_G(X_a, X_b) = (e_1 - e_2)^\top (P_{abz}^\top L_G P_{abz})^\dagger (e_1 - e_2)$$

where $e_i \in \mathbb{R}^{c+3}$ is a basis vector with 1 at i and zero otherwise.

Proof. Using P_{abz}, we can write the constraints in Proposition E.4 in a more compact form $P_{abz}^{\top}L = (e_1 - e_2)$. Applying these constraints with the Lagrange multiplier $\gamma \in \mathbb{R}^{c+3}$ we have

$$f(v, \gamma) = v^{\top}Lv + \gamma(P_{abz}^{\top}L - e_1 + e_2)$$

Since the optimization problem is convex, the effective resistance can be found by solving

$$R_G^s(X_a, X_b) = \min_v \max_\gamma f(v, \gamma)$$

The remaining steps can be found in the proof of Theorem 2, Song et al. [15]. □

Having established the setting, we can move on to the main proof.

Proof. Consider the reduced graph G_{abz} of G from Definition E.1. In this reduced graph the current is injected into the nodes aggregated in a and is extracted from the nodes aggregated in b. We denote the total current y_{ab}. From the conservation of current, we must have that the current y_{ak} and y_{bk} through any internal node $k \in X_c \cup \{z\}$ must satisfy $y_{ab} \geq y_{ak}$ and $y_{ab} \geq y_{bk}$. Let $v(i)$ be the voltage at node i. For the current to flow from a to b, we need the voltage at the nodes to satisfy $v(a) > v(k) > v(b)$ for all $i \in X_c \cup \{z\}$.

The effective resistance of the graph G_{abz} is an intrinsic property of the graph and does not depend on the choice of source and sink. We let a be the source and b the sink. We then have

$$y_{ab} = \frac{v(a) - v(b)}{R_{G_{abz}}(a, b)} \geq \frac{v(a) - v(z)}{R_{G_{abz}}(a, b)} \geq y_{az} \quad \text{and} \quad y_{ab} = \frac{v(a) - v(b)}{R_{G_{abz}}(a, b)} \geq \frac{v(z) - v(b)}{R_{G_{abz}}(a, b)} \geq y_{bz}$$

This gives

$$\frac{v(a) - v(z)}{v(a) - v(b)} \leq \frac{R_{G_{abz}}(a, z)}{R_{G_{abz}}(a, b)} \quad \text{and} \quad \frac{v(z) - v(b)}{v(a) - v(b)} \leq \frac{R_{G_{abz}}(b, z)}{R_{G_{abz}}(a, b)}$$

Adding the two inequalities gives

$$R_{G_{abz}}(a, b) \leq R_{G_{abz}}(a, z) + R_{G_{abz}}(z, b)$$

Using Lemma E.3 we have that $R_{G_{abz}}(p, q) = R_G^s(X_p, X_q)$ for $p, q \in \{a, b, z\}$. The result follows. □

Paper IV

Structure from voltage

Robi Bhattacharjee, **Alexander Cloninger**,
Yoav Freund, **Andreas Oslandsbotn**

In preparation for submission

(Co-first-authors: Robi Bhattacharjee and Andreas Oslandsbotn)

(arXiv: doi.org/10.48550/arXiv.2203.00063)

Structure from Voltage

Robi Bhattacharjee[*] Alexander Cloninger[†] Yoav Freund[‡] Andreas Oslandsbotn[§¶]

Abstract

Data is often represented as point clouds embedded in high-dimensional space, with an intrinsic structure that concentrates on or close to lower-dimensional sets. Dimensionality reduction is a field dedicated to uncovering these lower-dimensional structures to mitigate the curse of dimensionality. Since the intrinsic structure can be highly non-linear. Non-linear dimensionality reduction techniques are necessary.

A prominent technique for non-linear dimensionality reduction is spectral graph embeddings such as Laplacian eigenmaps (LP). The challenge with these techniques is that they rely on calculating the eigenfunctions of a large matrix that scales with the number of data points. These operations are expensive and hard to parallelize and typically require the data to be stored in memory.

In this paper, we propose a scalable non-linear dimensionality strategy that is easy to parallelize. The approach we propose is based on the notion of a localized voltage function defined on a graph constructed on the point cloud. This is closely connected to the strategy used to define the effective resistance (ER) on graphs. Unfortunately, it has been shown that when vertices correspond to a sample from a distribution over a metric space, the limit of the ER between distant points converges to a trivial quantity that holds no information about the graph's structure.

To circumvent this, we propose the notion of *grounded resistor graphs*, in which the source vertex in ER is replaced with a source region, and the sink vertex is replaced with a universal ground vertex that is connected to all points by a fixed constant resistor. We then show that the energy-minimizing voltage over this new construction converges towards a non-trivial solution in the large sample limit. These voltage solutions are both localized near their source regions and can be solved independently. Finally, we present preliminary results, both theoretical and numerical, demonstrating how these solutions can be used to provide low-dimensional embeddings for the underlying space.

1 Introduction

Dimensionality reduction is the problem of finding a low dimensional representation for locations in a point cloud. The importance of finding efficient algorithms to solve this problem has increased with the availability of datasets with millions of data points and thousands of dimensions.

Principal component analysis (PCA) is a very effective and efficient algorithm for *linear* dimensionality reduction. In particular, PCA depends on the shape of the point cloud only through the covariance matrix, and the transformations it produces are linear. However, large and complex point clouds typically exhibit non-linear structures, which makes traditional PCA insufficient. Because of this, the field of *non-linear* dimensionality reduction (NLDR) has emerged as a very active and productive area of research (see Section 1.1).

[*] Department of Informatics, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, United States

[†] Department of Mathematics, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, United States

[‡] Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, United States

[§] Department of Informatics, University of Oslo, Problemveien 7, 0315 Oslo, Norway

[¶] Simula Research Laboratory, Kristian Augusts gate 23, 0164 Oslo, Norway

One approach to NLDR is to view the point cloud as a sample from an underlying distribution and to characterize the Laplace-Beltrami operator (LBO) over that distribution. Two prominent representatives of this line of work are Eigen maps [1] and Diffusion maps [2], both of these algorithms are based on finding the eigenfunctions of the LBO.

Computing these eigen-functions for a point cloud of size n requires computational time of $O(n^3)$ on one computer and is hard to parallelize. As a result, finding these vectors for point clouds larger than 100,000 is impractical with current computers. Furthermore, typical implementations of the calculations of these vectors require *all* of the data to be stored in a single computer. The underlying reason that all of the data has to be available at the same time is that typical eigen-functions are "global", by which we mean that for most eigen-functions f one can find pairs of points a, b such that the distance $d(a, b) > r$ for some large value of r and at the same time $|f(a)| > \gamma$ and $|f(b)| > \gamma$ for some large γ. In this work, we suggest characterizing the LBO on the point cloud using *local* functions instead. We say that a function g is local if for any pair of points a, b such that $d(a, b) > r$ either $|g(a)| < \epsilon$ or $|g(b)| < \epsilon$ for some small ϵ.

The eigenvectors of the LBO are solutions of *homogeneous* systems of equations. Meanwhile, to construct *local* functions, one needs *non-homogeneous* solutions, i.e., pointwise constraints. To help our intuition, we follow Doyle and Snell [3] in using resistor circuits to model the graph Laplacian. In this setting, one can define the non-homogeneous constraints by choosing two vertices: a "source vertex" whose voltage is fixed to one, and a "sink vertex" whose voltage is fixed to zero. The remaining vertices are "floating" or "unconstrained" and the voltages on those nodes are set by minimizing the energy dissipated by the circuit.

The minimal energy solution determines the current flowing from the source to the sink. In turn, the reciprocal of the current determines the *effective resistance* (ER). The ER is well-defined for discrete graphs. However, it becomes trivial and uninformative when considering point clouds in a metric space and letting the number of points increase to infinity, as shown by Von-Luxburg et al. [4]. The first contribution of this paper is to show that this problem can be alleviated by considering the effective resistance between pairs of small *regions* rather than pairs of *points*. Keeping the regions fixed as the number of points increases to infinity produces non-trivial limits that can be used for NLDR.

A natural approach at this point is to use the voltage functions for different source-sink location pairs as an alternative to the eigenvector representation. Unfortunately, the resulting voltage functions are not local if the sink and the source are far from each other. To ensure that the functions are local, we introduce a particular form of the resistor graph, namely the "grounded resistor graph". A resistor graph is transformed into a grounded resistor graph by adding a single node, the "ground", connected to each node in the original graph. The voltage functions are generated by selecting a source and using the ground as the sink. The result is voltage functions that are one at their respective source and strictly decreasing away from it.

We consider sources to be *landmarks* and use the associated voltage function to measure the divergence from the landmark [1]. We propose using the divergences from a small fixed set of landmarks as a dimensionality-reducing mapping. To represent a non-landmark location, we measure its divergence from each landmark, and that set of distances uniquely identifies the location of the point. Moreover, If the point cloud lies on a k-dimensional manifold, then the divergence to the nearest $k + 1$ locations uniquely identify the location. Thus the identity of the close landmarks, together with the distances from those landmarks, can be used as a low-dimensional representation of the location.

Our contributions can be summarized as follows:

- We alleviate the problem described by Von-Luxburg et al. [4] by appropriately scaling the resistances as the sample size grows.
- We prove the existence, convergence to a non-trivial limit, and shape properties of the grounded metric voltage function.

[1] This divergence is not a metric because it is not symmetric. On the sum of the divergence from A to B with the divergence from B to A *is* a metric, and this metric is the effective resistance in the grounded graph.

- We derive an analytical solution for the grounded metric voltage on the sphere and support these solutions by numerical experiments.
- We show the results of a few experiments on real-world data sets, including MNIST and Frey-face data sets.

The rest of the paper is organized as follows. In section 2, we introduce the notion of a grounded resistor graph. While section 3 extends this to the metric graph and the grounded resistor graph on metric spaces. In section 4, we discuss the limitations of LE and ER in the large-data metric-graph limit and compare them to our method, which overcomes these limitations. In section 5, we theoretically analyze the grounded metric graph and show existence, convergence, and shape properties of the voltage solution. In section 6 we show theoretically how the voltage solution can be used to embed the sphere and support our result with numerical experiments. In Section 7 we discuss computational advantages with the metric voltage function while in Section 8 we conclude and discuss future work.

1.1 Relations to other work

In this section, we give a brief overview of related work.

Related work on non-linear dimensionality reduction

The published literature on NLDR is vast; see the survey [5]. Below is a non-exhaustive list of some of the significant approaches.

- **Kernel based** The kernel-PCA [6] method generalizes the classical linear PCA to non-linear problems.
- **Manifold based** methods rely on the assumption that the point cloud lies on a low dimensional manifold and use ideas from differential geometry. These include ISOMAP [7] that uses shortest paths as an approximation to geodesics, and Locally linear embeddings [8] uses local linear approximations of manifold.
- **Optimization based** Direct optimization method use gradient descent algorithms to improve an initial embedding. These include t-SNE [9], UMAP [10], force-based algorithms [11], and LDLE [12].
- **Laplacian based** are based on the Laplace-Beltrami operator. These include Laplacian eigenmaps [1, 13] and Diffusion maps [2]

There are various overlaps and combinations of these categories. For example, it has been shown that Laplacian based methods correspond to kPCA with particular choices of the kernel [14]. Similarly, Laplacian based methods have been used to initialize optimization based methods [15].

Related work on effective resistance

The ER was introduced as a distance function on graphs by Klein et al. [16]. Since then, it has proven a useful tool for capturing structural characteristics of graphs, with numerous applications, such as phylogenetic networks [17], detecting community structure [18], distributed control [19], graph edge sparsification [20], and measuring cascade effects [21] e.g. in power grids [22, 23, 24].

