On the Nyström Approximation for Preconditioning in Kernel Machines

Amirhesam Abedsoltan
CSE, UCSD

Parthe Pandit
C-MInDS, IIT Bombay

Luis Rademacher
Mathematics, UC Davis

Mikhail Belkin
HDSI, UCSD

Abstract

Kernel methods are a popular class of nonlinear predictive models in machine learning. Scalable algorithms for learning kernel models need to be iterative in nature, but convergence can be slow due to poor conditioning. Spectral preconditioning is an important tool to speed-up the convergence of such iterative algorithms for training kernel models. However computing and storing a spectral preconditioner can be expensive which can lead to large computational and storage overheads, precluding the application of kernel methods to problems with large datasets.

A Nyström approximation of the spectral preconditioner is often cheaper to compute and store, and has demonstrated success in practical applications. In this paper we analyze the trade-offs of using such an approximated preconditioner. Specifically, we show that a sample of logarithmic size (as a function of the size of the dataset) enables the Nyström-based approximated preconditioner to accelerate gradient descent nearly as well as the exact preconditioner, while also reducing the computational and storage overheads.

1 INTRODUCTION

Deep neural networks have consistently delivered remarkable performance across a wide range of machine learning tasks, setting unprecedented benchmarks, and reshaping the landscape of data modelling. Recent findings have drawn a connection between certain architectures of these networks (such as wide neural networks) and the more classical kernel methods. In the infinite width limit, neural networks converge to a specific form of kernels known as Neural Tangent Kernels (NTKs) [Jacot et al., 2018]. This insight has reignited enthusiasm for studying and applying kernel methods for large-scale machine learning problems. Moreover, emerging research suggests that in certain contexts, such as problems with limited data, kernel methods can surpass neural networks in performance [Arora et al., 2019, Shankar et al., 2020, Bietti and Bach, 2020]. Additionally, kernel methods have been instrumental in elucidating the complex feature learning dynamics within deep neural networks [Radhakrishnan et al., 2022, Beaglehole et al., 2023]. This intersection of classical kernel methods and deep neural networks offers a promising direction for further research towards a deeper understanding of contemporary model architectures.

This motivates the need for scalable approaches for training kernel models on large datasets. Off-the-shelf linear system solvers are often not scalable. Consequently, new algorithms and implementations for scalable training of kernel models have been proposed over the last few years. These include [Shalev-Shwartz et al., 2007, Pegasos], [Camoriano et al., 2016, Nytro], [Meanti et al., 2020, Rudi et al., 2017, Falkon], [Charlier et al., 2021, KeOps], [Gardner et al., 2018, GPyTorch], [Matthews et al., 2017, GPFlow], [Ma and Belkin, 2017, EigenPro], [Ma and Belkin, 2019, EigenPro2] and [Abedsoltan et al., 2023, EigenPro3]. Some of these can also take advantage of modern hardware such as graphical processing units (GPUs) effectively.

Consider a dataset $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$ and a positive definite kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Let $\mathcal{H}_K$ be the reproducing kernel Hilbert space (RKHS) corresponding to K. We are interested in the following model: the minimum $\mathcal{H}_K$-norm solution to the quadratic loss optimization problem

$$\min_{f \in \mathcal{H}_K} \mathsf{L}(f) = \frac{1}{2n} \sum_{i=1}^{n} (f(x_i) - y_i)^2, \qquad (1)$$

The number of iterations required for an iterative optimization algorithm, such as gradient descent (GD), to

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

Table 1: Trade-offs of using a Nyström approximation to obtain an approximated preconditioner of level q for the preconditioned gradient descent algorithm. Here ε is a tunable error parameter of the Nyström approximation, and $\{\lambda_i^*\}$ is the non-increasing sequence of eigenvalues of the integral operator (defined in (4)) which depends on the data distribution and kernel (but not n). The comparison with gradient descent is for illustration and based on a heuristic calculation provided in Section 3.2 and Equation (16). The speed-up calculation does not include setup time which is significant for PGD, making it impractical for cold-starts.

