

Assessing quantum advantage for Gaussian process regression

Received: 4 August 2025

Accepted: 31 July 2026

Published online: 14 August 2026

Cite this article as: Lowe, D., Kim, M.S., Bondesan, R. Assessing quantum advantage for Gaussian process regression. *npj Quantum Inf*(2026). <https://doi.org/10.1038/s41534-026-01350-8>

Dominic Lowe, M. S. Kim & Roberto Bondesan

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

Assessing Quantum Advantage for Gaussian Process Regression

Dominic Lowe^{1,*}, M.S. Kim^{1,*}, Roberto Bondesan^{2,*}

¹ Blackett Laboratory, Imperial College London, SW7 2AZ, United Kingdom

² Department of Computing, Imperial College London, SW7 2AZ, United Kingdom

Abstract

Gaussian Process Regression is a machine learning technique with established applications for which several quantum algorithms have been proposed. We show here that in a wide range of scenarios these algorithms show no exponential speedup. We achieve this by rigorously proving that the condition number of a kernel matrix scales at least linearly with the matrix size under general assumptions on the data and kernel. We additionally prove that the sparsity and Frobenius norm of a kernel matrix scale linearly under similar assumptions. Our results give similar conclusions for kernel ridge regression and quantum support vector machines under the same assumptions. The implications for the quantum algorithms runtime are independent of the complexity of loading classical data on a quantum computer and also apply to dequantised algorithms. We supplement our theoretical analysis with numerical verification for popular kernels in machine learning.

1 Introduction

Over the last decade, the question of whether quantum computers can speed up machine learning workloads has been under intense investigation [1, 2, 3]. With quantum computers excelling at performing matrix operations in high-dimensional spaces, a natural question is whether machine learning algorithms can be implemented using quantum linear algebra more efficiently [4, 5, 6]. The workhorse of these proposals is the HHL algorithm that approximates the solution of a linear system [7]. An important caveat of these quantum machine learning algorithms is that for exponential speedup over the best classical methods, we need to load the classical data into a quantum superposition in logarithmic time, an assumption whose practicality has been heavily debated [8]. In fact, classical algorithms supplied with a similar input data structure exhibit only a polynomial slowdown compared to their quantum analogues [9, 10]. In this paper, we consider the problem of benchmarking

* Correspondence to: dal20@ic.ac.uk, m.kim@imperial.ac.uk, r.bondesan@imperial.ac.uk

quantum machine learning algorithms assuming the data has been loaded on a quantum computer. We focus primarily on the problem of Gaussian Process Regression (GPR). GPR is a machine learning technique that is valued in several domains due to its ability to provide calibrated uncertainty over predictions [11], but that suffers from a computational bottleneck due to the requirement of inverting a covariance matrix. We show that for a broad class of kernels, the condition number of the covariance matrix – defined as the ratio of its largest to smallest eigenvalue – scales at least linearly with the number of datapoints. All currently proposed quantum algorithms for Gaussian Process Regression (GPR) have a time complexity that depends, either linearly or polynomially, on this condition number [12, 13, 14].

The significance of this result is that it allows a direct comparison of these quantum algorithms to their classical counterparts. From this it becomes clear that these algorithms (even before data-loading or error correction are taken into account) can only perform marginally better than a standard classical approach.

Several alternative classical approaches to matrix inversion also have a time complexity that depends at least linearly on the condition number, and so are affected similarly. For example, conjugate gradient methods for sparse matrices [15] and quantum inspired approaches for matrix inversion [16]. The algorithms mentioned in [16] instead depend on the condition number with respect to the Frobenius norm, but this still scales at least linearly with the number of datapoints by corollary 2.1.

Our main technical contribution is an asymptotic formula for the condition number in the regularised case, and a lower bound in the un-regularised case. We believe the former is the first result of its kind, in that it describes the condition number (for a wide variety of kernels) purely in terms of the regularisation parameter and the largest eigenvalue of the corresponding integral operator. Our results show that, independently of the practicality of loading the classical data on a quantum computer in logarithmic time, achieving exponential quantum advantage for GPR requires novel approaches.

GPR is a non-parametric machine learning technique with diverse applications [11]. Given training data in the form of input-output pairs, it models the underlying function as a Gaussian Process. The task is then to determine the predictive distribution at a test point given training data. From here samples of the function evaluated at the test points can be drawn, yielding predicted outputs.

Suppose we have some training data $\{x_i, y_i\}_{i=1}^m$ with inputs $x_i \in \chi$, where χ is a measurable space and noisy outputs $y_i \in \mathbb{R}$. We aim to model a function $f(x)$ that satisfies:

$$y = f(x) + \epsilon \tag{1}$$

Here ϵ denotes the noise in the outputs and we assume that $\epsilon \sim N(0, \sigma_n^2)$.

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. In the scenario described here we treat $f(x_i)$ as Gaussian random variables. The Gaussian process has a prior distribution specified by a mean

function $\bar{m}(x)$ and a covariance function $k(x, x')$ given by:

$$\bar{m}(x) = \mathbb{E}[f(x)] \quad (2)$$

$$k(x, x') = \mathbb{E}[(f(x) - \bar{m}(x))(f(x') - \bar{m}(x')))] \quad (3)$$

Both functions can be chosen by the user. Typically a choice of $\bar{m}(x) \equiv 0$ is used, but if there is prior information suggesting otherwise then this can be altered. The covariance function (or kernel) specifies how closely related two function values should be at different distances. Generally, the covariance function determines how smooth and stationary the considered functions should be.

For a single test point x_* , the distribution of $f(x_*)$ (denoted by f_* for simplicity), conditional on the training data and x_* , is Gaussian. We denote the mean and variance of this distribution by $\bar{f}_*$ and $\mathbb{V}[f_*]$ respectively, such that:

$$p(f_* | x_*, \{x_i, y_i\}) \sim N(\bar{f}_*, \mathbb{V}[f_*])$$

We let K denote the $m \times m$ matrix $[k(x_i, x_j)]_{i,j}$ where i, j range over all the training data and m denotes the number of training points. We also let $\mathbf{k}_*$ denote the vector $[k(x_i, x_*)]_{i=1}^m$. Then with the choice of mean function $\bar{m}(x) = 0$ it can be shown that:

$$\bar{f}_* = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y} \quad (4)$$

$$\mathbb{V}[f_*] = k(x_*, x_*) - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_* \quad (5)$$

where $\mathbf{y}$ is the vector of training outputs. A full derivation can be found in [11]. Calculating these is how inference is performed using GPR.

It is clear that the largest computational cost of performing GPR is the matrix inversion of $K + \sigma_n^2 I$. In a classical setting this is typically performed via a Cholesky decomposition, leading to a time complexity of $O(m^3)$. Here m denotes the number of training points which coincides with the size of the matrix. There are currently three different quantum algorithms proposed for the problem of GPR.

Zhao et al. [12] published the original quantum GPR algorithm, employing HHL to perform the matrix inversion. The use of HHL also means that this algorithm relies on the assumption that the matrix K is sparse in order to achieve an exponential speedup.

Chen et al. [13] highlight some issues with Zhao's algorithm that they aim to rectify. They point out that Zhao's algorithm relies on the assumption that preparing $|k_*\rangle$ and simulating $K + \sigma_n^2 I$ can be done efficiently. Here $|k_*\rangle$ is a quantum state representing the classical vector $\mathbf{k}_*$. In fact, Chen's algorithm is able to remove the precomputation of K completely, meaning that it takes only the training and test data as input rather than the covariance matrix. This is promising as it means that this algorithm makes very few assumptions about quantum state preparation. They achieve this by encoding the data as approximate coherent states. The advantage of this is that given two states encoding classical vectors $|x_i\rangle |x_j\rangle$, we have $\langle x_i | x_j \rangle \approx k(x_i, x_j)$, where this approximation can be taken to arbitrary precision. However, one downside to this algorithm is that the kernel choice is limited to

the RBF kernel and the number of qubits required seems to be high. Also, in the best case the runtime is $O(m \log(m))$, achieving only a small polynomial speedup.

The final algorithm by Farooq et al. [14] uses a Hilbert space basis function approximation to remove the dependence on sparsity that arises in Zhao’s algorithm. This enables them to reduce the dependence on m in the time complexity to logarithmic. We include the complexity of each algorithm in Table 1.

Algorithm	Time Complexity
Zhao et al.[12]	$O(\log(m)\kappa^2 s^2/\epsilon)$
Chen et al.[13]	$O(\frac{1}{\sqrt{P_k}} d \log(\frac{d}{\delta}) m \log(m) \epsilon^{-3} \kappa)$
Farooq et al.[14]	$O(\text{poly}(\log(mM)) \log(M) \epsilon^{-3} \kappa^2)$

Table 1: Comparison of complexities for quantum GPR algorithms. κ and s denote the condition number and sparsity of K respectively. M denotes the number of basis functions needed for the approximation in the algorithm of [14]. In the algorithm of [13] the input data is taken from $\mathbb{R}^d$ and P_k denotes the probability of successfully preparing the state $|k_*\rangle$. Finally ϵ and δ both denote chosen precisions at various steps in each algorithm.