2 Resistor Graphs

In this section, we introduce the notion of a *grounded resistor graph*, which will serve as our fundamental tool for analyzing graphs along with the spaces from which they are sampled. We first review several key ideas and concepts about *resistor graphs.*

2.1 General Resistor Graphs

Let (X, W) be an undirected, weighted graph with nodes $X = \{x_1, \dots, x_n\}$, and edge weights $W_{i,j}$. Let the *voltage* v be a function $v : X \to \mathbb{R}$. The *energy* of the voltage v is defined as

$$E(v) \doteq \sum_{x_i, x_j \in X} W_{i,j}(v(x_i) - v(x_j))^2 = v^T L v \tag{2.1}$$

where $L = D - W$ is the Laplacian matrix, and D is a diagonal matrix such that $D_{ii} = \sum_j W_{ij}$.

It is easy to see that, with no constraints, the minimal energy is zero, and that zero energy is attained for any v where all of the entries are equal; $v(x_i) = v$ for all vertices in the graph. Therefore, to obtain a non-trivial voltage distribution on the graph, it is necessary to constrain the system. This motivates the Energy Minimizing Voltage (EMV), which is the main object we study in this paper.

Definition 2.1 (EMV)**.** *The energy-minimizing voltage of a weighted graph (X, W) with respect to source and sink nodes X^s and X^g, is the solution to the following optimization problem:*

$$\min_v \sum_{x_i, x_j \in X} W_{i,j}(v(x_i) - v(x_j))^2$$

$$\textit{Subject to} \quad v(x_i) = 1 \quad \forall x_i \in X^s, \quad v(x_i) = 0 \quad \forall x_i \in X^g,$$

where X^s are the source nodes and X^g the sink nodes.

The EMV owes its name to a helpful interpretation of the graph as an electrical network. In the following, we give a brief description of this view, while more details can be found in Doyle and Snell [3].

Electric networks: The EMV can be thought of as the voltage in an electrical network, where the graph corresponds to the underlying electric circuit. In this view, each vertex has an associated *voltage* $v(x_i)$ and each edge (x_i, x_j) has associated a non-negative *resistance* $R_{i,j} = \frac{1}{W_{i,j}}$ and a signed *current* $J_{i,j} = -J_{j,i}$. Furthermore, Ohm's law relates the current and resistance at an edge with the voltages at the vertices connected by the edge. The relation can be written as

$$v(x_i) - v(x_j) = R_{i,j} J_{i,j} \quad \text{or alternatively} \quad J_{i,j} = W_{ij}(v(x_i) - v(x_j)). \tag{2.2}$$

Meanwhile, from Kirchhoff's law, the sum of currents entering a node i must be zero, namely

$$\sum_{j \sim i} J_{i,j} = J_{ext,i}, \tag{2.3}$$

where $I_{ext,i}$ is an external current that can be either a source, a sink or zero if the node is un-constrained (no external source applied). Combining these laws we have that

$$(Lv)_i = \sum_{j \sim i} W_{ij}(v(x_i) - v(x_j)) = J_{ext,i} \tag{2.4}$$

from which it is easy to see that the EMV in Definition 2.1 can be attained. In particular, the constraints on the source and ground voltages can be enforced by the external current.

Because of this, we can interpret Equation (2.1) as the energy dissipated in a circuit in the form of heat when no power is injected. With no external source, it is clear that this energy is zero. Meanwhile, the constrained system in Definition (2.1) corresponds to a circuit connected to an external source for which a non-trivial minimal energy voltage vector exists.

2.2 Grounded Resistor Graphs

Computing the EMV over arbitrary choices of sources X^s and sinks X^g can reveal aspects of the global structure of a graph – for example, measuring the total current that flows from X^s to X^g can give a measure of the connectivity between these two sets. In this work, we are particularly interested in the case where X^s is a small set of closely connected vertices, and X^g is selected to reveal the *local* structure of the graph around X^s. Motivated by this, we introduce the idea of a *grounded resistor graph*, which replaces the set of sink nodes X^g with a dummy node g that represents a universal sink. We refer to this sink as the grounding node.

Definition 2.2 (Grounded resistor graph)**.** *Let (X, W) be a weighted graph. Then its grounded graph with grounded weight ρ, (X, W, ρ), is the graph in which the extra node, g, is connected to all vertices with an edge of weight ρ.*

Here, we don't consider g as a node of X as its behavior is completely determined by the weight ρ. As we will see later, it will be convenient to keep X, W unchanged when considering the grounded graph. Because everything is connected to the ground, the EMV over (X, W, ρ) will naturally decay to 0 on nodes far from the source. We define the grounded EMV as follows.

Definition 2.3 (Grounded EMV)**.**

$$\min_{v:X\to[0,1]} \sum_{i,j\in X} W_{ij}(v(x_i) - v(x_j))^2 + \sum_i \rho v^2(x_i)$$

$$\textit{Subject to} \quad v(x_i) = 1 \quad \textit{for all} \quad x_i \in X^s.$$

Here, the term $\sum_i \rho v(x_i)$ represents the amount of energy corresponding to the edges connecting each node to the ground vertex. Furthermore, we exclude the ground node from v because it is always defined to have a voltage of 0. For a given source X^s, we can describe the solution of the EMV using the *grounded weight matrix* $\widetilde{D}^{-1}\widetilde{W}^{(s)}$. Here $\widetilde{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $D_{ii} = 1$ for $x_i \in X^s$ and otherwise $\widetilde{D}_{ii} = \rho + \sum_{j=1}^{n} W_{ij}$. Furthermore, $\widetilde{W}^{(s)} \in \mathbb{R}^{n\times n}$ is defined as

$$\widetilde{W}^{(s)}_{ij} = \begin{cases} 1, & \text{if} \quad i = j, x_i \in X^s \\ 0, & \text{if} \quad i \neq j, x_i \in X^s \\ W_{ij}, & \text{otherwise.} \end{cases}$$

We note that the source is incorporated by $\widetilde{W}^{(s)}$ and $\widetilde{D}$ incorporates the effect of the grounding node through the additional ρ in the sum. With the definition of the *grounded weight matrix* we have from Lemma 2.4 that the solution to the grounded EMV can be written as:

Lemma 2.4. *The solution to the EMV in Def. 2.3 is the unique voltage function $v^* : X \to [0, 1]$, which satisfies $v^* = \widetilde{D}^{-1}\widetilde{W}^{(s)}v^*$ and $v^*(x_i) = 1$ for all $x_i \in X^s$.*

The motivation for writing the EMV in this way will become clear later when we introduce the metric

Remark 2.5. *We note that the sum of the i-th row of $\widetilde{W}^{(s)}$ is $1/(1 + \rho/\sum_i W_{ij})$ where $1/\rho$ is the resistance to ground. This relation highlights the importance of the ground node because $\rho/\sum_i W_{ij} \approx 0$ means a trivial solution since the rows sum to one. Therefore, as the large graph limit means increasing $\sum_i W_{ij}$, tuning of ρ can be used as a counterweight. Furthermore, as we will show, the voltage decay away from the source is tightly linked to the magnitude of ρ.*

3 Grounded Resistor Graphs over Metric Spaces

We focus on a special type of graphs, namely graphs constructed from samples drawn from a distribution over a metric space. Let (M, d) be a compact metric space and μ the probability measure over M. Let the kernel function $k : M \times M \to [0, 1]$ be a function that defines what it means for two points to be "near" each other. Two commonly used kernel functions are:

- The radial kernel: $k_r(x,y) = \mathbf{1}(d(x,y) \leq r)$ where $r > 0$ is some fixed radius.
- The Gaussian kernel: $k_\sigma(x,y) = \exp(\frac{-d(x,y)^2}{2\sigma^2})$, where $\sigma > 0$ is the fixed temperature parameter.

Let $X_n = \{x_1, \cdots, x_n\}$ be a sample of points $x_i \in M$ drawn i.i.d. from μ. Our main idea is to construct a weighted grounded graph, (X_n, W, ρ) by connecting points x_i, x_j by weights $W_{ij} \propto k(x_i, x_j)$, and by utilizing a grounded weight proportional to ρ. Then, if $M^s \subseteq M$ is a local region in M, we can leverage the EMV over the grounded graph to understand the structure of M nearby M^s.

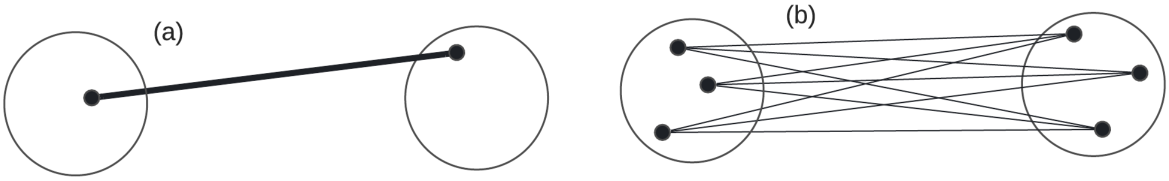


Figure 1: Example of resistance scaling. **(a)** Number of edges connecting T and T' for $m = 1$ point sampled from each region. **(b)** Number of edges connecting T and T' for $m = 3$ points sampled from each region. Notice that the number of edges between these regions is m^2.

To do so, it is essential that our computations converge towards a non-trivial solution as the number of sampled points, n, goes towards infinity. In particular, it is necessary for W, ρ to appropriately scale as n changes. It is therefore natural to demand that the physical properties of the graph, embodied by the resistance, current and voltage, should remain relatively stable as n increases. Thus, it is crucial to understand how the edge resistances should scale with the number of points sampled.

Scaling based on regional density To this end, consider two small regions T and T' such that $k(x, x') > 0$ for all $x \in T$ and $x' \in T'$. Note, these are not necessarily sources and sinks, just two small regions. For simplicity let $k(x, x')$ be constant, which means each edge has equal resistance $R = 1/k(x, x')$. Our goal is to keep the resistances between these two regions constant as the number of points changes. For a fixed X_n, on average there are m points $x \in S_n = \{x \in X_n : x \in T\}$ and m points $x' \in S'_n = \{x \in X_n : x \in T'\}$. This results in m^2 edges between T and T'. This means the total resistance between these regions is R/m^2, given that these edges are connected in parallel.

The issue here is that, once we move to a denser sample X_{2n} there will be, on average, $2m$ points in S_{2n} and S'_{2n} respectively. This will create a net resistance $\frac{1}{4}R/m^2$, which means the resistance between these physical regions T, T' is decreasing and will go to 0 as n goes to infinity. We illustrate this construction in Figure 1.

Based on these considerations, we formally define the grounded metric resistor graph, illustrated in Fig. 2.

Definition 3.1 (Grounded metric resistor graph). *Let $X_n = \{x_1, x_2, \ldots, x_n\} \sim \mu$ be a set of points sampled from data distribution μ over M, $k : M \times M \to [0,1]$ be a kernel similarity function, and ρ_g be a fixed scaling constant for the grounded weight. Then the* ***grounded metric resistor graph****, (X_n, W, ρ), is the weighted graph defined with grounded weight, $\rho = \frac{\rho_g}{n}$, and edge weights $(W)_{ij} = \frac{k(x_i, x_j)}{n^2}$.*

Next, we can define the grounded EMV for a region M^s by considering all points inside M^s as sources.