Iterative algorithm	**Speed-up over GD**	**Storage**	**Setup time**
PGD (Preconditioned gradient descent)	$\frac{\lambda_1^*}{\lambda_q^*}$	qn	qn^2
nPGD (PGD w/ approximated preconditioner)	$\frac{\lambda_1^*}{\lambda_q^*} \cdot \frac{1}{(1+\varepsilon)^4}$	$q \cdot O(\frac{\log^4 n}{\varepsilon^4})$	$q \cdot O(\frac{\log^8 n}{\varepsilon^8})$

Table 2: Notation

n	number of training samples
q	level of preconditioner
s	number of Nyström samples
κ_1	condition number w/o preconditioning
κ_q	condition number w/ preconditioning
$\kappa_{s,q}$	condition number w/ approx. preconditioner

converge depends on the condition number of a specific operator known as the empirical covariance operator $\mathcal{K} : \mathcal{H}_K \to \mathcal{H}_K$. Operator $\mathcal{K}$ is the Hessian $\nabla_f^2 \mathsf{L}$ of the optimization problem above (see Section 3). For our purposes, we define the condition number of an operator $\mathcal{A} : \mathcal{H} \to \mathcal{H}$ by

$$\kappa(\mathcal{A}) := \frac{\sup_{\|f\|=1} \|\mathcal{A}f\|}{\inf_{f \in \mathrm{Null}(\mathcal{A})^\perp, \|f\|=1} \|\mathcal{A}f\|}, \tag{2}$$

where Null$(\cdot)$ is the null-space. A higher condition number means that the iterative algorithm requires more iterations to converge.

In this work we will generally be interested in the condition number of finite-rank operators. For a rank-k operator $\mathcal{A}$ our definition can be written as $\kappa(\mathcal{A}) = \sigma_1(\mathcal{A})/\sigma_k(\mathcal{A})$, where $\sigma_1(\mathcal{A}) \geq \sigma_2(\mathcal{A}) \geq \cdots \geq \sigma_k(\mathcal{A}) > 0$ are the singular values of $\mathcal{A}$ (defined by operator theory, or matrix theory expressing the operator as a matrix in terms of orthonormal bases).

In practice, condition number $\kappa_1 := \kappa(\mathcal{K})$ is often large. A common approach to overcome this challenge is to use a *preconditioner* operator, $\mathcal{P} : \mathcal{H}_K \to \mathcal{H}_K$. This enables solving the same problem with a modified condition number $\kappa(\mathcal{P}\mathcal{K})$ instead of $\kappa(\mathcal{K})$, for example in preconditioned gradient descent (PGD). Many types of preconditioners have been studied in the literature on numerical methods for learning kernel models (see [Cutajar et al., 2016] for a comparison). Approximated preconditioners have also been studied, see for example [Avron et al., 2017] and [Rudi et al., 2017].

Spectral preconditioning. In this paper we focus on a specific type of preconditioner. For the operator $\mathcal{K}$, we say $\mathcal{P}_q$ is a *spectral preconditioner* if the top-q eigenvalues of $\mathcal{P}_q\mathcal{K}$ are equal to the q^{th} largest eigenvalue of $\mathcal{K}$ (and the rest of the eigenvalues stay unchanged), and we refer to q as the *level* of the preconditioner. The resulting condition number is $\kappa_q := \kappa(\mathcal{P}_q\mathcal{K})$. This preconditioner works well in practice as shown in [Ma and Belkin, 2017], which introduced EigenPro — a state-of-the-art method for learning kernel models based on stochastic gradient descent (SGD).

Nyström approximation. However, computing and storing a spectral preconditioner $\mathcal{P}_q$ is challenging. It requires an additional $O(qn^2)$ time [Williams and Seeger, 2000] and $O(qn)$ memory [Ma and Belkin, 2017]. See the discussion following Proposition 1 in Section 3.2 of our paper for details.