We note that all three of these algorithms have polynomial dependence on the condition number κ , and as such any claims about these algorithms providing a speedup are usually met with the assumption that the matrix is well conditioned, essentially meaning that the condition number scales as $O(\log(m))$. In this paper we investigate how reasonable of an assumption this is and show that in a wide setting the condition number instead scales linearly with m .

We note that the dependency of quantum linear system solvers (QLSS) on the condition number has been improved after some of these algorithms for GPR have been published. The state-of-the-art is [17], which has a runtime linear in κ , and a quantum algorithm for GPR that employs this paper’s solver is similarly affected by our results.

The complexity given for Zhao et al. [12] in table 1 is for their original algorithm using HHL. The authors discuss how their algorithm can be generalised for use with QLSS other than HHL. We consider this in more detail in section 3.

Both the condition number of random matrices and the eigenvalues of kernel matrices have been studied somewhat extensively in the machine learning and pure mathematics literature. Edelman [18] proves that the expected value of the log condition number of a matrix with normally distributed entries scales logarithmically with the matrix size. Tao and Vu [19] consider randomly perturbed matrices and are able to show that the condition number grows polynomially with the matrix size under some mild assumptions. These results are both indicative of an issue regarding the growth of the condition number, but do not consider kernel matrices specifically. Posa [20] performs a numerical analysis of the condition numbers of covariance matrices for several different kernels and Zimmerman

[21] investigates the effect of varying hyperparameters on the condition number, focusing on the RBF kernel and giving arguments as to why it often leads to ill-conditioned kernel matrices.

2 Results

Main Theorems

Let (χ, μ) be a probability space and $k \in L_2(\chi^2, \mu^2)$ be a symmetric, measurable, positive semi-definite kernel function. We define the corresponding integral operator $T_k : L_2(\chi, \mu) \rightarrow L_2(\chi, \mu)$ as follows:

$$T_k(f)(x') = \int_{\chi} k(x, x')f(x)d\mu(x) \quad (6)$$

Then since k is symmetric, T_k is a self adjoint compact operator, so we can apply the following spectral decomposition. That is, the eigenvalues of T_k , denoted $\{\lambda_i\}_{i=1}^{\infty}$, are real, and when written in order of descending absolute value, including multiplicities we have:

$$k(x, y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y) \quad (7)$$

where ϕ_i are the corresponding (normalised) eigenfunctions to each λ_i and the above series converges in L_2 norm. Moreover we note that the eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ are square summable. These results are well known in spectral analysis, see for example [22, 23].

Let $X_i : \Omega \rightarrow \chi$ be independent, μ distributed random variables with sample space Ω . These are the inputs of the GPR algorithm. The matrix K_m is given by $(K)_{i,j} = k(X_i, X_j)$ for $1 \leq i, j \leq m$. Let $\tilde{\lambda}_{n,m}$ denote the n 'th eigenvalue of the $m \times m$ matrix K_m , where the eigenvalues of K_m are written in descending order. That is, for all $m \in \mathbb{N}$,

$$\tilde{\lambda}_{1,m} \geq \tilde{\lambda}_{2,m} \geq \dots \geq \tilde{\lambda}_{m,m} \quad (8)$$

Note that here the matrix K_m is random and so the eigenvalues $\tilde{\lambda}_{n,m}$ are also technically random variables, although we suppress this notation for the most part and only discuss it where necessary. As above we denote the eigenvalues of the integral operator T_k by λ_n for $n \in \mathbb{N}$, where each eigenvalue occurs a number of times equal to its multiplicity and again they are taken to be in descending order. In our current notation, the condition number $\kappa_{m,\sigma}$ of the matrix $K_m + \sigma^2 I$ is given by:

$$\kappa_{m,\sigma} = \frac{\tilde{\lambda}_{1,m} + \sigma^2}{\tilde{\lambda}_{m,m} + \sigma^2} \quad (9)$$

Recall that here σ^2 is often just a regularisation added to the diagonal in order to ensure that the kernel matrix is invertible, but that it also corresponds to the variance in the noise (as per the GPR description above).

We restrict ourselves to the scenario where as m increases, the sampled data used to construct the matrix K_m is kept. That is, instead of drawing new samples for the entire training data every time m is increased, we keep the old data and instead sample one more datapoint to add. This assumption accurately reflects the way that the machine learning algorithm is implemented in practice [11]. We state this formally below and it will be assumed in the proof of theorems 1&2.

Assumption (Nested Data). We assume the Gram matrices K_m are constructed such that $\forall n < m$, $(K_n)_{i,j=1}^n = k(x_i, x_j)$ and $(K_m)_{i,j=1}^m = k(x_i, x_j)$ where $(x_i|i = 1, \dots, n)$ contains the first n elements of $(x_i|i = 1, \dots, m)$ in the same order.

In all of the theorems below we have assumed $\lambda_1 \neq 0$. This is a reasonable assumption as if $\lambda_1 = 0$ then it follows that all eigenvalues of T_k are 0 and hence the kernel is equal to 0 μ^2 almost everywhere on χ^2 . Therefore, this case is of little practical interest.

Theorem 1. Let $k \in L_2(\chi^2, \mu^2)$ be a measurable, symmetric, positive semi-definite kernel function with bounded diagonal. Assume also that $\lambda_1 \neq 0$. Let K_m be the corresponding Gram matrices constructed from i.i.d., μ -distributed, nested data. Finally, let C denote the random variable given by:

$$C(\omega) = \lim_{m \rightarrow \infty} \tilde{\lambda}_{m,m}(\omega). \quad (10)$$

for $\omega \in \Omega$. Then $\forall \sigma > 0$, we have

$$\kappa_{m,\sigma}(\omega) \sim \frac{\lambda_1}{C(\omega) + \sigma^2} m \quad (11)$$

where the above holds with probability 1, in the sense that:

$$\lim_{m \rightarrow \infty} \frac{C + \sigma^2}{\lambda_1 m} \kappa_{m,\sigma} = 1 \quad (a.s) \quad (12)$$

First note that the limit defining C is guaranteed to exist by the nested data assumption. Also since C is a random variable, the coefficient in the asymptotic formula may vary depending on the particular sample X_i . We also have $C(\omega) \equiv 0$ for finite rank kernels and infinite rank kernels that become quickly ill-conditioned. The latter includes typical instances with the RBF and Matern kernels.

Since C is upper bounded by the assumed bound on the diagonal of the kernel, we can also construct fully deterministic upper and lower bounds on the condition number.

Corollary 1.1. Under the assumptions of Theorem 1, we have:

$$\forall \sigma > 0, \kappa_{m,\sigma} = \Theta(m) \quad (a.s) \quad (13)$$

where $\Theta(\cdot)$ denotes an asymptotic upper and lower bound. Note the upper and lower bounds on $\kappa_{m,\sigma}(\omega)$ above are independent of $\omega \in \Omega$, although the threshold for m beyond which the bounds hold may still depend on the sample.

This conclusively shows that for a wide variety of kernels and domains, the condition number of $K + \sigma^2 I$ grows linearly with the size of the matrix when $\sigma > 0$. Moreover this result does not depend on the distribution of the input data, or the particular sample obtained.

We note that the constraint of $\sigma > 0$ is not too restricting here as this is common practice within the machine learning community (since often the matrix K will be singular).

Now we consider the case $\sigma = 0$. In this scenario we impose that the kernel k be strictly positive definite (rather than semi-definite as above). We also require that with probability 1 the random variables $\{X_i\}_{i=1}^m$ (i.e the input data) are distinct. We do this to ensure that the matrix K is invertible, as if it is not the condition number instead has to be defined using the Moore-Penrose Pseudoinverse [24] of K . Under this assumption the condition number of K is given by:

$$\kappa_m = \frac{\tilde{\lambda}_{1,m}}{\tilde{\lambda}_{m,m}} \quad (14)$$

We have the following result:

Theorem 2. Let $k \in L_2(\chi^2, \mu^2)$ be a measurable, symmetric, strictly positive definite kernel function with bounded diagonal such that $\lambda_1 \neq 0$. Let K_m denote the $m \times m$ Gram matrix formed from i.i.d., μ -distributed, nested data, such that the datapoints are almost surely distinct. Then with probability 1, the condition number of K_m scales at least linearly with m .

To be precise, the event $E \subset \Omega$ given by:

$$E = \{\omega \in \Omega \mid \kappa_m(\omega) = \Omega(m)\} \quad (15)$$

occurs with probability 1.

Note we use $\Omega(\cdot)$ to denote an asymptotic lower bound, which should not be confused with Ω denoting the sample space of the random variables.

The lower bound in this case depends on the random variable $\frac{1}{\tilde{\lambda}_{1,1}(\omega)}$, see [Proof of Theorem 2](#). However, this is still sufficient to conclude that with probability 1, the condition number will scale at least linearly with m for any sample.