Definition 3.2 (Grounded metric voltage function). *Let $M^s \subseteq M$ be any set, ρ be a constant, and (X_n, W, ρ) be the corresponding grounded resistor graph constructed from $X_n \sim \mu^n$. We define the* ***grounded metric voltage function*** *with respect to M^s, denoted $v_n^* : X_n \to [0,1]$, as the grounded EMV with respect to $X^s = M^s \cap X_n$.*

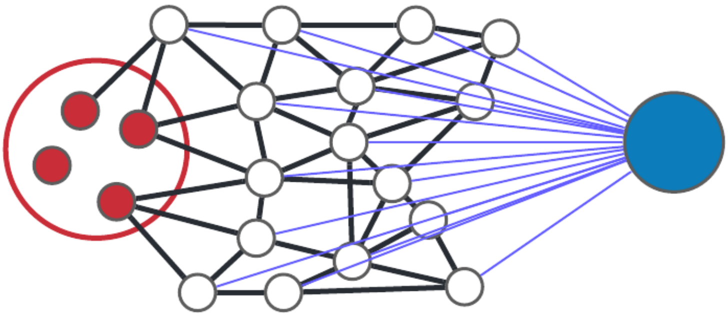


Figure 2: An example of a grounded graph with red nodes denoting X^s and the blue node g for the terminating ground node. Note that all points are connected to the ground.

4 Outlining our method and comparing to existing work

We propose to use the grounded voltage function from Definition 3.2 as an embedding tool. The idea is to distribute sources (landmarks) across the point cloud we want to embed and, for each source, solve for an independent grounded voltage function. With d-independent voltages, we can construct a d-dimensional embedding, a strategy we call Localized voltage eigenmaps (LVE).

In the remainder of this paper, we will establish the theoretical foundations for LVE. However, before we do this, we will outline two existing approaches, namely the effective resistance (ER) and the Laplacian eigenmaps (LE), and explain how our method improves on these schemes.

4.1 Effective resistance

The ER is a measure for calculating distances on graphs [16]. It arises naturally from the electrical system interpretation by considering each node pair (l, k) as a source-sink connection and enforcing a unit current between them. The standard formulation of ER follows by enforcing these constraints on Equation (2.4), which gives

$$R_{lk}^{eff} = v(x_l) - v(x_k) = (e_l - e_k)^\top L^\dagger (e_l - e_k) = \|z_l - z_k\|^2, \tag{4.1}$$

where the effective resistance R_{lk}^{eff} is an euclidean distance matrix [25]. The last term in Eq. (4.1) shows how the ER can be considered an n-dimensional embedding with $z_l = \Lambda^{-1/2} V^\top e_l$ for $L = V \Lambda V^\top$. The advantage of ER is that the distance between nodes can be computed without explicitly calculating the voltage functions v. Instead, it suffices to solve for $L^\dagger$, the pseudo-inverse of L. Methods have also been proposed for distributed computation [26].

Effective resistance as an EMV We can relate the ER to the grounded voltage function by reformulating the ER as the solution to an EMV optimization problem. For a node pair (l, k), the voltage function defining the effective resistance $R_{lk}^{eff} = v(x_l) - v(x_k)$ is the solution to the following minimization problem.

$$\begin{gathered} \min_{v: X \to \mathbb{R}} \quad \sum_{i,j} W_{ij}(v(x_i) - v(x_j))^2 - \sum_{i \in (l,k)} J_i v(x_i) \\ \text{Subject to} \quad J_i = 1;\ i = l, \quad J_i = -1;\ i = k \quad \text{and} \quad J_i = 0 \quad \text{otherwise.} \end{gathered} \tag{4.2}$$

By considering Ohm's law $J_{ij} = (v_j - v_i)/R_{ij}$ with $v_j = 0$, $R_{ij} = \rho$ and summing over all nodes instead of (l, k), the relation to the grounded EMV becomes clear.

Limitations with the effective resistance The problem with the ER, demonstrated by [4], is that the distance between nodes connected by a path converges in the large graph limit to a trivial quantity which only depends on the degree of the end-nodes.

4.2 Laplacian eigenmaps

Laplacian Eigenmaps is a widely used tool for finding a $d < n$ dimensional representation of the n nodes on a graph. The goal is to assign coordinates $z_k \in \mathbb{R}^d$ to each node k, so nearby points on the graph remain close. This is done by the following: for $l = 1, \ldots, d$ solve

$$\begin{aligned} \min_{v^{(l)}:X\to\mathbb{R}} \quad & \sum_{i,j} W_{ij}(v^{(l)}(x_i) - v^{(l)}(x_j))^2 \\ \text{Subject to} \quad & v^{(l)} \perp v^{(l')} \quad \text{and} \quad v^{(l)} \perp 1. \end{aligned} \tag{4.3}$$

To avoid the trivial solution $v^{(l)} = 1$, the constraint $v^{(l)} \perp 1$ is enforced. Meanwhile, the orthogonality constraint $v^{(l)} \perp v^{(l')}$ is introduced to avoid solving for the same function d times. It turns out that the minimizing set, satisfying these constraints, are the first d eigenfunctions of L, modulo the first, which turns the optimization into an eigenvalue problem.

Laplacian eigenmaps as an electrical network The eigenfunctions that solves the LE problem satisfies trivially $L^{(l)} = \lambda_l D v^{(l)}$. By thinking of the right-hand side as an external current $J_{ext,l} = \lambda_l D v^{(l)} = \lambda_l(d_1 v^{(l)}(x_1), \ldots, d_n v^{(l)}(x_n))^\top$ we can think of each $v^{(l)}$ as a voltage over the nodes, that satisfies Kirchhoff's and Ohm's law through Equation (2.4).

This means that the LE optimization problem in Equation (4.3) can be considered an EMV defined over an electrical resistor graph, where constraints are: no current accumulation due to $v^{(l)} \perp 1$ and orthogonality between each voltage solution. Furthermore, the interpretation of the external current $J_{ext,i}$ is that for each voltage function $v^{(l)}$, the nodes in the graph can act as a source or a sink depending on the sign of $v^{(l)}(x_k)$ at node k. In particular, we note that there are no constraints on the locality of these sources and sink nodes.

Limitations with Laplacian eigenmaps An important limitation of LE is the orthogonality condition, which prevents the eigenvectors from being computed independently, preventing distributed computation. Furthermore, the Laplacian eigenvectors are typically global and, therefore, expensive to compute. By global, we mean in this context that the eigenfunctions have non-zero support almost everywhere on the graph, meaning we effectively need all nodes to compute the eigenfunctions. We can understand this global behavior from the electrical network interpretation because, as we have seen, the source and sink nodes are not confined to specific regions of the graph, and because of this, neither are the voltage solutions.

4.3 Advantages of the grounded metric voltage function

In this paper, we are motivated by making an embedding for metric spaces, which requires a computationally cheap and non-trivial solution in the large graph limit. As we have seen, both LE and ER face limitations in this limit; The ER suffers from a trivial solution; Whereas the LE is prevented from distributed computations and suffers from high computational complexity. Because of these limitations, both LE and ER are in their traditional form, prevented from being used in the metric graph setting. In this paper, we show that embedding using the grounded voltage function has the potential to overcome these limitations.

Overcoming the limitations of LE As we show in Section 5.2, combining a localized source with a universal ground creates a localized voltage solution, reducing computational complexity. Furthermore, the grounded voltage functions are free of dependency conditions, such as the orthogonality condition enforced by LE, allowing distributed computations.

Overcoming the limitations of ER As apparent from Eq. (4.2) there is a close connection between the ER and the grounded EMV. Therefore, one could expect a trivial limit also for the grounded voltage function. However, as we show in Sections 5.1 and 5.3, the grounded metric voltage function converges to a non-trivial limit as the number of samples goes to infinity. This non-trivial limit is achieved because we think of each node in the graph as a region with a density instead of individual points, a view discussed in detail in Section 3. In particular, this view introduces region-based scaling and sources that cover a finite region instead of the point sources used in the ER setting.

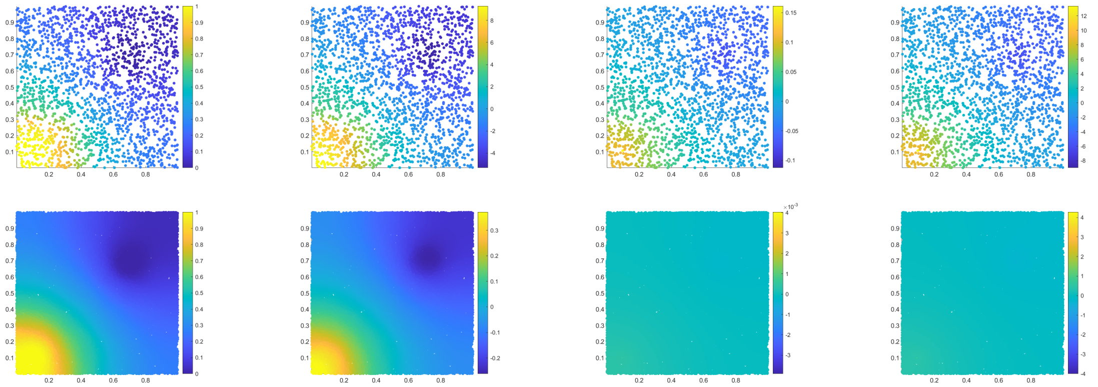
Figure 3 illustrates the difference between using a point source and a regional source whose strength increases proportionally with the number of samples. We demonstrate this with four algorithms across different numbers of points sampled from the same metric space. All of these are done without the universal grounding node but with a source at $s = (0.1, 0.1)$ and a sink at $g = (0.7, 0.7)$. Both source and sink regions are of radius 0.1. We compute voltage curves through the following four methods: (PM) the power method described in Lemma 2.4; (RegionER) ER using a source vector e_{X^s} where the voltage curve is

$$v = L^{\dagger}\left(e_{X^s} - e_{X^g} - (p_s - p_g)\right), \quad e_{X^s}(x) = \begin{cases} 1, & \text{if } x \in X^s \\ 0, & \text{else} \end{cases}, \quad e_{X^g}(x) = \begin{cases} 1, & \text{if } x \in X^g \\ 0, & \text{else} \end{cases},$$

where $p_s = \frac{1}{|X|}\sum_{x \in X} e_{X^s}(x)$ is the density of the source, same for p_g and the sink. The mean is subtracted so that the total external current is mean 0; (DensityER) ER using an indicator source vector e_s localized at the exact source node $s \in X^s$, an equivalent sink vector e_g, and the voltage curve

$$v = L^{\dagger}\left(p_s \cdot e_s - p_g \cdot e_g - (p_s - p_g)\right),$$

where the mean is subtracted so that the total external current is mean 0; (ER) standard ER using the indicator source vector e_s, the equivalent sink vector e_g, and the voltage curve $v = L^{\dagger}(e_s - e_g)$. It is clear from Figure 3 that using a source region instead of a source point (even if that point is weighted by the local density) is critical to attaining a nontrivial limit as the number of points increases.



PM RegionER DensityER ER

Figure 3: Trivial Limit of ER: (top) 2^{11} points, (bottom) 2^{15} points.

Finally, Figure 4 summarizes our contribution compared to the LE and ER schemes.