The Nyström extension is a technique used to approximate $\mathcal{K}$ with a smaller number of samples, denoted by s, where $s \ll n$. Subsequently, we can approximate spectral preconditioner $\mathcal{P}_q$ using these s samples and denote it as $\mathcal{P}_{s,q}$. This method requires time $O(qs^2)$ and storage $O(qs)$.

Just like PGD we can use this approximated preconditioner to accelerate GD. We refer to this version of PGD with a Nyström approximated preconditioner as (nPGD). The convergence of nPGD depends on

$$\kappa_{s,q} := \kappa(\mathcal{P}_{s,q}^{\frac{1}{2}} \mathcal{K} \mathcal{P}_{s,q}^{\frac{1}{2}}), \tag{3}$$

as shown later in Section 3. In fact, EigenPro2.0 [Ma and Belkin, 2019] applies $\mathcal{P}_{s,q}$ to improve the scalability of EigenPro, which used $\mathcal{P}_q$, the exact preconditioner.

Prior to our work, there was no rigorous way to choose s, the size of the Nyström sample, so that one can

guarantee a speed-up when using the approximated preconditioner.

1.1 Main contribution

We show that nPGD can achieve nearly the same speed-up as PGD does over GD, while only requiring a poly-logarithmic storage overhead and setup-cost. See Table 1 for the trade-off of using a Nyström approximated preconditioner.

In order to show this, we analyze how many Nyström samples are sufficient to achieve a particular approximation quality of approximated preconditioner. Our main result (Theorem 2) shows that for a given $\varepsilon > 0$, we can achieve $\kappa_{s,q} \leq (1+\varepsilon)^4 \kappa_q$ if the number of Nyström samples satisfies $s = \Omega\big(\frac{\log^4 n}{\varepsilon^4}\big)$.

While we only provide explicit speed-up computations for versions of preconditioned GD, our main result can be applied to guarantee speed-ups for other spectrally preconditioned algorithms such as conjugate gradient and Nesterov's accelerated gradient methods.

Organization: In Section 2 we discuss preliminaries needed to state our main result. Section 3 details the problem formulation, and Section 4 provide the statement of the main result in Theorem 2, followed by its proof in Section 5. The intermediate lemmas needed in the proof of Theorem 2 are detailed in Section 6.

2 PRELIMINARIES

Notation: We denote by $\mathcal{H}$ a separable Hilbert space, and by $\mathcal{H}_K$ a reproducing kernel Hilbert space (RKHS) associated with a symmetric positive definite kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. We assume that K is continuous and bounded. For any point $x \in \mathbb{R}^d$, the function $K(x,\cdot) : \mathbb{R}^d \to \mathbb{R}$ belongs to $\mathcal{H}_K$. Let $L(\mathcal{H})$ denote the set of bounded linear operators from $\mathcal{H}$ to $\mathcal{H}$. For operators and when defined, we will denote the Hilbert-Schmidt norm by $\|\cdot\|_{\mathsf{HS}}$ and the operator norm by $\|\cdot\|_{\mathsf{OP}}$. Without any subscript, $\|\cdot\|$ denotes the norm of $\mathcal{H}$ or $\mathcal{H}_K$ when there is no confusion.

Square-root operator: For a self-adjoint positive semidefinite operator $\mathcal{A}$, we denote by $\mathcal{A}^{\frac{1}{2}}$ the unique self-adjoint square-root of $\mathcal{A}$, such that $\mathcal{A}^{\frac{1}{2}}\mathcal{A}^{\frac{1}{2}} = \mathcal{A}$. Thus if $\mathcal{A}$ has an eigen-decomposition $\mathcal{A} = \sum_i \sigma_i \phi_i \otimes \phi_i$ (with $\sigma_i \geq 0$), we define

$$\mathcal{A}^{\frac{1}{2}} := \sum_i \sqrt{\sigma_i}\ \phi_i \otimes \phi_i.$$

Eigenvalue thresholding: We will need the following elementary version of functional calculus. It will be convenient to represent preconditioner operators as the result of modifications to the eigenvalues of a covariance operator. We will need an operation that replaces every eigenvalue below a threshold by the threshold.