Whilst positive definiteness of the kernel is enough to ensure $\tilde{\lambda}_{1,1}(\omega) > 0, \forall \omega \in \Omega$ it is not enough to guarantee it is bounded away from 0. This means that we cannot achieve a deterministic lower bound (as in the $\sigma > 0$ case) unless we place further regularity conditions on the kernel.

Together, these results are enough to show that the currently proposed quantum algorithms for GPR scale at least quadratically in the number of datapoints in a wide number of scenarios. In particular all three algorithms in table 1 provide no exponential speedup when compared to a classical approach. They also allow us to show the following corollary:

Corollary 2.1. Let $k \in L_2(\chi^2, \mu^2)$ satisfy the conditions of theorem 2. Then let κ_F denote the condition number of $K_m + \sigma^2 I$ with respect to the Frobenius norm. That is,

$$\kappa_F := \|K_m + \sigma^2 I\|_F \|(K_m + \sigma^2 I)^{-1}\|_2 \quad (16)$$

where $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the Frobenius and spectral norms respectively. Then:

$$\kappa_F = \Omega(m) \quad (17)$$

The collection of kernels, distributions and domains that these results apply to is broad. For example, any bounded kernel will automatically satisfy both $L_2(\mu)$ integrability and boundedness of the diagonal. Moreover, any continuous kernel on a compact domain will also satisfy these conditions by the extreme value theorem. This means that the result applies to a large number of the kernel functions used in machine learning applications. For example, the RBF kernel, the Matern kernel and the rational quadratic kernel [11] fulfill the conditions of this result on an arbitrary domain with arbitrary data distribution. However, unbounded continuous kernels, such as dot product and polynomial kernels, will not satisfy these conditions on non-compact domains (like $\mathbb{R}^d$).

We note that kernels on finite domains are automatically bounded and so fulfill the requirements of all our theorems when i.i.d. data is assumed. This situation arises in the context of graph kernels [25] and string kernels [26].

In addition to our results on the condition number, closely related arguments can be used to show the following about the sparsity and Frobenius norm of kernel matrices.

Theorem 3. Let $k \in L_2(\chi^2, \mu^2)$ be a measurable, bounded, symmetric, positive semi-definite kernel function with $\lambda_1 \neq 0$. Further let $s(m)$ denote the sparsity of the Gram matrix K_m , that is the maximum number of non-zero entries in a row. Then with probability 1 we have:

$$s(m) = \Theta(m) \quad (18)$$

Theorem 4. Let $k \in L_2(\chi^2, \mu^2)$ be a measurable, symmetric, positive semi-definite kernel function with bounded diagonal. Moreover suppose $\lambda_1 \neq 0$ and let $\|\cdot\|_F$ denote the Frobenius norm. Then $\forall \sigma \geq 0$, we have the following with probability 1.

$$\|K_m + \sigma^2 I\|_F = \Omega(m) \quad (19)$$

This is relevant to our discussion on GPR as Zhao's algorithm [12] depends quadratically on the sparsity of $K + \sigma_n^2 I$. Also note that Theorems 3 & 4 do not require nested data.

Numerical Results

We provide numerical verification of the asymptotic formula for the condition number $\kappa_{m,\sigma}$ from theorem 1. In order to do this we require a kernel and measure for which an explicit

formula of the first eigenvalue is known. A simple candidate for this is the RBF kernel with Gaussian measure, which is given by:

$$k(x, y) = \exp\left(-\frac{1}{2l^2} \|x - y\|^2\right) \quad (20)$$

where l is a parameter representing the characteristic length scale of the kernel and $x, y \in \chi \subset \mathbb{R}^d$.

In the case of a $1d$ normal distribution there is an exact spectral decomposition of T_k from Zhu et al. [27].

We are able to generalise their results to a spectral decomposition of T_k in the case of a non-degenerate, multivariate normal distribution. Recall, this corresponds to drawing the sample data $\{X_i\}_{i=1}^m$ for $(K)_{i,j} = k(X_i, X_j)$ from a distribution $N(\vec{\nu}, \Sigma)$. For the proof of this generalisation see Supplementary Note 1. The result states the following; let $\{\nu_i\}_{i=1}^d$ denote the eigenvalues of Σ , note they are all positive as Σ is positive definite. Then for the RBF kernel with the above normal density, the eigenvalues of the operator T_k are indexed by a multi index $\vec{k} \in \mathbb{N}^d$ and are given by:

$$\lambda_{\vec{k}} = \prod_{i=1}^d \left(\frac{2a_i}{A_i}\right)^{1/2} B_i^{k_i} \quad (21)$$

where:

$$\begin{aligned} a_i &= \frac{1}{4\nu_i} \\ b &= \frac{1}{2l^2} \\ c_i &= \sqrt{a_i^2 + 2a_i b} \\ A_i &= a_i + b + c_i \\ B_i &= \frac{b}{A_i} \end{aligned} \quad (22)$$

In particular we see that the largest eigenvalue occurs when $\vec{k} = \vec{0}$. It is a well known heuristic in the literature that the smallest eigenvalue of K_m decays to 0, as the size of the matrix (m) increases, for the RBF kernel with normally distributed input data. This can be easily verified numerically and motivates our choice of $C(\omega) \equiv 0$ in the formula from theorem 1. This then gives the following predicted asymptotic behaviour for the condition number:

$$\kappa_{m,\sigma} \sim \frac{m}{\sigma_n^2} \prod_{i=1}^d \left(\frac{2a_i}{A_i}\right)^{1/2} \quad (23)$$

To perform our numerical analysis, for each m we construct an $m \times m$ Gram matrix from normally distributed data and compute the condition number of $K + \sigma_n^2 I$ directly, we then

plot this for different values of m . We plot on the same graph the predicted complexity formula above also. Note that both figures 1 and 2 are generated from a single realisation of the data. Multiple realisations show identical behaviour.

Note that figures 1,2,3 are generated from a single realisation of the data. Multiple realisations show identical behaviour. Also note that all main figures here were generated using nested data.

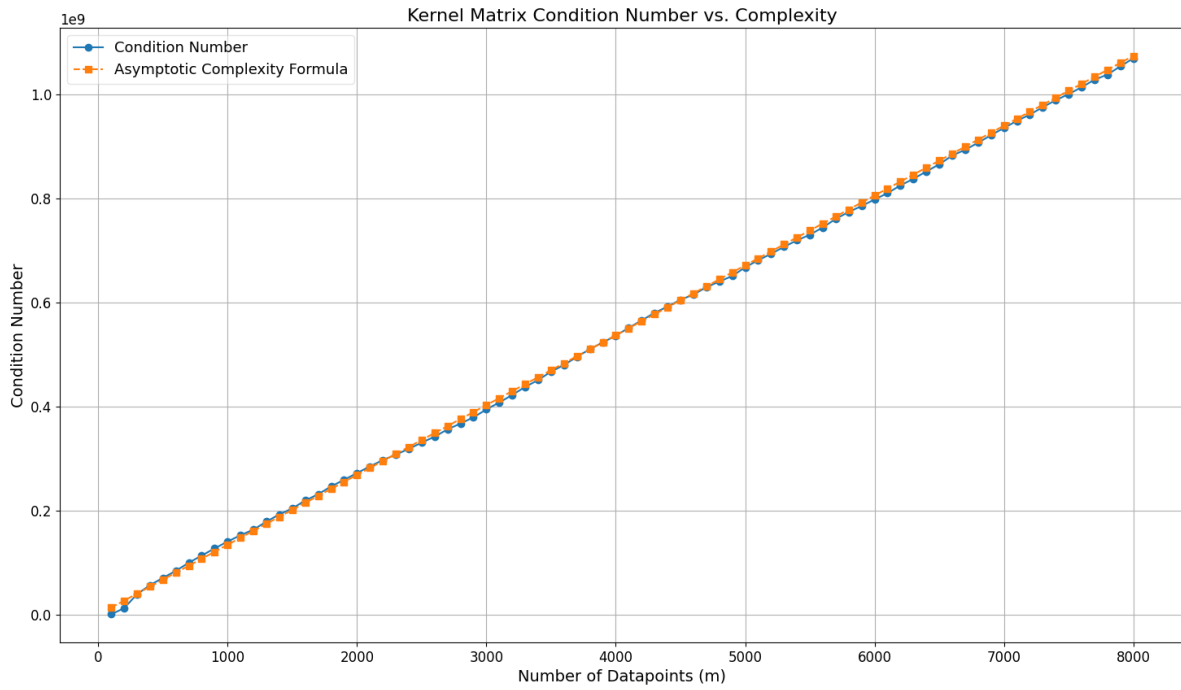


Figure 1: Condition numbers of Gram matrices formed from the RBF kernel with multivariate normal data vs predicted asymptotic behaviour. Parameter values: $\sigma_n = 10^{-3}$, $l = 1$.