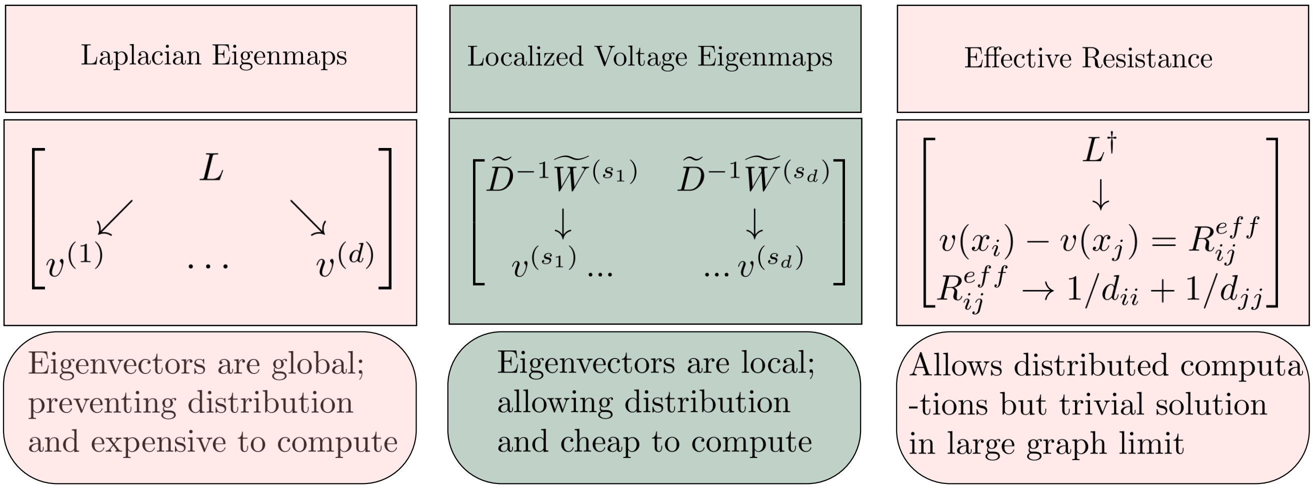

Laplacian Eigenmaps
Localized Voltage Eigenmaps
Effective Resistance
L
$v^{(1)}$ ... $v^{(d)}$
$\widetilde{D}^{-1}\widetilde{W}^{(s_1)}$ $\widetilde{D}^{-1}\widetilde{W}^{(s_d)}$
$v^{(s_1)}$ ... ... $v^{(s_d)}$
$L^\dagger$
$v(x_i) - v(x_j) = R_{ij}^{eff}$
$R_{ij}^{eff} \to 1/d_{ii} + 1/d_{jj}$
Eigenvectors are global; preventing distribution and expensive to compute
Eigenvectors are local; allowing distribution and cheap to compute
Allows distributed computa -tions but trivial solution in large graph limit

Figure 4: Illustration of our contribution

5 Analysis of the grounded metric voltage function

Let v_n^* be the grounded metric voltage function from Def. 3.2. In this section, we show that this voltage function converges to a non-trivial function as the sample size increase and provide shape bounds on the voltage decay away from the source.

5.1 Convergence of the grounded metric voltage function

We start by showing that $v_1^*, v_2^*, \ldots$ converge as our sample size increases. Our first step is to define the voltage function formally, $v^* : M \to \mathbb{R}$, that they converge to. To this end, we start with an explicit expression for v_n^* which follows directly from Lemma 2.4.

Proposition 5.1. *Let $M^s \subseteq M$ be a source subset, $X_n \sim \mu^n$ be a finite sample, $X^s = M^s \cap X_n$ be the set of source nods, and ρ be a scaling constant. Let v_n^* be the induced grounded metric voltage over (X_n, W, ρ). Then, for all $x_i \in X_n$, we have*

$$v_n^*(x_i) = \frac{\sum_{x_j \in X^s} k(x_i, x_j) + \sum_{x_j \in X_n \setminus X^s} v_n^*(x_j) k(x_i, x_j)}{\rho + \sum_{\mathbf{x}_j \in X_n} k(x_i, x_j)},$$

along with $v_n^*(x_i) = 1$ for $x_i \in X^s$.

Proof. Follows from Lemma 2.4, by writing each element in the voltage vector explicitly, for a given choice of kernel k as weights.

□

Proposition 5.1 suggests a natural limit object for v_n^*.

Theorem 5.2. *Let M be a metric space, and M^s denote a measurable set of source vertices. Let $\rho > 0$ be a scaling constant. Then there exists a unique map $v^* : M \to [0, 1]$ such that $v^*(x) = 1$ for all $x \in M_1$ and the following holds for all $x \in M \setminus M^s$:*

$$v^*(x) = \frac{\int_{M^s} k(x, y) d\mu(y) + \int_{M \setminus M^s} v^*(y) k(x, y) d\mu(y)}{\rho + \int_M k(x, y) d\mu(y)}.$$

Finally, to prove convergence, we must extend our solutions, v_n^* over grounded metric resistor graphs to the entire metric space M.

Definition 5.3. *Let v_n^* be the EMV over grounded graph (X_n, W, ρ). For all $x \in M$, we define the* ***extension*** *of v_n^* as the map $v_n^* : M \to [0,1]$ satisfying the following: $v_n^*(x) = 1$ if $x \in M^1$, and otherwise,*

$$v_n^*(x) = \frac{\sum_{i=1}^n k(x,x_i) v_n^*(x_i)}{\sum_{i=1} k(x,x_i)}.$$

Theorem 5.4. *Fix ρ and $M_1 \subseteq M$. Let v_n^* denote the extension of the EMV over the grounded graph, (X_n, W, ρ), and v^* denote the limit object described in Theorem 5.2. Then for any $x \in M$, the sequence $v_1^*(x), v_2^*(x), \ldots$ converges to $v^*(x)$ in probability (taken over the randomness of sampling $X_n \sim \mu^n$).*

5.2 Bounding the shape of the voltage function

We have shown that the grounded metric voltage converges to a non-trivial function in the large sample limit. In this section, we want to gain insights into the shape of this voltage function. Throughout this analysis we assume the radial kernel $k(x,y) = d(||x-y|| \leq r)$, and restrict the analysis to the unit sphere S^{d-1}. Furthermore, we assume a uniform density and use the Lebesgue measure, $\mu(A) = vol(A)$ to have a distribution to draw from.

We want to build a grounded metric graph on the sphere. Suppose the source region M_1 consists of the density contained in a ball $B(x_s, r_s)$ of radius r_s centered on the source landmark x_s, where x_s is a point on the sphere. We can then construct a grounded metric resistor graph as described in Def 3.1, with resistance to ground ρ. In the previous sections, we showed that a non-trivial voltage function $v^* : S^{d-1} \to [0,1]$ exists for this setting. For our configuration, we denote this function λ. When the dimension d is fixed, this function is determined by three parameters, namely the source radius r_s, the kernel radius r, and resistance to ground ρ. In this setting, we have the following bounds on the shape of the grounded metric voltage function:

Theorem 5.5. *Let $z := d_M(x_s, x) = \arccos(\langle x_s, x\rangle)$ be the geodesic distance from the source landmark x_s to a point x in the unit sphere. Furthermore, let $\phi(r) := d_m(x_c, x)$ be the arch-length (geodesic) between two points that have euclidean distance r, where r is also the radius of the kernel k used to construct the grounded metric graph. Let r_s, r, ρ be fixed. It then exists a unique map $\lambda : S^{d-1} \to [0,1]$ satisfying the following properties:*

$$\lambda(x) = \frac{\int_{B(0,r_s)} k(x,y) d\mu(y) + \int_{S^{d-1}} k(x,y) \in d(y \in S^{d-1} \setminus B(0,r_s)) \lambda(y) d\mu(y)}{\rho + a}$$

for $x \notin B(0,r)$ where a denotes the volume of the ball of radius r, $\lambda(x) = 1$ for $x \in B(0,r)$ and for $t \in [1,\infty)$

- *λ is radially symmetric and monotonically decreasing in distance outside of $B(0,r_s)$. In particular, there exists a function h such that $\lambda(x) = h(z)$.*
- *(Upper Bound) For $z \geq 2r$, $h(z + t\phi(r)) \leq \exp(-t \ln(1 + 2\rho/a))$.*
- *(Lower Bound) There exists a constant Γ s.t. $h(z + t\phi(r/2)) \geq \exp(-t \ln((a+\rho)/\Gamma)$*

Theorem 5.5 shows that the voltage function λ is essentially bounded between two functions that exponentially decay with respect to $z = d_M(x_s, x)$, the geodesic distance from x to the source landmark located at x_s. We also note that the result in Theorem 5.5 can easily be translated to a Disk in $\mathbb{R}^d$ by replacing the geodesic distance with the euclidean distance. This holds because the symmetry arguments used to derive the result for a sphere are also true for a disk. Corollary 5.6 summarize this result.

Corollary 5.6. *For a Disk in $\mathbb{R}^d$ we have the result in Theorem 5.5, with the geodesic replaced by the euclidean distance, s.t. $z = d(x_s, x)$, and the following bounds on h:*

- *(Upper Bound) For $z \geq 2r$, $h(z + tr) \leq \exp(-t \ln(1 + 2\rho/a))$.*
- *(Lower Bound) There exists a constant Γ s.t. $h(z + tr) \geq \exp(-2t \ln((a+\rho)/\Gamma)$*

5.3 Examples

We demonstrate the convergence of the grounded metric graph voltage on several basic examples in Figure 5.

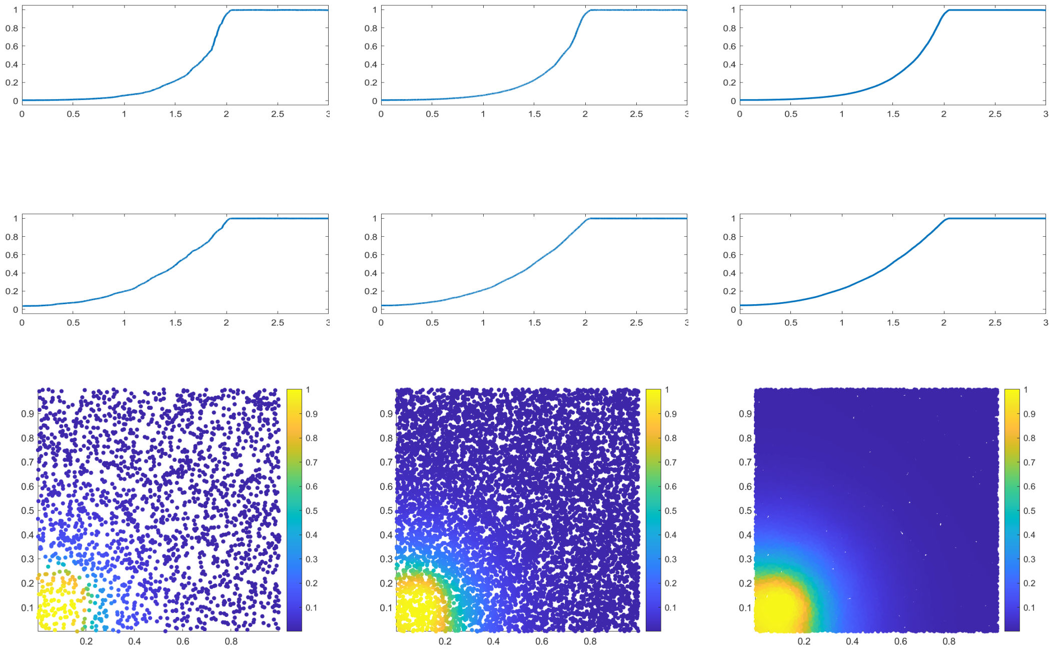


Figure 5: Grounded metric graph voltage. **Top:** X_1 on $[2,3]$ with large ρ to encourage fast decay. **Middle:** X_1 on $[2,3]$ with small ρ to encourage slow decay. **Bottom:** X_1 centered at $(.1,.1)$ with radius 0.1 with large ρ. **Left:** $n=2^{11}$, **Center:** $n=2^{13}$, **Right:** $n=2^{15}$

The first is on the 1D line $[0,3]$, where X_1 is a source of radius 0.5 on the interval $[2,3]$, and the sink X_0 is of radius 0.5 on the interval $[0,1]$. The second example is on the 2D unit square, with the source and sink on opposite corners of the square. In both cases, the Laplacian is formed with radius $r=0.05$, and the number of points range from $2^{11}, 2^{13}, 2^{15}$. We also vary ρ to demonstrate the effect of the ground radius on the speed of decay of the voltage. In all cases, the voltage function is strictly non-increasing away from the source.