Figure 1 shows the comparison between the predicted condition number and the actual condition number for $3d$ normally distributed data with mean $\vec{v} = \vec{0}$ and covariance matrix Σ , where

$$\Sigma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2.5 & -0.5 \\ 0 & -0.5 & 2.5 \end{pmatrix}$$

We see from figure 1 that the asymptotic formula provides a very close approximation even for small values of m .

We also analyse the condition number scaling as l varies, for the RBF kernel and same multivariate distribution. We do this for several large values of m and the results can be seen in figure 2

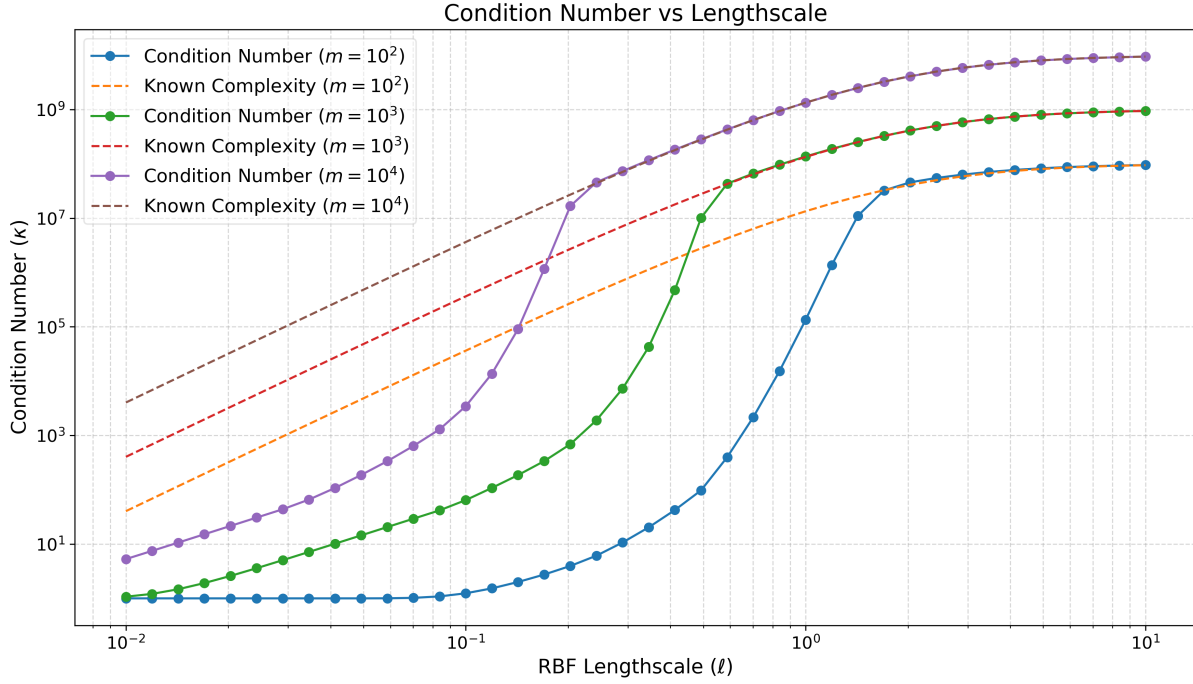


Figure 2: Condition numbers of Gram matrices formed with the RBF kernel from multi-variate normal data vs lengthscale. Parameter values: $\sigma_n = 10^{-3}$.

From figure 2, we can see the accuracy of the asymptotic formula gradually improving up to the $m = 10^4$ value that we consider. It is also clear that convergence to the asymptotic formula is fast when $l \gg 1$ and slower when $l \ll 1$. This may be due to smaller lengthscales causing the kernel to be more local, which in turn makes the smallest eigenvalue more sensitive to fluctuations. It would be interesting to see if any formal results on how the lengthscale affects rate of convergence can be proved in a future work.

In an attempt to verify our results beyond a toy distribution, we also analyse the condition number scaling for data from the California housing market [28]. The dataset consists of eight real valued inputs (features) such as median income, house age and average rooms as well as a real valued output (target) of median house price. This dataset has been selected as we expect the i.i.d. assumptions in our results to be approximately true in this case. For this simulation we take a random sample from the dataset and standardise it (as initially several of the features have incompatible units).

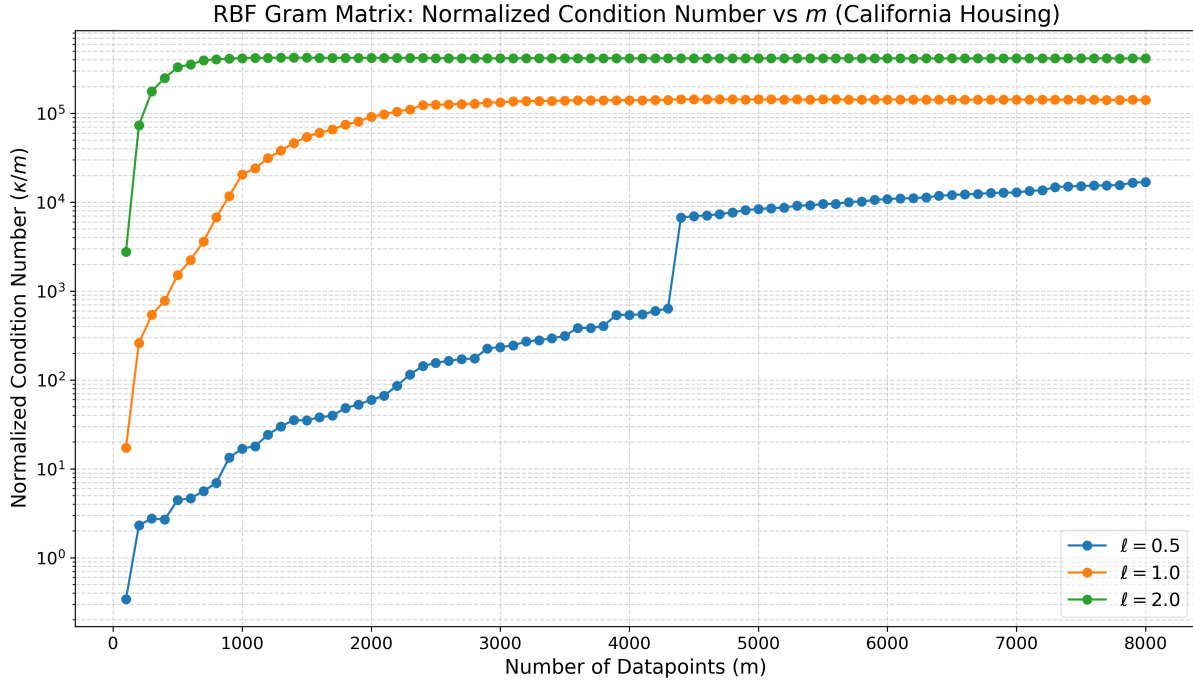


Figure 3: Plot of κ/m vs m for Gram matrices formed using the RBF kernel at various lengthscales. Data is drawn from the California Housing Market dataset. Parameter values: $\sigma_n = 10^{-3}$

We see from figure 3 that κ/m is asymptotically constant for $l = 1, 2$, supporting the results of theorem 1. For $l = 0.5$ the graph has not quite reached constant scaling, although this is consistent with the slower convergence we expect from figure 2. This shows that our results extend beyond theory and can be accurate on real datasets.

In addition to the above, we also perform simulations using the lengthscale that maximises the log marginal likelihood. The optimization for this was performed over the entire dataset and returned a value of $l \approx 0.207$. This more accurately reflects the way GPR is implemented in practice.