6 Embedding using localized voltage functions

In this section, we show how the grounded metric voltage function from Def. 3.2 can be used as an embedding tool for various manifolds. The experiments in section 5.3 along with Thm. 5.5 demonstrate, for a unit sphere S^{d-1} and a disk in $\mathbb{R}^d$, a consistent decay of the voltage solution away from the source. The advantage of this behavior is that the voltage solution gives information about how far other points in M are from the source vertex; points with high voltages must be close, whereas points with low voltages must be far. Our idea is that the decay property can be used to construct an embedding using a selection of independent voltage functions generated by sources distributed across the manifold.

Our first step is to show theoretically that an embedding exists for the unit sphere S^{d-1}. With a basis in the discussion in [27] on spherical principal components analysis, we argue that there exists a large family of manifolds that can be approximated locally by sphere segments, which makes this result apply to a variety of cases beyond S^{d-1}. We illustrate the method's applicability on several manifolds, including a sphere and the MNIST data set.

6.1 Embedding of a unit sphere S^{d-1}

We show theoretically that we can embed the unit sphere, S^{d-1}, using ground voltage vectors from d voltage sources. We will use a disk kernel k with bandwidth $0 < r < 1$, and we will also take source radius $r_s = r$. We begin with the notion of an ϵ-injective embedding. We will also assume that our data distribution over S^{d-1} is the uniform distribution.

Definition 6.1. *Let $f : X \to Z$ be a map between metric spaces X, Z. For $\epsilon > 0$, we say that f is **s-injective** if*

$$d(x, x') > \epsilon \implies f(x) \neq f(x').$$

Our goal will be to find an ϵ-injective embedding of S^{d-1}. To do so, we consider the standard embedding of S^{d-1} in $\mathbb{R}^d$, and let $||x - x'||$ denote the ℓ_2 distance between points x, x'. Note that because the distance metric completely determines our voltage functions, it follows that any result for this setting will carry over to any isometric embedding of S^{d-1}.

Next, we characterize voltage functions over S^{d-1}. We let $\angle(x, x')$ denote the angle between them on the sphere. That is $\angle(x, x') = \arccos\langle x, x'\rangle$. We also let $r' = \angle(x, x') : ||x - x'|| = r$, denote the angle between two points with an ℓ_2 distance of r.

Theorem 6.2. *Let $x_0 \in S^{d-1}$ be an arbitrary point, and let v_0 be the grounded voltage function centered at x_0 with ground resistance ρ, source radius $r_s = r$, and kernel function k being the disk kernel with bandwidth r. Then there exists a function $f : [0, \pi] \to 1$ with the following properties.*

1. *f fully determines v_0. That is, $v_0(x) = f(\angle(x_0, x))$ for all $x \in S^{d-1}$).*
2. *f satisfies $f(\theta) > f(\theta')$, for all $\theta' > \theta + r$*

Proof. Property 1 immediately holds due to the rotational symmetry of the sphere about x_0. Furthermore, by the radial symmetry of the sphere, the same function f must suffice for all $x_0 \in S^{d-1}$.

To show property 2, we first show that f is weakly monotonic, that is $f(\theta) \geq f(\theta')$ if $\theta \leq \theta'$. To do so, let A_*, b_* be the operator and function from Definition B.1 such that $v_0 = A_* v_0 + b_*$. As we showed earlier, we have that

$$v_0 = \sum_{i=0}^{\infty} A_*^i b_0.$$

Thus, it suffices to show that all of the partial sums of this series are weakly monotonic. We do so through induction.

The base case is trivial as b_0 is clearly weakly monotonic with respect to the geodesic distance. For the inductive step, suppose that $v_0^n = \sum i = 0^n A_*^i b_0$ is weakly monotonic. Fix any $\theta \leq \theta'$ and let $x, x' \in S^{d-1}$ be points such that $\angle(x_0, x) = \theta, \angle(x_0, x') = \theta'$, and x_0, x, x' all lie on a great circle (that is, along a geodesic).

Let B denote the disk of radius r centered at x intersected with S^{d-1}, and B' denote the analogous disk for x'. The key observation is that B' is precisely the reflection of B across the perpendicular bisector of x, x' in S^{d-1}. Furthermore, this reflection is clearly an isometry and preserves the uniform measure μ over S^{d-1}. We let τ denote this reflection. Finally, observe that for all $y \in B \setminus (B \cap B')$,

$$\angle(x_0, y) \leq \angle(x_0, \tau(y)).$$

This holds even in the extreme case where x' is the antipodal point to x_0.

We now substitute τ into the equation $v_0^{n+1} = A_* v_0^n + b_*$. Doing so, we have

$$\begin{aligned} v_0^{n+1}(x') = A_* v_0^n(x') + b_*(x') &= \frac{\int_{B'} k(x', y) v_0^n(y) d\mu(y)}{\rho + \int_{B'} k(x', y) d\mu(y)} + b_*(x') \\ &= \frac{\int_{B} k(x', y) v_0^n(\tau(y)) d\mu(y)}{\rho + \int_{B'} k(x', y) d\mu(y)} + b_*(x') \\ &\leq \frac{\int_{B} k(x', y) v_0^n(y) d\mu(y)}{\rho + \int_{B'} k(x', y) d\mu(y)} + b_*(x') = v_0^{n+1}(x), \end{aligned}$$

with the inequalities holding from our observations about τ.

Finally, having shown that v_0 is weakly monotonic, we now turn to Property 2. Fix θ, θ' and let x, x' be such that $\angle(x_0, x) = \theta$, $\angle(x_0, x') = \theta'$. The key observation is that for all $y \in B(x', r)$, $\angle(x_0, y) > \theta$. This is from simple geometry as $\theta' > \theta + r'$. Thus, it consequently follows that $v_0(y) \leq v_0(x)$ for all such y. Substituting this, we have that

$$\begin{aligned} v_0(x') = A_* v_0(x') + b_*(x') &= \frac{\int_{B'} k(x', y) v_0(y) d\mu(y)}{\rho + \int_{B'} k(x', y) d\mu(y)} + b_*(x') \\ &\leq \frac{\int_{B'} k(x', y) v_0(x) d\mu(y)}{\rho + \int_{B'} k(x', y) d\mu(y)} + b_*(x') \\ &\leq v_0(x) \frac{\int_{B'} k(x', y) d\mu(y)}{\rho + \int_{B'} k(x', y) d\mu(y)} + b_*(x') \end{aligned}$$

with the last inequality holding since $b_*(x') = 0$ as x' cannot be inside the source region as $\angle(x_0, x') > r'$. □

We now show how to obtain an ϵ-injective embedding of S^{d-1}.

Theorem 6.3. *Let $e_1, \ldots, e_d$ denote the standard normal basis of $\mathbb{R}^{d-1}$, and associate them as voltage sources on S^{d-1}. Let $v_1, \ldots, v_d$ denote their respective voltage functions using a disk kernel of radius r (and a source radius of r). Then the map*

$$x \mapsto (v_1(x), \ldots, v_d(x))$$

is an $r'\sqrt{d}$-injective map, where $r' \in [0, \pi]$ denote the angle subtending a chord of length r on the unit circle.

Proof. Let x, x' be arbitrary points on S^{d-1} with $||x - x'|| > r\sqrt{d}$. Let θ_i denote the geodesic angle distance of x from e_i, and θ_i' be the same for x'. It follows that

$$\sum_{i=1}^{d} (\cos\theta_i - \cos\theta_i')^2 > r^2 d.$$

Thus, for some i, we must have $(\cos\theta_i - \cos\theta_i')^2 > r^2$. WLOG, this holds for $i = 1$. Applying this, we have

$$\begin{aligned} r'^2 < (\cos\theta_1 - \cos\theta_1')^2 &= (2 \sin\frac{\theta_1 + \theta_1'}{2} \sin\frac{\theta_1' - \theta_1}{2})^2 \\ &\leq 4 \sin^2 \frac{\theta_1' - \theta_1}{2} \leq (\theta_1' - \theta_1)^2. \end{aligned}$$

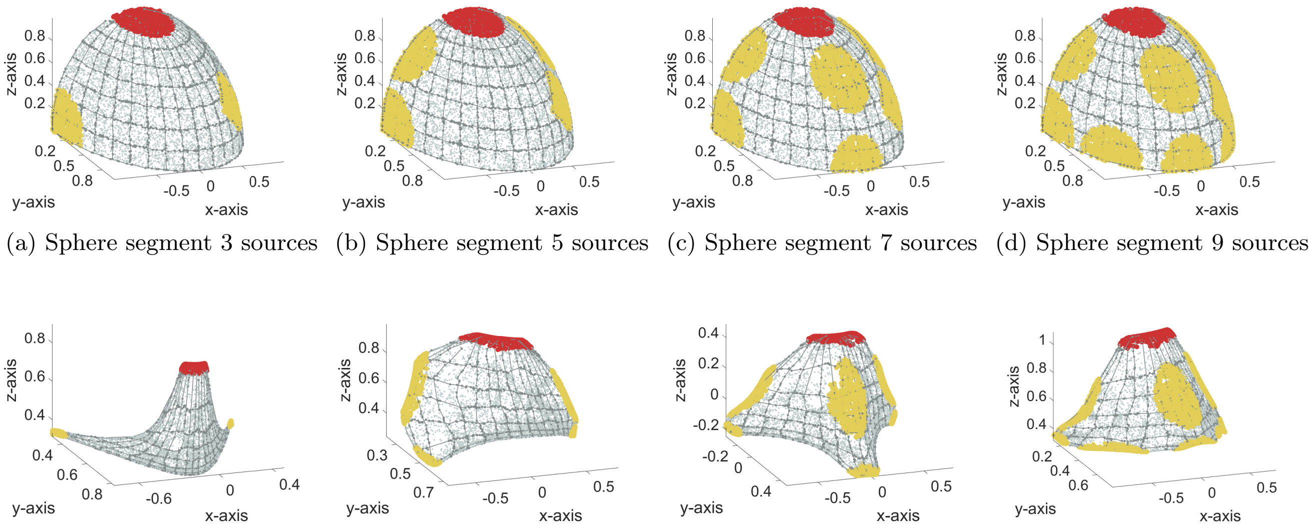
Thus $\theta_1' - \theta_1 > r'$ which implies $v_1(x') \neq v_1(x)$, as desired. □

6.2 Examples

To support our theoretical results, we show numerically how the grounded metric voltage function can be utilized to embed the unit sphere. Furthermore, we illustrate our method on two real-world data sets, namely the Frey faces data-set [28, Accessed: 2022-09-30] and MNIST [29].

In the experiments, we build an embedding by computing m independent voltage functions $v_n^{(i)}$ from m different sources $\widetilde{x}_i$ (landmarks), selected randomly from the manifold. From these voltages we then construct an m dimensional embedding $Z = (v_n^{(1)}, \ldots, v_n^{(m)}) \in \mathbb{R}^{n \times m}$. Since $m > 3$ for our experiments, the embeddings can not be visualized directly. Because of this, we create a projection of the embedding into $d \leq 3$ dimensions using $X_d = U_d \Lambda_d \in \mathbb{R}^{n \times d}$, which corresponds to a multi-dimensional scaling embedding (MDS) [30]. Here $Z_s = U \Lambda V^\top$ is a centering of Z and U_d, Λ_d are the d leading eigenvectors and eigenvalues.

Unit sphere experiment We consider the embedding of the two first quadrants of the unit sphere S^3. Using $n = 2^{13}$ points sampled i.i.d. from this sphere segment we calculate the voltage for $m = \{3, 5, 7, 9\}$ sources. In Fig. 6 we show the results from running these experiments. We see that increasing the number of sources gives an increasingly better embedding, which demonstrates the robustness of this algorithm w.r.t. choosing more sources than the intrinsic dimension.