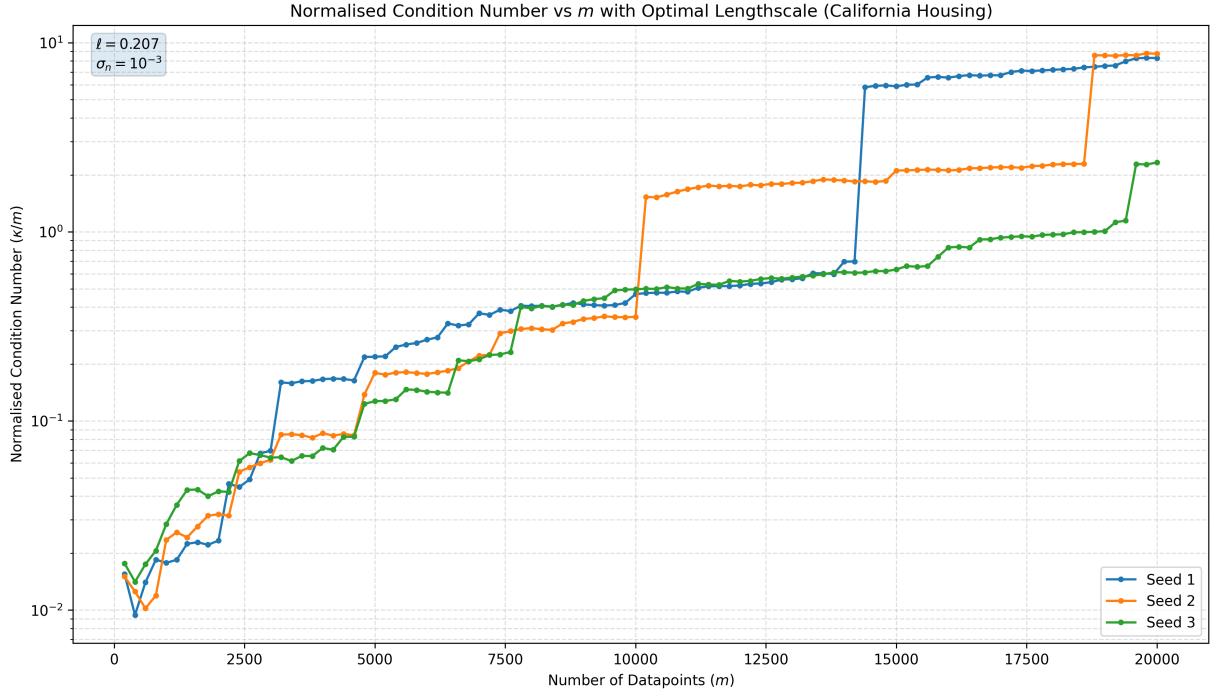


Figure 4: Plot of κ/m vs m for Gram matrices formed using the RBF kernel and lengthscale chosen by LML maximisation. Data is drawn from the California Housing Market dataset with different initial shuffles. Parameter values: $\sigma_n = 10^{-3}$, $l = 0.207$

In figure 4 we show 3 different simulations, each with a different seed for the initial shuffle of the data. Again, the data is standardised before any computations.

Despite the small lengthscale, there are patches of these graphs in which the ratio appears constant. However, in between these there are sharp increases in the condition number that are reminiscent of a step function. It appears that these jumps occur when a new datapoint is too close to one that is already in the Gram matrix, causing sharp decreases in the smallest eigenvalue $\tilde{\lambda}_{m,m}$. This is intuitive as if two input vectors are identical then the resulting Gram matrix will have two identical rows and hence be singular. Indeed we were able to verify that the points at which the jumps occurred were a small distance away from previous datapoints. In particular, this distance was two orders of magnitude smaller than the minimum pairwise distance of points occurring before the jump.

As m increases, the smallest eigenvalue $\tilde{\lambda}_{m,m}$ will eventually approach its limit C or drop below the value of σ_n^2 (if $C < \sigma_n^2$). In either case, the linear scaling of $\tilde{\lambda}_{1,m}$ will begin to dominate and κ/m will appear asymptotically constant. However, in the above simulation we have exhausted almost the entire dataset and so it is not possible to see this behaviour unless we remove points that are too close together. This shows a limitation of our results; their asymptotic nature means they are incapable of capturing finite sample behaviour. This is, of course, the case with all asymptotic results but it would be interesting to see how the finite sample behaviour can be modelled in a future work.

In addition to the above we provide several supplementary figures. Supplementary Figure 1 verifies that our results hold for all values of σ_n , Supplementary Figure 2 gives evidence that our results may hold for unbounded kernels and Supplementary Figure 3 gives evidence that our results may hold without nested data.

3 Discussion

We have proven that under a mild data assumption, a square integrable kernel with bounded diagonal and i.i.d. data will give a regularised Gram matrix whose condition number scales linearly in m and unregularised Gram matrix whose condition number scales at least linearly in m . Moreover, if the kernel is additionally bounded then the sparsity also scales linearly in m . Under these assumptions, we can put the scaling of all three quantum GPR algorithms in terms of m and obtain the complexities in Table 2.

Algorithm	Updated Time Complexity
Zhao[12]	$O(\log(m)m^4/\epsilon)$
Chen [13]	$O(\frac{1}{\sqrt{P_k}}d\log(\frac{d}{\delta})m^2\log(m)\epsilon^{-3})$
Farooq [14]	$O(\text{poly}(\log(mM))\log(M)\epsilon^{-3}m^2)$
Classical Cholesky	$O(m^3)$

Table 2: Comparison of updated complexities for quantum GPR algorithms, in the case of bounded kernels with i.i.d. data and $\sigma_n > 0$. For definitions of the other variables see Table 1.

We see that in this scenario these algorithms scale polynomially with the number of datapoints. Moreover, the best options are only linearly better than the standard classical approach. Therefore we can conclude that, for a wide variety of kernels, these algorithms offer at most a speedup from $O(m^3)$ to $O(m^2\log(m))$. We observe that this reduction in speedup occurs even if QRAM [29] access is allowed for the quantum models, which is a reasonably strong assumption. This provides an argument against the construction of a QRAM in the case of quantum GPR.

The comparison in Table 2 is with exact matrix inversion but there are classical methods that give approximate solutions in similar time to the complexities listed. Reduced rank Hilbert space methods [30] scale as $O(mM^2)$ where M is the number of basis functions used for the approximation.

However, some classical algorithms have time complexities dependent on condition numbers or sparsity. These are then similarly affected by our results. Some notable examples are conjugate gradient methods [15] and quantum inspired classical algorithms [16]. The conjugate gradient algorithm has a time complexity of $O(sm\sqrt{\kappa}\log(\frac{1}{\epsilon}))$ and the Shao quantum inspired algorithm scales as $\tilde{O}(\kappa_F^4\kappa^2/\epsilon^2)$. Under the assumptions of theorems 1 and 3

these methods will scale as $O(m^{2.5} \log(\frac{1}{\varepsilon}))$ and $\tilde{O}(m^6/\varepsilon^2)$ respectively. A review of further classical techniques can be found in [31].

The discussion for Zhao's algorithm above applies only to its original formulation using HHL. If a different QLSS is used then a better time complexity can be achieved. However, even the state of the art QLSS [17] has time complexity linear in κ and so will also scale at least linearly with the number of datapoints when applied to GPR. Moreover, it is known that such a linear dependence on the condition number is optimal under standard complexity theoretic assumptions [7].

There is a QLSS that scales independently of the condition number and sparsity of a matrix [32]. This algorithm relies on having a decomposition for the matrix in the Pauli basis. The runtime then depends quadratically on the l_1 norm of the coefficients in this decomposition. However, considering the spectral norm of such a matrix, we see by the triangle inequality that this l_1 norm will be lower bounded by the largest matrix eigenvalue. Therefore, under the assumptions of lemma 1, this algorithm will lead to an overall time complexity that is at least quadratic in m for the problem of GPR.

Our results leave the possibility for exponential advantage in the case of unbounded kernels, such as the dot product and polynomial kernels. It is not immediately clear how the condition number scales in this setting but numerics for the dot product kernel seem to indicate similar behaviour, see Supplementary Figure 2.

There are stronger possibilities in the case of non i.i.d. data. Whilst i.i.d. data is a widely used assumption, it is clear that scenarios without i.i.d data can lead to a more favourable scaling of the sparsity and condition number. We can see this explicitly if we consider a kernel with highly structured data and compact support. For instance fix $r > 0$ and consider the kernel

$$k_r(x, y) = \left(1 - \frac{\|x - y\|}{2r}\right)_+^{\lfloor d/2 \rfloor + 1}$$

where $(x)_+ = \max(0, x)$ and we take input data $\{x_i\}_{i=1}^M$, $x_i \in \mathbb{R}^d$ such that $\|x_i - x_j\| = |i - j|r$ for all i, j . Here k_r is a standard Wendland kernel [33] that has been stretched by a factor of $2r$. Note also that this kernel satisfies all of our assumptions as it is bounded and measurable. It is clear that $k_r(x_i, x_j) > 0$ if $\|x_i - x_j\| < 2r$ and $k_r(x_i, x_j) = 0$ otherwise. Hence with this input data, the kernel matrix will only have non-zero entries when evaluated on (x_i, x_j) with $i = j$ or $i = j \pm 1$. This matrix clearly has $O(1)$ sparsity, containing at most 3 non-zero elements in each row and column. It also then follows by the Gershgorin circle theorem (see the proof of Theorem 3) that the condition number of $K + \sigma^2 I$ is $O(1)$ when $\sigma > 0$.

The above example shows how our results do not apply for non-i.i.d. input data. In fact, non-i.i.d. data is a reasonable assumption in several practical applications of GPR. For instance, time series data will often have timestamps that are equally spaced [34] and in spatial applications the data may refer to locations on a fixed grid [35]. These applications are therefore good candidates for quantum advantage. We note that the compact support of the kernel also appears to be an important factor here, and there is subtlety in how one

constructs such a kernel. For example, if we take an RBF kernel and simply set it equal to 0 when $\|x - y\| > r$ for some $r > 0$, then this will generally not be positive semi-definite.