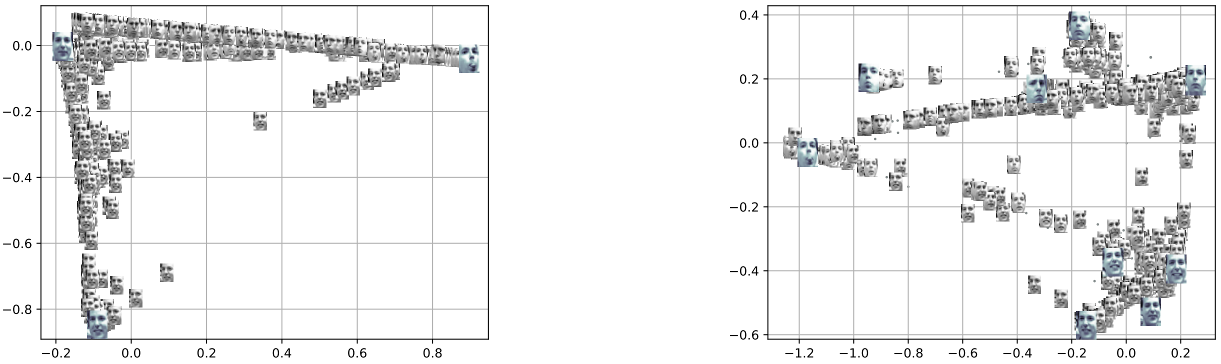
(a) Sphere segment 3 sources (b) Sphere segment 5 sources (c) Sphere segment 7 sources (d) Sphere segment 9 sources

(e) Embedding 3 sources (f) Embedding 5 sources (g) Embedding 7 sources (h) Embedding 9 sources

Figure 6: Illustration of the embedding of the two first quadrants of a sphere using localized voltage sources. The yellow and red colored regions correspond to the sources. The upper row is the true manifold and the lower row is the recreation using $m = \{3, 5, 7, 9\}$ sources. In order to align the embedding with the initial data we used orthogonal Procrustes analysis [30].

Frey-face experiment The Frey face data set is taken from [28, Accessed: 2022-09-30] and consists of 1965 images of size 20×28 of Brendan Frey's face. We generate two embeddings using respectively $m = \{3, 10\}$ landmark sources selected randomly from the images. Fig. 7 illustrates a 2-dimensional projection of these embeddings. From the figure we see that the images are clustered based on facial expressions. In particular, the images tend towards landmarks with a similar expression and change gradually as you move away. Comparing Fig. 7a and Fig. 7b, we see that additional landmarks open up local clusters by spreading the images out in between.

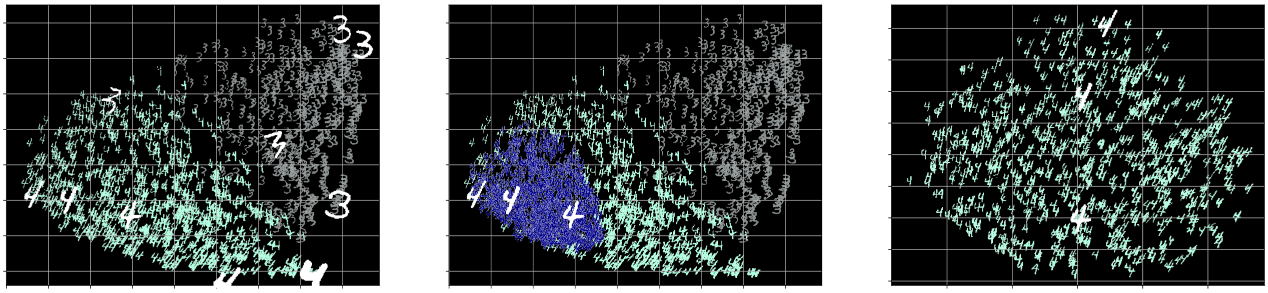
MNIST experiment The MNIST data-set we use is taken from [28, Accessed: 2022-09-30] and consists of 60000 28×28 images of handwritten digits. For our experiment, we extract 5000 images from each of the digits $\{3, 4\}$. From each digit, we also select at random $m = 5$ landmarks. Fig. 8 illustrates the embedding. In particular, Fig. 8a shows how the embedding separates digit 3 and 4. Furthermore, we can see how the orientation/rotation of the digits determines their clustering, especially apparent for digit 4. Selecting a subset of the landmarks, see Fig. 8b, we make a local embedding of their neighborhood as shown in Fig. 8c.



(a) Embedding with 3 landmarks

(b) Embedding with 10 landmarks

Figure 7: Embedding of the Frey-face dataset. Larger images corresponds to landmarks.



(a) MNIST embedding with 10 landmarks

(b) Select a subset of landmarks and their neighborhood

(c) Local embedding of neighborhood of selected landmarks

Figure 8: Embedding of digit 3 and 4 from the MNIST dataset suing 10 landmarks and 5000 samples from each digit.

7 A note on computational efficiency

We have seen how the grounded metric voltage function decays exponentially with the distance from the source. This property can be thought of as a localization property, as the voltage will be negligible on most of the graph, except for finite support around the source.

The advantage of this localization is that when computing the voltage, only a subset of the graph needs to be used for any given source, reducing both memory and computational requirements. Importantly, the finite support of the voltage solution is a great alternative to traditional methods such as Laplacian eigenmaps, which rely on the computation of eigenfunctions that are typically global (meaning computations relies on the entire graph).

In the following, we characterize the effective support of the voltage solution and its dependency on the parameters of the grounded metric graph. Let M_1 be a source region of radius r as defined in Def. 3.1. Furthermore, let $f : M \to [0, 1]$ be a radially symmetric function over the metric space (M, d) such that $f(x) = h(d(x_1, x))$. For $\tau > 0$ and $x_1 \in M_1$ we say that f is localized around x_1 with support radius r_{supp} if for $d(x_1, x) > r$,

$$h(x) \geq \tau \quad \text{for} \quad d(x_1, x) \leq r_{supp} \quad \text{and} \quad h(x) < \tau \quad \text{for} \quad d(x_1, x) > r_{supp}.$$

Corollary 7.1. *(Locality of the voltage solution) Consider a disk in $\mathbb{R}^d$. As a consequence of the exponential decay of the grounded metric voltage described in Theorem 5.5 and Corollary 5.6, the voltage* *solution will be localized around the source region M_1, with a support radius bounded by $r_l \leq r_{supp} \leq r_u$ where*

$$r_l := \frac{\frac{r}{2}\log 1/\tau}{\log\left(Cr^d + \rho\right)/\Gamma} \quad \textit{and} \quad r_u := \frac{r \log 1/\tau}{\log\left(1 + \rho/Cr^d\right)}.$$

From Corollary 7.1 it follows that the support of the voltage solution is restricted to a subset of the manifold, centered around the source. In particular, we see how the resistance to ground $1/\rho$ plays a crucial role in its relation to the effective support. Namely, as the resistance to ground decrease to zero, i.e. $\rho \to \infty$, then the effective support will also go to zero $r_{supp} \to 0$. The intuition here is that, with zero resistance to the ground, the ground drains all the current. Similarly, when the resistance to the ground goes to infinity, i.e. $\rho \to 0$ then the effective support goes to infinity $r_{supp} \to \infty$. This demonstrates the importance of the grounding node in achieving localized effective support for the voltage solution.

8 Conclusion and future work

In this paper, we have demonstrated how the grounded metric voltage function holds promise as a tool for low-cost and distributed embedding of manifolds. The goal of this paper has been to establish a theoretical basis for further development of this embedding strategy. In particular, we have shown existence, convergence, and shape properties and demonstrated how the voltage captures local structure of the manifold.

There are three main directions for expanding on this work. First, we are interested in exploring the potential of constructing low-dimensional embeddings of data sets using grounded voltage functions. While we took the first steps in this direction in this paper (with theoretical results on the sphere and experiments over the sphere, MNIST, and Frey-faces), we are actively working towards more general results over arbitrary manifolds and more extensive real-world data sets.

Second, we are interested in the computational aspect of this technique – the local nature of the grounded voltage function suggests that distributed computation approaches are viable. Thus, one important problem is to develop algorithms that can provably and empirically leverage this. Finally, we are interested in describing the limiting operator of our approach.

References

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

[2] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In *Proceedings of the National Academy of Sciences*, volume 102, pages 7426–7431, 2005.

[3] Peter G Doyle and J Laurie Snell. *Random walks and electric networks*, volume 22. American Mathematical Soc., 1984.

[4] Ulrike Luxburg, Agnes Radl, and Matthias Hein. Getting lost in space: Large sample analysis of the resistance distance. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[5] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction: a comparative review. *Journal of machine learning research*, 10(66-71):13, 2009.

[6] Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel PCA and de-noising in feature spaces. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1998.

[7] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[8] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[9] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11):2579–2605, 2008.

[10] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[11] Stefan Steinerberger and Yulan Zhang. t-SNE, forceful colorings, and mean field limits. *Research in the Mathematical Sciences*, 9(3):42, 2022.

[12] Dhruv Kohli, Alexander Cloninger, and Gal Mishne. LDLE: Low distortion local eigenmaps. *Journal of machine learning research*, 22:282–1, 2021.

[13] Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2008.

[14] Jihun Ham, Daniel D Lee, Sebastian Mika, and Bernhard Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the 21st international conference on Machine learning*, page 47, 2004.

[15] Dmitry Kobak and George C Linderman. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature biotechnology*, 39(2):156–157, 2021.

[16] Douglas J Klein and Milan Randić. Resistance distance. *Journal of mathematical chemistry*, 12(1):81–95, 1993.

[17] Stefan Forcey and Drew Scalzo. Phylogenetic networks as circuits with resistance distance. *Frontiers in Genetics*, 11:1177, 2020.

[18] Teng Zhang and Changjiang Bu. Detecting community structure in complex networks via resistance distance. *Physica A: Statistical Mechanics and its Applications*, 526:120782, 2019.

[19] Prabir Barooah and Joao P. Hespanha. Graph effective resistance and distributed control: Spectral properties and applications. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3479–3485, 2006.

[20] Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[21] Sotharith Tauch, William Liu, and Russel Pears. Measuring cascade effects in interdependent networks by using effective graph resistance. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 683–688, 2015.

[22] Yakup Koç, Martijn Warnier, Piet Van Mieghem, Robert E. Kooij, and Frances M.T. Brazier. The impact of the topology on cascading failures in a power grid model. *Physica A: Statistical Mechanics and its Applications*, 402:169–179, 2014.

[23] Xiangrong Wang, Yakup Koç, Robert E. Kooij, and Piet Van Mieghem. A network approach for power grid robustness against cascading failures. In *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 208–214, 2015.

[24] Guido Cavraro and Vassilis Kekatos. Graph algorithms for topology identification using power grid probing. *IEEE Control Systems Letters*, 2(4):689–694, 2018.

[25] Arpita Ghosh, Stephen Boyd, and Amin Saberi. Minimizing effective resistance of a graph. *SIAM Review*, 50(1):37–66, 2008.

[26] Iqra Altaf Gillani and Amitabha Bagchi. A queueing network-based distributed laplacian solver for directed graphs. *Information Processing Letters*, 166:106040, 2021.

[27] Didong Li, Minerva Mukhopadhyay, and David B Dunson. Efficient manifold and subspace approximations with spherelets. *arXiv preprint arXiv:1706.08263*, 2017.

[28] Internet. Sam Roweis. `https://cs.nyu.edu/~roweis/data.html`, 2022. Accessed: 2022-09-30.

[29] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[30] Ivan Dokmanic, Reza Parhizkar, Juri Ranieri, and Martin Vetterli. Euclidean distance matrices: Essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015.

A Proof of Lemma 2.4

To prove Lemma 2.4 we first include some notation. Let $s \in X^s$ be the source node, g the ground node, and $e_i \in \mathbb{R}^{n+1}$ the indicator of node i.