There have been attempts to overcome poor condition number scaling in the literature, with a notable example being preconditioners [36, 37]. However, several quantum preconditioners are affected by our results also. For example, [36] depends polynomially on the sparsity of K and [37] is polynomial in the Frobenius norm. Theorems 3 & 4 therefore imply that a GPR algorithm relying on these will still scale polynomially in m . We note that there is broad literature on classical preconditioners [38] and so a hybrid approach incorporating classical preconditioners may lead to an overall speedup.

Due to the similarity between GPR and other kernel methods, it is natural to consider how our results can be applied in these cases also. Kernel ridge regression (KRR) [39] relies on the ability to invert a matrix $K + \sigma^2 I$ and i.i.d. data is often a reasonable assumption when KRR is applied to passive supervised learning. The linear scaling of the condition number should therefore be taken into account when performing this matrix inversion classically. There has also been a proposal for implementing KRR via quantum linear algebra [40]. The algorithm uses a phase estimation procedure resembling HHL and as a result has a complexity that will scale polynomially with the condition number of $K + \sigma^2 I$. Hence our results imply that, for a bounded kernel with i.i.d. nested data, this algorithm will not achieve an exponential speedup.

The scenario with support vector machines (SVMs) is more subtle. The original proposal for QSVMs [41] relies on inverting the matrix $\hat{F}$, where

$$F = \begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & K + \gamma^{-1} I \end{bmatrix} \quad (24)$$

$$\hat{F} = \frac{1}{\text{Tr}(F)} F \quad (25)$$

and γ is a user specified parameter that also serves as a regularisation. Whilst not immediate, our results imply that $\kappa(\hat{F}) = \Omega(m)$ under the assumptions of Theorem 1. That is, the condition number scales at least linearly here. We give a proof of this in the supplementary material.

However, the authors of the QSVM proposal also give a way to circumvent poor condition number scaling by using a low rank approximation. This is valid when a constant number of the eigenvalues of $\hat{F}$ are $O(1)$ and the remaining eigenvalues are $O(1/m)$. In this case, one can ignore the smaller eigenvalues when inverting the matrix, so that the runtime instead depends on the reciprocal of the smallest eigenvalue considered. When the low rank assumptions are valid, this will only incur an error of $O(1/\sqrt{m})$. Whilst this will be a valid approximation in some scenarios, we find that for a bounded kernel with i.i.d. data this low rank approximation will only be valid for finite rank kernels. That is, kernels whose integral operator has finitely many non-zero eigenvalues. For a brief proof of this see the supplementary material. Hence under our assumptions, the scenarios where QSVMs can achieve an exponential speedup are limited to cases where the kernel is finite rank. There

are of course still opportunities for exponential advantage when data is non-i.i.d. and the low rank approximation holds.

One may then consider if the same low rank approximation is possible in the case of GPR. The rescaling of the matrix is key here, as if $K + \sigma^2 I$ is not normalised by its trace then any fixed eigenvalue is $\Theta(m)$ and so its exclusion will lead to an error scaling polynomially in m . However, our findings above apply in this case also. Meaning that for i.i.d. data with a bounded kernel the low rank approximation is only valid when the kernel is finite rank. Whilst these scenarios are limited, this does show a potential for exponential advantage in the case of finite rank kernels, when this low rank approximation can be used.

It is also natural to consider how these results will impact the area of quantum kernels [42, 43]. Note that we refer to scenarios where the kernel is evaluated on a quantum computer, but the linear algebra is done classically, as ‘quantum kernel methods’. This is distinct from the methods of GPR, QSVMs and KRR described above, which (despite also using kernels) fundamentally rely on quantum linear algebra. Provable advantage can be demonstrated by quantum kernels for artificial problems [44] but it is still unclear whether they exhibit quantum advantage in applicable scenarios [45].

As our treatment of the domain χ has been completely general, our results apply to Gram matrices formed from evaluating a quantum kernel. In this case we hope that our results can help provide an educated choice as to which classical algorithm will be optimal to perform any required matrix inversion. In particular, when making this choice any condition number dependence should be treated as at least linear in the size of the matrix in scenarios where our assumptions are satisfied.

We finally note that there is a connection between kernel methods and neural networks via the neural tangent kernel (NTK) [46, 47]. Specifically, in the infinite width limit the NTK is deterministic and so our result will apply when it is bounded. In this scenario, the linear scaling of the condition number may affect the rate of convergence for training a neural network via gradient descent, as this depends on the smallest Gram matrix eigenvalue [48].

4 Methods

Statement and Proof of Lemmas

In order to prove our results we require the use of two lemmas, which we state below:

Lemma 1. Let $k \in L_2(\chi^2, \mu^2)$ be symmetric, measurable, positive semi-definite, and have bounded diagonal. Let λ_n denote the n 'th eigenvalue of the integral operator T_k and let $\tilde{\lambda}_{n,m}$ denote the n 'th eigenvalue of the $m \times m$ Gram matrix K_m formed from evaluating the kernel on i.i.d. μ -distributed data. Then,

$$\lim_{m \rightarrow \infty} \left\| \left\{ \frac{1}{m} \tilde{\lambda}_{n,m} - \lambda_n \right\}_{n=1}^{\infty} \right\|_{l_2} = 0 \quad (\text{a.s.}) \quad (26)$$

In particular, for any fixed $n \in \mathbb{N}$,

$$\frac{1}{m} \tilde{\lambda}_{n,m} \rightarrow \lambda_n \quad (\text{a.s.}) \quad \text{as } m \rightarrow \infty \quad (27)$$

where convergence in both limits occurs almost surely.

Lemma 2. Under the nested data assumption regarding the construction of the matrices $\{K_m\}_{m=1}^\infty$ and the assumptions of lemma 1, the sequence $\{\tilde{\lambda}_{m,m}\}_{m=1}^\infty$ is decreasing. Therefore, there exists a random variable $C(\omega) \geq 0$ such that for all $\omega \in \Omega$:

$$\lim_{m \rightarrow \infty} \tilde{\lambda}_{m,m}(\omega) = C(\omega) \quad (28)$$

Versions of lemma 1 have been acknowledged in the machine learning literature previously. This result appears in some form as theorem 3.4 in Baker [49], but this treatment is fully deterministic and is proved using a Riemann sum argument.

A treatment where the training inputs are treated as random variables X_i , meaning that the kernel matrix K is random, is given by Koltchinskii and Giné [50]. In this paper the authors predominantly focus on the adjusted matrix that is obtained by deleting the diagonal of the matrix K . They show that the normalised eigenvalues of this matrix converge to the eigenvalues of T_k almost surely, if and only if the operator T_k is Hilbert Schmidt. Note that T_k is Hilbert Schmidt iff $k \in L_2(\chi^2, \mu^2)$.

Their proof of this result goes through for the matrix K (without deleted diagonal) if we have the following:

$$\lim_{R \rightarrow \infty} \lim_{m \rightarrow \infty} \frac{1}{m^2} \sum_{i=1}^m (k - k_R)^2(X_i, X_i) = 0 \quad (a.s) \quad (29)$$

Where the X_i are independent, χ valued, μ distributed random variables, k is a symmetric, real kernel satisfying the series expansion above and k_R has the truncated expansion:

$$k_R(x, y) = \sum_{i=1}^R \lambda_i \phi_i(x) \phi_i(y) \quad (30)$$

It is easy to show that the above condition holds when the kernel k has bounded diagonal, that is:

$$\exists B > 0, \quad \forall x \in \chi, \quad |k(x, x)| \leq B \quad (31)$$

From here the proof of lemma 1 is as follows:

Proof of Lemma 1. The convergence in l_2 norm follows immediately from the argument outlined above and Theorem 3.1 in [50], since the sequences $\{\lambda_n\}_{n=1}^\infty$ and $\{\tilde{\lambda}_{n,m}\}_{n=1}^\infty$ (where the latter is padded with 0s beyond $n = m + 1$) are both non-negative and decreasing.

Non-negativity of both sequences of eigenvalues follows from k being positive semi-definite. The final statement holds as, for any fixed $n \in \mathbb{N}$ and $\forall m \in \mathbb{N}$

$$\left| \frac{1}{m} \tilde{\lambda}_{n,m} - \lambda_n \right|^2 \leq \sum_{i=1}^{\infty} \left| \frac{1}{m} \tilde{\lambda}_{i,m} - \lambda_i \right|^2 \quad (32)$$

$$= \left\| \left\{ \frac{1}{m} \tilde{\lambda}_{n,m} - \lambda_n \right\}_{n=1}^{\infty} \right\|_{l_2} \quad (33)$$

$$\rightarrow 0 \quad (a.s) \quad (34)$$

□

The proof of Lemma 2 relies on the nested data assumption. This assumption is helpful as it means that $\forall m \in \mathbb{N}$, K_{m-1} is a principal submatrix of K_m . Recall that a principal submatrix is a submatrix obtained by repeatedly deleting rows and columns of the same index. This will allow us to apply the following theorem:

Theorem (Cauchy Interlacing Theorem). Let A be an $m \times m$ hermitian matrix with eigenvalues $\{\lambda_i\}_{i=1}^m$ and let B be an $(m-1) \times (m-1)$ principal submatrix of A , with eigenvalues $\{\mu_i\}_{i=1}^{m-1}$. Assume both sets of eigenvalues are taken to be in decreasing orders and repeated according to their multiplicities. Then the two sets of eigenvalues are interlaced in the following way:

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \dots \geq \lambda_{m-1} \geq \mu_{m-1} \geq \lambda_m \quad (35)$$

This result is well known to the linear algebra literature and a short proof of the theorem is given in [51]. This enables us to prove lemma 2:

Proof of Lemma 2. We show first that $\{\tilde{\lambda}_{m,m}\}_{m=1}^{\infty}$ is decreasing.