Proof. The constraints $v(s) = 1$ and $v(g) = 0$ can be written as $v^\top e_s = 1$ and $v^\top e_g = 0$ respectively. We apply these constraints using the Lagrangian multipliers λ_s and λ_g, which gives $f(v) = v^\top Lv - \lambda_s v^\top e_s - \lambda_g v^\top e_g$. When equating the derivative of f to zero, and using $Lv = (D - W)v = D(v - D^{-1}Wv)$ we can write this equation as

$$v = D^{-1}Wv + D^{-1}\lambda_s e_s - D^{-1}\lambda_g e_g. \tag{A.1}$$

Since we enforce $v(s) = 1$ on the source node, it follows that row s in Eq. (A.1) is $\lambda_s = d_{ss} - (Wv)_s$. Similarly, we have for row g that $\lambda_g = (Wv)_g$. It follows, $v = \widetilde{D}^{-1}\widetilde{W}^{(s)}v$, where

$$\widetilde{W}^{(s)}_{ij} = \begin{cases} 1, & \text{if} \quad i = j, x_i \in X^s \\ 0, & \text{if} \quad i \neq j, x_i \in X^s \\ W_{ij}, & \text{otherwise,} \end{cases}$$

and $\widetilde{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $D_{ii} = 1$ for $x_i \in X^s$ and $\widetilde{D}_{ii} = \rho + \sum_{j=1}^{n} w_{ij}$ otherwise. Note that the $n+1$'th row and column (row and column of the ground) are dropped, since the voltage on the ground is always zero anyway. If we have more source nodes in X^s, these can be incorporated similarly by applying Lagrangian multipliers $\lambda_i v_i e_i$ for each $i \in X^s$. □

B Proofs from Section 5

Throughout this section, we treat M^s as fixed so as to avoid cumbersome notation addressing the source region. Furthermore, because v_n^* and v^* are defined to equal 1 over M^s, we will reinterpret them as maps $M \setminus M^s \to [0, 1]$. Any such map v can be immediately transformed into a map over all M by including $v(x) = 1$ for all $x \in M^s$.

B.1 Proof of Theorem 5.2

We begin by expressing Proposition 5.1 with an affine transformation.

Definition B.1. *Let μ be a measure over M, and let $v : M \setminus M^s \to \mathbb{R}$ be a measurable map. Then define $A_\mu v : M \setminus M^s \to \mathbb{R}$ and $b_\mu : M \setminus M^s \to \mathbb{R}$ as*

$$(A_\mu v)(x) = \frac{\int_{M\setminus M^s} k(x, y)v(y)d\mu(y)}{\rho + \int_M k(x, y)d\mu(y)} \quad \textit{and} \quad b_\mu(x) = \frac{\int_{M^s} k(x, y)d\mu(y)}{\rho + \int_M k(x, y)d\mu(y)}.$$

Together, we let $T_\mu v = A_\mu v + b_\mu$.

Thus, to prove Theorem 5.2, it suffices to show that for any choice of μ, there exists a unique map $v^* : M \setminus M^s \to \mathbb{R}$ that satisfies $T_\mu v^* = A_\mu v^* + b_\mu$. To this end, the key idea will be to leverage that A_μ is a contraction.

Definition B.2. *Let $\mathcal{F}$ denote the space of measurable functions $f : M \to \mathbb{R}$. Define $||f||_\infty$ as*

$$||f||_\infty = \sup_{x\in X} |f(x)|.$$

Furthermore, the ℓ_∞ distance between two functions f, g is defined as $||f - g||_\infty$.

It is well known that $\mathcal{F}$ is a closed metric space under the ℓ_∞ metric. We now show that A_μ is a contraction with respect to the ℓ_∞-metric.

Lemma B.3. *For any measurable function $f : M \to \mathbb{R}$,*

$$||A_\mu f||_\infty \leq \frac{1}{1+\rho}||f||_\infty$$

Proof. This follows from algebraic manipulations. We have

$$\begin{aligned}\sup_{x\in M}|(A_\mu f)(x)| &= \sup_{x\in M\setminus M^s}\frac{\int_{M\setminus M^s}k(x,y)f(y)d\mu(y)}{\rho+\int_M k(x,y)d\mu(y)} \leq \sup_{x\in M}\frac{\int_{M\setminus M^s}k(x,y)||f||_\infty d\mu(y)}{\rho+\int_M k(x,y)d\mu(y)}\\ &= ||f||_\infty \sup_{x\in M}\left(\frac{\int_{M\setminus M^s}k(x,y)d\mu(y)}{\rho+\int_M k(x,y)d\mu(y)}\right) \leq ||f||_\infty\frac{1}{1+\rho},\end{aligned}$$

with the last inequality holding since k has range $[0,1]$ by assumption. □

We now prove Theorem 5.2.

Proof. Set $u_0 : M \setminus M^s \to \mathbb{R}$ as the 0 function. That is $u_0(x) = 0$ for all x. For $i \geq 1$, define $u_i = A_\mu(u_{i-1}) + b_\mu$. We claim that this defines a Cauchy sequence over $\mathcal{F}$ (Definition B.2). To see this, observe that for any $i \geq 1$,

$$||u_{i+1} - u_i||_\infty = ||(A_\mu u_i + b_\mu) - (A_\mu u_{i-1} + b_\mu)||_\infty = ||A_\mu(u_i - u_{i-1})||_\infty \leq C||u_i - u_{i+1}||_\infty$$

with the last inequality holding by Lemma B.3. This implies that the distances between consecutive elements of our sequence decrease geometrically. Since $\mathcal{F}$ is closed with respect to ℓ_∞, it follows that this sequence converges to some function v.

Next, we show v satisfies $v = A_\mu v + b_\mu$. Fix any $\epsilon > 0$. Then there exists n such that $||v - u_n||_\infty, ||v - u_{n+1}|| < \epsilon$. It follows that

$$\begin{aligned}||v - (A_\mu v + b_\mu)||_\infty &\leq ||v - u_{n+1}||_\infty + ||(A_\mu v + b_\mu) - u_{n+1}||_\infty\\ &\leq \epsilon + ||(A_\mu v + b_\mu) - (A_\mu u_n + b_\mu)||_\infty = \epsilon + ||A_\mu(v - u_n)||_\infty \leq \epsilon + \frac{\epsilon}{1+\rho}.\end{aligned}$$

Since ϵ was arbitrary, it follows that $v = A_\mu v + b_\mu$.

Next, to show v has range $[0,1]$, we simply observe that u_n has range $[0,1]$ for all n. This can be shown by induction on n. The base case clearly holds, and for the inductive step, observe that $(A_\mu v_n)(x), b_\mu(x) \geq 0$ k is always non-negative which implies $v_{n+1}(x) = (A_\mu v_n)(x) + b_\mu(x) \geq 0$. To show that it is at most 1, we have

$$v_{n+1}(x) = (A_\mu v_n)(x) + b_\mu(x) = \frac{\int_M k(x,y)v_n(y)d\mu(y)}{\rho+\int_M k(x,y)d\mu(y)} \leq \frac{\int_M k(x,y)d\mu(y)}{\rho+\int_M k(x,y)d\mu(y)} \leq 1.$$

Finally, to show uniqueness, suppose that v' also satisfies $v' = A_\mu v' + b_\mu$. Then we have

$$||v - v'||_\infty = ||(A_\mu v + b_\mu) - (A_\mu v' + b_\mu)||_\infty = ||A_\mu(v - v')||_\infty \leq \frac{1}{1+\rho}||v - v'||_\infty.$$

which implies $||v - v'||_\infty = 0$ as desired. □

B.2 Proof of Theorem 5.4

We begin by observing that the extended voltage solution (Definition 5.3 for a finite sample S exhibits similar behavior to the voltage solution over a measure μ. The key idea is to define the measure μ_S over M as the measure induced by the uniform distribution over S.

Definition B.4. *Let $S = \{x_1, x_2, \ldots, x_n\}$ be a finite set of n points from M. Then μ_S is the measure on M defined as*

$$\mu_S(P) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(x_i \in P),$$

for all measurable sets P.

By substituting μ_S into Definitions B.4 and B.1, we have

$$v_n^* = A_{\mu_S} v_n^* + b_{\mu_S}.$$

Here note that we are considering its restriction to $M - M^s$ as discussed in the beginning of this section. Furthermore, the existence and uniqueness of v follow directly Theorem 5.2.

We now desire to show that as the sample size increases to infinity, v_n^* pointwise converges towards v^*. To prove this, we begin by showing that for large values of n, T_{μ_S} serves as an approximation for T_μ when applied to v^*.

Lemma B.5. *Let $x \in M \setminus M^s$ be a point, and $\epsilon > 0$ be a real number. Then*

$$\Pr_{S\sim\mu^n}\left[\left|(T_\mu v^*)(x) - (T_{\mu_S} v^*)(x)\right| > \epsilon\right] < 4\exp(-\frac{n\rho^2\epsilon^2}{9}).$$

Proof. Let $y \sim \mu$, and $Y = k(x,y)v^*(y)\mathbb{1}(y \notin M^s) + k(x,y)\mathbb{1}(y \in M^s)$. Let $Y_1, Y_2, \ldots, Y_n$ are i.i.d copies of Y. Since k, v^* both have ranges in $[0,1]$, it follows that Y has range $[0,1]$ as well. It follows by Hoeffding's inequality that

$$\Pr\left[\left|\mathbb{E}[Y] - \frac{1}{n}\sum_{i=1}^{n} Y_i\right| > \rho\epsilon\right] < 2\exp\left(-n\rho^2\epsilon^2\right).$$

Similarly, we let Z denote the random variable $k(x,y)$, $y \sim \mu$ and $Z_1, \ldots, Z_n$ be i.i.d copies of Z. We also have

$$\Pr\left[\left|\mathbb{E}[Z] - \frac{1}{n}\sum_{i=1}^{n} Z_i\right| > \rho\epsilon\right] < 2\exp\left(-n\rho^2\epsilon^2\right).$$

Next, we express $(T_\mu v^*)(x)$ and $(T_{\mu_S} v^*)(x)$ in terms of these variables. We have,

$$(T_\mu v^*)(x) = \frac{\int_{M^s} k(x,y)d\mu(y) + \int_{M\setminus M^s} k(x,y)v^*(y)d\mu(y)}{\rho + \int_M k(x,y)d\mu(y)} = \frac{\mathbb{E}[Y]}{\rho + \mathbb{E}[Z]}.$$

Similarly,

$$(T_{\mu_S} v^*)(x) = \frac{\frac{1}{n}\sum Y_i}{\rho + \frac{1}{n}\sum Z_i}.$$

We will now use these expressions to bound the difference between $(A_\mu u + b_\mu)(x)$ and $(A_{\mu_S}(u) + b_{\mu_S})(x)$ in terms of Z, Y, Z_i, Y_i. For convenience, let

$$\Delta = \max\left(|\frac{1}{n}\sum Y_i - \mathbb{E}[Y]|, |\frac{1}{n}\sum Z_i - \mathbb{E}[Z]|\right).$$

Then we have

$$|(A_\mu u)(x) - (A_{\mu_S} u)(x)| = \left|\frac{\mathbb{E}[Y]}{\rho + \mathbb{E}[Z]} - \frac{\frac{1}{n}\sum Y_i}{\rho + \frac{1}{n}\sum Z_i}\right|.$$

To bound this, we split the difference into two parts. We have

$$\left|\frac{\mathbb{E}[Y]}{\rho+\mathbb{E}[Z]} - \frac{\frac{1}{n}\sum Y_i}{\rho+\mathbb{E}[Z]}\right| \leq \frac{\Delta}{\rho},$$

$$\left|\frac{\frac{1}{n}\sum Y_i}{\rho+\mathbb{E}[Z]} - \frac{\frac{1}{n}\sum Y_i}{\rho+\frac{1}{n}\sum Z_i}\right| \leq \frac{(\frac{1}{n}\sum Y_i)\Delta}{(\rho+\mathbb{E}[Z])(\rho+\frac{1}{n}\sum Z_i)} \leq \frac{\Delta}{\rho}.$$

Similarly, we can also show that $|(b_\mu - b_{\mu_S})(x)| \leq \frac{\Delta}{\rho}$. Applying a union bound for the deviations of Y and Z, we see that $\Delta > \frac{\rho}{3}$ with probability at most $4\exp(-\frac{n\rho^2\epsilon^2}{9})$. □

Next, we show how to adapt Lemma B.5 to hold uniformly over the entire sample S. To do so, we first define a type of metric over the space of functions on M.