Note that $\forall m \geq 2$ the matrix K_{m-1} is a principal submatrix of K_m . Moreover each matrix K_m is real symmetric (and hence hermitian) so we can apply the Cauchy Interlacing Theorem. The last two terms of the inequality stated in the theorem give $\tilde{\lambda}_{m-1,m-1} \geq \tilde{\lambda}_{m,m}$, so the sequence is decreasing.

Existence of the limit follows simply because k being positive semi-definite implies that the matrices K_m are also positive semi-definite $\forall m$, therefore we have $\tilde{\lambda}_{m,m} \geq 0 \forall m \in \mathbb{N}$. Hence the sequence $\{\tilde{\lambda}_{m,m}\}_{m=1}^{\infty}$ converges by the monotone convergence theorem. Non-negativity of C follows by positive semi-definiteness of K_m also. □

Recall that since each $\tilde{\lambda}_{m,m}$ is a random variable the limit C is also a random variable

$C : \Omega \rightarrow \mathbb{R}$. However a deterministic upper bound on $C(\omega)$ can be obtained as follows:

$$\begin{aligned} \forall \omega \in \Omega, \forall m \in \mathbb{N}, \\ \tilde{\lambda}_{m,m}(w) &\leq \tilde{\lambda}_{1,1}(w) \\ &= k(X_1(w), X_1(w)) \\ &\leq B \end{aligned} \quad (36)$$

$$\implies \forall \omega \in \Omega, C(\omega) \leq B \quad (37)$$

Where $B > 0$ is the assumed bound on the diagonal of the kernel as above. Hence the random variable obtained in the limit is bounded between 0 and B .

We remark that the convergence of $\tilde{\lambda}_{m,m}(\omega)$ to $C(\omega)$ holds $\forall \omega \in \Omega$, meaning it holds pointwise (or surely).

Proof of Theorems

Now using the lemmas we are ready to prove Theorems 1 & 2.

Proof of Theorem 1. The result immediately follows from Lemma 1 and 2, as if we let $A \subset \Omega$ such that:

$$A = \left\{ \omega \in \Omega \mid \lim_{m \rightarrow \infty} \frac{1}{m} \tilde{\lambda}_{1,m}(\omega) = \lambda_1 \right\} \quad (38)$$

Note that λ_1 is deterministic. Then by Lemma 1, the event A occurs with probability 1. Then $\forall \omega \in A$ we have:

$$\lim_{m \rightarrow \infty} \kappa_{m,\sigma}(\omega) \frac{C(\omega) + \sigma^2}{\lambda_1 m} = \lim_{m \rightarrow \infty} \frac{\tilde{\lambda}_{1,m}(\omega) + \sigma^2}{\tilde{\lambda}_{m,m}(\omega) + \sigma^2} \frac{C(\omega) + \sigma^2}{\lambda_1 m} \quad (39)$$

$$= \lim_{m \rightarrow \infty} \frac{\frac{1}{m} \tilde{\lambda}_{1,m}(\omega) + \frac{1}{m} \sigma^2}{\tilde{\lambda}_{m,m} + \sigma^2} \frac{C(\omega) + \sigma^2}{\lambda_1} \quad (40)$$

$$= \frac{\lambda_1}{C(\omega) + \sigma^2} \frac{C(\omega) + \sigma^2}{\lambda_1}, \text{ By definition of } A \text{ and Lemma 2} \quad (41)$$

$$= 1 \quad (42)$$

Since the event A occurs with probability 1 and the limit holds $\forall \omega \in A$, we conclude that the limit holds almost surely. $\square$

The deduction of corollary 1.1 follows from this as:

$$\forall \omega \in \Omega, \quad 0 \leq C(\omega) \leq B \quad (43)$$

$$\implies \frac{\lambda_1}{B + \sigma^2} m \leq \frac{\lambda_1}{C(\omega) + \sigma^2} m \leq \frac{\lambda_1}{\sigma^2} m \quad \forall m \in \mathbb{N} \quad (44)$$

In addition we can prove theorem 2 as follows:

Proof of Theorem 2. Let $A \subset \Omega$ denote the following event:

$$A = \left\{ \omega \in \Omega \mid \lim_{m \rightarrow \infty} \frac{1}{m} \tilde{\lambda}_{1,m}(\omega) = \lambda_1 \right\} \quad (45)$$

Where again the event occurs with probability 1 by Lemma 1. We also note that for any $\omega \in A$ we have the following by Lemma 2:

$$\frac{1}{m} \kappa_m(\omega) = \frac{\tilde{\lambda}_{1,m}(\omega)}{m \tilde{\lambda}_{m,m}(\omega)} \geq \frac{\tilde{\lambda}_{1,m}(\omega)}{m \tilde{\lambda}_{1,1}(\omega)} \rightarrow \frac{\lambda_1}{\tilde{\lambda}_{1,1}(\omega)} \quad (46)$$

Note that $\tilde{\lambda}_{1,1} = k(X_1, X_1)$. It follows from the above that for any fixed $\omega \in A$

$$\forall \epsilon > 0, \exists M \in \mathbb{N} \text{ s.t. } \forall m \geq M, \quad (47)$$

$$\left| \frac{\frac{1}{m} \tilde{\lambda}_{1,m}}{\tilde{\lambda}_{1,1}} - \frac{\lambda_1}{\tilde{\lambda}_{1,1}} \right| < \epsilon \quad (48)$$

$$\implies \frac{\lambda_1}{\tilde{\lambda}_{1,1}} - \epsilon < \frac{\frac{1}{m} \tilde{\lambda}_{1,m}}{\tilde{\lambda}_{1,1}} \quad (49)$$

Taking $\epsilon = \frac{\lambda_1}{2\tilde{\lambda}_{1,1}}$ we then obtain for any fixed $\omega \in A$, $\exists M \in \mathbb{N}$ s.t. $\forall m \geq M$,

$$\frac{\lambda_1}{2\tilde{\lambda}_{1,1}(\omega)} < \frac{\frac{1}{m} \tilde{\lambda}_{1,m}(\omega)}{\tilde{\lambda}_{1,1}(\omega)} \quad (50)$$

$$\leq \frac{1}{m} \frac{\tilde{\lambda}_{1,m}(\omega)}{\tilde{\lambda}_{m,m}(\omega)} \quad (51)$$

$$= \frac{1}{m} \kappa_m(\omega) \quad (52)$$

Hence $\forall \omega \in A$, $\kappa_m(\omega) = \Omega(m)$ and so $\omega \in E$, which implies $A \subset E$. Since A occurs with probability 1 we conclude that E does also which completes the proof. $\square$

Proof of Corollary 2.1. For any matrix A we have:

$$\|A\|_2 \leq \|A\|_F \quad (53)$$

where $\|\cdot\|_2$ denotes the spectral norm. This gives:

$$\kappa = \|K + \sigma^2 I\|_2 \|(K + \sigma^2 I)^{-1}\|_2 \quad (54)$$

$$\leq \|K + \sigma^2 I\|_F \|(K + \sigma^2 I)^{-1}\|_2 \quad (55)$$

$$= \kappa_F \quad (56)$$

The result then follows in the $\sigma > 0$ case by theorem 1 and in the $\sigma = 0$ case by theorem 2. $\square$

Proof of Theorem 3. Let $A = K + \sigma_n^2 I$ and let $A_{i,j}$ denote the i, j 'th element of A . Consider the largest eigenvalue of K , $\tilde{\lambda}_{1,m}$. Then by the Gershgorin Circle theorem $\exists i \in \{1, \dots, m\}$ such that:

$$|\tilde{\lambda}_{1,m} - k(x_i, x_i)| \leq \sum_{j \neq i} |k(x_i, x_j)| \quad (57)$$

$$\leq s(m)B \quad (58)$$

Where B is the assumed bound on the kernel. Then by the triangle inequality this implies:

$$|\tilde{\lambda}_{1,m}| \leq (s(m) + 1)B \quad (59)$$

Then by lemma 1 we have:

$$\liminf_{m \rightarrow \infty} \frac{(s(m) + 1)B}{m} \geq \lim_{m \rightarrow \infty} \frac{\tilde{\lambda}_{1,m}}{m} \quad (60)$$

$$= \lambda_1 > 0 \quad (61)$$

Where the above holds with probability 1. It then immediately follows that:

$$\liminf_{m \rightarrow \infty} \frac{1}{m} s(m) > 0 \quad (a.s) \quad (62)$$

Finally, we have $s(m) \leq m$ by its definition and so we conclude that with probability 1:

$$s(m) = \Theta(m) \quad (63)$$

$\square$

Proof of Theorem 4. To see the lower bound, note that since $K + \sigma^2 I$ is symmetric, positive semi-definite, its Frobenius norm is given by:

$$\|K + \sigma^2 I\|_F^2 = \sum_{i=1}^m (\tilde{\lambda}_{i,m} + \sigma^2)^2 \quad (64)$$

$$\geq \tilde{\lambda}_{1,m}^2 \quad (65)$$

Therefore,

$$\frac{1}{m} \|K + \sigma^2 I\|_F \geq \frac{1}{m} \tilde{\lambda}_{1,m} \quad (66)$$

$$\rightarrow \lambda_1 > 0 \quad (67)$$

Where the last line holds with probability 1 by lemma 1. $\square$

Acknowledgments This work was supported by the UK EPSRC (EP/W032643/1, EP/Y004752/1, and EP/Z53318X/1), KIST Open Research programme and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00413957). We acknowledge helpful discussions with Joe Fitzsimons and Mario Berta.