Definition B.6. *Let $S = \{x_1, \ldots, x_n\} \subset M$ be a set of points, and let $u : M \setminus M^s \to \mathbb{R}$ be a function. Then $||u||_S = \max_{x_i \in M\setminus M^s} |u(x_i)|$ is the largest absolute value of u over S.*

Lemma B.7. *Let $S = \{x_1, \ldots, x_n\} \sim \mu^n$, and $\epsilon > 0$ be a real number. Then for $n > \frac{6}{\epsilon\rho}$,*

$$\Pr_{S\sim\mu^n}[||v^* - T_{\mu_S}v^*||_S > \epsilon] < 4n\exp\left(-\frac{(n-1)\rho^2\epsilon^2}{36}\right).$$

Proof. Fix $x_i \in M \setminus M^s$. It suffices to show that $|(T_\mu v^*)(x_i) - (T_{\mu_S}v^*)(x_i)| > \epsilon$ with probability at most $4\exp\left(-\frac{(n-1)\rho^2\epsilon^2}{36}\right)$. To do so, we essentially apply Lemma B.5. The only difficulty is that S and x_i are no longer independent as $x_i \in S$. To resolve this, we observe that x_i is independent from $S \setminus x_i$, and use this to bound the difference. Applying this, we have,

$$|(T_\mu v^*)(x_i) - (T_{\mu_{S\setminus x_i}}v^*)(x_i)| \leq \left|(T_\mu v^*)(x_i) - (T_{\mu_{S\setminus x_i}}v^*)(x_i)\right| + \left|(T_{\mu_{S\setminus x_i}}v^*)(x_i) - (T_{\mu_S}v^*)(x_i)\right|.$$

The former term can be directly bounded using Lemma B.5. Because x_i and $S \setminus x_i$ are independent, we have that with probability at most $4\exp\left(-\frac{(n-1)\rho^2\epsilon^2}{36}\right)$,

$$\left|(T_\mu v^*)(x_i) - (T_{\mu_{S\setminus x_i}}v^*)(x_i)\right| > \frac{\epsilon}{2}.$$

It thus suffices to show that the latter term is at most $\frac{\epsilon}{2}$. To do so, we split Tv^* into $Au + b$ for both μ_S and $\mu_{S\setminus x_i}$. Using the same variables, Y, Y_j, Z, Z_j as in the proof of Lemma B.5 (and substituting $x = x_i$), we have

$$\begin{aligned}\left|A_{\mu_{S\setminus x_i}}v^*(x_i) - A_{\mu_S}v^*(x_i)\right| &= \left|\frac{\frac{1}{n-1}\sum_{j\neq i}Y_j}{\rho+\frac{1}{n-1}\sum_{j\neq i}Z_j} - \frac{\frac{1}{n}\sum Y_j}{\rho+\frac{1}{n}\sum Z_j}\right| \\ &\leq \left|\frac{\frac{1}{n}+\frac{1}{n}\sum Y_j}{\rho-\frac{1}{n}+\frac{1}{n}\sum Z_j} - \frac{\frac{1}{n}\sum Y_j}{\rho+\frac{1}{n}\sum Z_j}\right| \leq \frac{2}{n\rho},\end{aligned}$$

with the last inequality coming from the same manipulations applied in the proof of Lemma B.5. We can similarly show that

$$\left|b_{\mu_{S\setminus x_i}}(x_i) - b_{\mu_S}(x_i)\right| \leq \frac{1}{n\rho}.$$

However, since $n > \frac{6}{\epsilon\rho}$, it follows that $|T_{\mu_{S\setminus x_i}}(u)(x_i) - T_{\mu_S}(u)(x_i)| \leq \frac{\epsilon}{2}$, as desired. □

We are now prepared to prove Theorem 5.4.

Proof. Let $S \sim \mu^n$ be a sample of n i.i.d points, and let v_n^* be its corresponding voltage solution. By using natural analogs to Theorem 5.2 and Lemma B.3, we immediately have that for any map $u : M \to [0, 1]$, $||A_{\mu_S} u||_S \leq \frac{1}{1+\rho} ||u||_S$, and the sequence $\{T_{\mu_S}^m u\}_{m \geq 1}$ converges pointwise to v_n^*. We also have that the norm induced by Definition B.6 induces a metric over the space of maps $M \setminus M^s \to \mathbb{R}$.

Next, by Lemmas B.5 and B.7, with probability $1 - 2\exp(O(-n))$ over S, both of their desired bounds hold for some given point $x \in M$. Using this, along with the fact that v^*, v_n^* are the fixed points of T_μ, T_{μ_S}, we have,

$$\begin{aligned}
|v^*(x) - v_n^*(x)| = |T_\mu v^*(x) - T_{\mu_S} v_n^*(x)| &\leq |T_\mu v^*(x) - T_{\mu_S} v^*(x)| + |T_{\mu_S} v^*(x) - T_{\mu_S} v_n^*(x)| \\
&\leq \epsilon + |A_{\mu_S} v^*(x) - A_{\mu_S} v_n^*(x)| \leq \epsilon + \frac{\sum_{x_i \in M \setminus M^s} k(x, x_i) |v^*(x_i) - v_n^*(x_i)|}{\rho + \sum_{x_i} k(x, x_i)} \\
&\leq \epsilon + |v^* - v_n^*|_S \leq \epsilon + |v^* - T_{\mu_S} v^*|_S + \sum_{i=1}^{\infty} |T_{\mu_S}^i v^* - T_{\mu_S}^{i+1} v^*|_S \leq \epsilon + \frac{\epsilon}{\rho},
\end{aligned}$$

Since ϵ is arbitrary, the claim follows as ρ is a fixed constant. □

C Proofs on the shape and support of the GMV

C.1 Proof of Theorem 5.5

In this section, we prove Theorem 5.5. First, we establish existence, radial symmetry, and monotonicity regarding λ. Let A_ν and b_ν be as defined in Definition B.1, with $M = S^{d-1}$ and $M^s = B(0, r_s)$. The existence of $\lambda(x) = v^*(x)$, where $v^*(x) = (A_\mu v^*)(x) + b_\mu$, then follows straightforwardly from Theorem 5.2. Meanwhile, from Section B, we have already established that $v^*(x)$ corresponds to a local average of its neighbors, which implies that λ must be strictly non-increasing away from the source. Furthermore, due to the radial symmetry of the sphere and the kernel k, it follows that also A_μ and $b_\mu(x)$ are radially symmetric. Moreover, in Section B we establish that $v_{n+1}(x) = (A_\mu v_n)(x) + b_\mu$ converges to v^*. With v_0 radially symmetric, it follows that also λ must be radially symmetric. Finally, we examine the upper and lower bounds.

Proof. (Theorem 5.5) We have already shown that λ exists and is radially symmetric (meaning h exists) and that h is strictly non-increasing. We now prove the bounds on h.

Upper bound: Let x be a point on the sphere, and $z = d_m(x_s, x)$ be the geodesic distance from the source landmark x_s. Furthermore, since we consider the unit sphere, we have for two points on the sphere ,whose euclidean distance is r, that $d_M(x_i, x_j) = \arccos(\langle x_i, x_j \rangle) = \phi(r)$.

The key idea is now to bound the integral, $\int_M k(x, y)v(y)d\mu(y)$. To do so, observe that by the definition of k, this integral is only non-zero over the ball $B(x, r)$. Furthermore, at most half of the probability mass of this ball satisfies $v(y) \geq v(x)$, and all points inside this ball satisfy $v(y) \leq h(z - \phi(r))$ as $z - \phi(r)$ is the closest that any point in this ball gets to the origin. Thus, bounding the expectation, we have

$$\int_M k(x, y)v(y)d\mu(y) \leq \frac{a}{2} h(z - r) + \frac{a}{2} h(z).$$

Substituting this, we have that

$$h(z) = v(x) = T_\mu v(x) = \frac{\int_M k(x, y)v(y)d\mu(y)}{\rho + a} \leq \frac{\frac{a}{2} h(z - \phi(r)) + \frac{a}{2} h(z)}{\rho + a}.$$

With $z = z' + \phi(r)$ the upper bound can be written as $h(z + \phi(r)) \leq C_u h(z)$ where $C_u = 1/(1 + 2\rho/a)$. Recursion of this expression gives $h(z + t\phi(r)) \leq C_u^t h(z)$. We now let $z = z_1$ where z_1 defines the center of the source, since $h(z_1) = 1$ this gives $h(z + t\phi(r)) \leq \exp(-t \ln(1 + 2\rho/a))$.

Lower Bound: We use a similar strategy as we did with the upper bound. This time, we let Γ the probability mass of the ball $B(x,r)$ that consists of points y for which $||y|| \leq z - \phi(r/2)$. While this value depends on z, it can be lower bounded by the case in which $||z|| = \phi(r)$. This constant thus equals the intersection volume between 2 d-dimensional spheres. Using this constant, we see that

$$\int_M k(x,y)v(y)d\mu(y) \geq \Gamma h(z - \phi(r/2)).$$

Substituting this, we have

$$h(z) = v(x) = T_\mu v(x) = \frac{\int_M k(x,y)v(y)d\mu(y)}{\rho + a} \geq \frac{\Gamma h(z - \phi(r/2))}{\rho + a},$$

We now treat the lower bound similarly to the upper bound. Here $h(z + t\phi(r/2)) \geq C_l^t$ where $C_l = \Gamma/(a+\rho)$, which gives $h(z + t\phi(r/2)) \geq \exp\left(-t \ln\left((a+\rho)/\Gamma\right)\right.$. □

C.2 Proof of Corollary 7.1

Consider a disk in $\mathbb{R}^d$. Let $z = z_1$ where z_1 defines the center of the source, $h(z_1) = 1$. We are interested in the radial distance r_{supp} from z_1 such that $h(z_1 + r_{supp}) \geq \tau$. Let $r_{supp} = tr$. From Corollary 5.6 we then have that

$$h_u(r_{supp}) := \exp\left(-\frac{r_{supp}}{r}\ln\left(1 + 2\rho/a\right)\right) \quad \text{and} \quad h_l(r_{supp}) := \exp\left(-2\frac{r_{supp}}{r}\ln\left((a+\rho)/\Gamma\right)\right)$$

The upper bound for r_{supp} should therefore be $r_{s,u} = \max\{r_{supp} : h_u(z_1 + r_{supp}) \geq \tau\}$. Similarly the lower bound should be $r_{s,l} = \max\{r_{supp} : h_l(z_1 + r_{supp}) \geq \tau\}$. Solving for r_{supp} we find that $h_l(r_{supp}) \geq \tau$ and $h_u(r_{supp}) \geq \tau$ implies

$$r_{supp} \leq \frac{\frac{r}{2}\log 1/\tau}{\log\left(Cr^d + \rho\right)/\Gamma} \quad \text{and} \quad r_{supp} \leq \frac{r \log 1/\tau}{\log\left(1 + \rho/Cr^d\right)}.$$

respectively. Here we used that $a = Cr^d$ as it is the volume of a d dimensional sphere. From the definition of $r_{s,l}$ and $r_{s,u}$ the result follows.