Data Availability The California housing dataset is publicly available at the reference provided. Random data for the other plots was generated using numpy.

Code Availability The code used to generate the figures is available at: <https://github.com/Dominic-Lowe/GPR-Condition-Number>

Competing Interests The authors declare no competing interests.

Author Contributions All three authors conceived the idea, DL performed the calculations, proved the theorems and wrote the first draft. All three participated in discussions and writing the final draft. The work was supervised by RB and MSK.

References

- [1] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, September 2017.
- [2] M. Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, September 2022.
- [3] Carlo Ciliberto, Mark Herbster, Alessandro Davide Ialongo, Massimiliano Pontil, Andrea Rocchetto, Simone Severini, and Leonard Wossnig. Quantum machine learning: a classical perspective. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2209):20170551, January 2018.
- [4] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning, November 2013. arXiv:1307.0411.
- [5] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, September 2014. Publisher: Nature Publishing Group.

- [6] Iordanis Kerenidis and Anupam Prakash. Quantum Recommendation Systems. *LIPICs, Volume 67, ITCS 2017*, 67:49:1–49:21, 2017. Artwork Size: 21 pages, 517106 bytes ISBN: 9783959770293 Medium: application/pdf Publisher: Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [7] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for solving linear systems of equations, September 2009. arXiv:0811.3171.
- [8] Samuel Jaques and Arthur G. Rattew. Qram: A survey and critique, 2023.
- [9] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems, May 2019. arXiv:1807.04271.
- [10] Ewin Tang. Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions, August 2021. arXiv:1811.00414.
- [11] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, November 2005.
- [12] Zhikuan Zhao, Jack K. Fitzsimons, and Joseph F. Fitzsimons. Quantum-assisted Gaussian process regression. *Physical Review A*, 99(5):052331, May 2019. Publisher: American Physical Society.
- [13] Meng-Han Chen, Chao-Hua Yu, Jian-Liang Gao, Kai Yu, Song Lin, Gong-De Guo, and Jing Li. Quantum algorithm for Gaussian process regression. *Physical Review A*, 106(1):012406, July 2022.
- [14] Ahmad Farooq, Cristian A. Galvis-Florez, and Simo Säikkä. Quantum-assisted Hilbert-space Gaussian process regression. *Physical Review A*, 109(5):052410, May 2024.
- [15] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994.
- [16] Changpeng Shao and Ashley Montanaro. Faster quantum-inspired algorithms for solving linear systems, April 2023. arXiv:2103.10309 version: 2.
- [17] Pedro C. S. Costa, Dong An, Yuval R. Sanders, Yuan Su, Ryan Babbush, and Dominic W. Berry. Optimal scaling quantum linear systems solver via discrete adiabatic theorem, 2021.
- [18] Alan Edelman. Eigenvalues and Condition Numbers of Random Matrices. *SIAM Journal on Matrix Analysis and Applications*, 9(4):543–560, October 1988. Publisher: Society for Industrial and Applied Mathematics.
- [19] Terence Tao and Van Vu. The condition number of a randomly perturbed matrix, March 2007. arXiv:math/0703307 version: 1.
- [20] D. Posa. Conditioning of the stationary kriging matrices for some well-known covariance models. *Mathematical Geology*, 21(7):755–765, 1989.

- [21] R. Zimmermann. On the condition number anomaly of Gaussian correlation matrices. *Linear Algebra and its Applications*, 466:512–526, February 2015.
- [22] F. Chaitin-Chatelin. *Spectral approximation of linear operators*. Computer science and applied mathematics. Academic Press, 1983. tex.lccn: lc82006744.
- [23] N. Dunford and J.T. Schwartz. *Linear operators, part 2: Spectral theory, self adjoint operators in hilbert space*. Wiley classics library. Wiley, 1988. tex.lccn: 88116254.
- [24] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955.
- [25] Nils M. Kriege, Fredrik D. Johansson, and Christopher Morris. A Survey on Graph Kernels. *Applied Network Science*, 5(1):6, December 2020. arXiv:1903.11835 [cs].
- [26] Ren Qi, Fei Guo, and Quan Zou. String kernels construction and fusion: a survey with bioinformatics application. *Frontiers of Computer Science*, 16(6):166904, January 2022.
- [27] Huaiyu Zhu, Christopher K. I. Williams, Richard Rohwer, Michal Morciniec, and Michal Hammel. Gaussian Regression and Optimal Finite Dimensional Linear Models, 1997.
- [28] P. B. Meyer and J. B. Dunn. California housing dataset, 1997. https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html.
- [29] Samuel Jaques and Arthur G. Rattew. QRAM: A Survey and Critique, May 2023. arXiv:2305.10310.
- [30] Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, March 2020.
- [31] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When Gaussian Process Meets Big Data: A Review of Scalable GPs, April 2019. arXiv:1807.01065 [stat].
- [32] Samson Wang, Sam McArdle, and Mario Berta. Qubit-Efficient Randomized Quantum Algorithms for Linear Algebra. *PRX Quantum*, 5(2):020324, April 2024.
- [33] Holger Wendland. *Compactly supported functions*, page 119–132. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.
- [34] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550, February 2013.
- [35] Andrew Gordon Wilson, Elad Gilboa, Arye Nehorai, and John P. Cunningham. GPatt: Fast Multidimensional Pattern Extrapolation with Gaussian Processes, December 2013. arXiv:1310.5288 [stat].

- [36] B. D. Clader, B. C. Jacobs, and C. R. Sprouse. Preconditioned Quantum Linear System Algorithm. *Physical Review Letters*, 110(25):250504, June 2013. Publisher: American Physical Society.
- [37] Changpeng Shao and Hua Xiang. Quantum circulant preconditioner for a linear system of equations. *Physical Review A*, 98(6):062321, December 2018. Publisher: American Physical Society.
- [38] Jonathan Wenger, Geoff Pleiss, Philipp Hennig, John P. Cunningham, and Jacob R. Gardner. Preconditioning for Scalable Gaussian Process Hyperparameter Optimization, June 2022. arXiv:2107.00243 [cs].
- [39] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 515–521, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [40] Dan-Bo Zhang, Shi-Liang Zhu, and Z. D. Wang. Nonlinear regression based on a hybrid quantum computer, August 2018. arXiv:1808.09607 [quant-ph].
- [41] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum Support Vector Machine for Big Data Classification. *Physical Review Letters*, 113(13):130503, September 2014. Publisher: American Physical Society.
- [42] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, March 2019. Publisher: Nature Publishing Group.
- [43] Maria Schuld and Nathan Killoran. Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters*, 122(4):040504, February 2019. Publisher: American Physical Society.
- [44] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9):1013–1017, September 2021. arXiv:2010.02174 [quant-ph].
- [45] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1):2631, May 2021. Publisher: Nature Publishing Group.
- [46] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks, February 2020. arXiv:1806.07572 [cs].
- [47] Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent, December 2019. arXiv:1902.06720 [stat].

- [48] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient Descent Provably Optimizes Over-parameterized Neural Networks, February 2019. arXiv:1810.02054 [cs].
- [49] Christopher T. H. Baker. Numerical Integration in the Treatment of Integral Equations. In G. Hämmerlin, editor, *Numerische Integration: Tagung im Mathematischen Forschungsinstitut Oberwolfach vom 1. bis 7. Oktober 1978*, pages 44–53. Birkhäuser, Basel, 1979.
- [50] Vladimir Koltchinskii and Evarist Giné. Random matrix approximation of spectra of integral operators. *Bernoulli*, 6(1):113–167, February 2000. Publisher: Bernoulli Society for Mathematical Statistics and Probability.
- [51] Steve Fisk. A very short proof of cauchy’s interlace theorem for eigenvalues of hermitian matrices, 2005.