For self-adjoint finite rank $\mathcal{A} \in L(\mathcal{H})$ with eigendecomposition $\mathcal{A} = \sum_{i\in I} \mu_i \phi_i \otimes \phi_i$, $\mu_i \neq 0$ and for a continuous function $h : \mathbb{R} \to \mathbb{R}$, the corresponding operator function is $h(\mathcal{A}) = \sum_{i\in I} h(\lambda_i)\phi_i \otimes \phi_i + h(0)\pi_{\mathrm{Null}(\mathcal{A})}$. We will need the following thresholding function: $h_\alpha(x) = \max\{x, \alpha\}$. When $\mathcal{H}$ is finite dimensional (and more generally, but this is all we need), we have $h_\alpha(\mathcal{A}) = h_0(\mathcal{A} - \alpha\mathcal{I}) + \alpha\mathcal{I}$.

3 PROBLEM SETUP

Let ρ be a probability measure on $\mathbb{R}^d$. Let $X_n = \{x_i\}_{i=1}^n$ be the training samples drawn i.i.d. from ρ. Let $(i_1, i_2, \ldots, i_s)$ be a random tuple of $s \leq n$ distinct indices $1 \leq i_k \leq n$ chosen independently of X_n, and define $X_s = \{x_{i_k}\}_{k=1}^s$ as the Nyström samples. Namely, for our results to hold the distribution of the tuple can be arbitrary as long as it is independent of X_n and the indices are all distinct with probability 1. Two natural choices of this distribution are *uniform sampling in* $[n]$ *without replacement* and *any fixed tuple*, e.g. $i_k = k$.

Consider n targets $y_1, y_2, \ldots, y_n \in \mathbb{R}$. Assume that if $x_i = x_j$, then $y_i = y_j$. Our model is the unique minimum $\mathcal{H}_K$-norm solution f^* to problem (1). Assume K is a bounded, continuous, symmetric positive definite kernel. Note that under our assumptions on K we have that f^* interpolates the data, namely the optimal value of problem (1) is 0.

Define the following operators in $L(\mathcal{H}_K)$:

$$\mathcal{T} := \int_{\mathbb{R}^d} K(x,\cdot) \otimes K(x,\cdot)\rho(\mathrm{d}x) \tag{4}$$

$$\mathcal{K} := \frac{1}{n}\sum_{i=1}^n K(x_i,\cdot) \otimes K(x_i,\cdot) \tag{5}$$

$$\mathcal{K}' := \frac{1}{s}\sum_{i=1}^s K(x'_i,\cdot) \otimes K(x'_i,\cdot). \tag{6}$$

For $f \in \mathcal{H}_K$, due to the reproducing property of kernel K, the above operators act as follows: $\mathcal{T}f(x) = \int K(x,z)f(z)\rho(\mathrm{d}z)$, $\mathcal{K}f(x) = \frac{1}{n}\sum_{i=1}^n K(x,x_i)f(x_i)$, and $\mathcal{K}'f(x) = \frac{1}{s}\sum_{i=1}^s K(x,x'_i)f(x'_i)$.

By the linearity of the trace we can show that

$$\mathrm{trace}(\mathcal{T}) \leq \beta(K), \quad \text{and} \quad \mathrm{trace}(\mathcal{K}) \leq \beta(K), \tag{7}$$

where we define

$$\beta(K) := \max_{x\in\mathbb{R}^d} K(x,x). \tag{8}$$

In (7) we have used the fact that $\text{trace}(K(x,\cdot) \otimes K(x,\cdot)) = \langle K(x,\cdot), K(x,\cdot)\rangle_{\mathcal{H}_K} = K(x,x)$.

Next, let ψ_i^* be an eigenfunction of $\mathcal{T}$ with eigenvalue λ_i, i.e., $\mathcal{T}\psi_i^* = \lambda_i\psi_i^*$. Similarly denote by (λ_i, ψ_i) the eigenpairs of $\mathcal{K}$ and by (λ_i', ψ_i') the eigenpairs for $\mathcal{K}'$. Note that $\mathcal{T}$ is a compact operator whereas $\mathcal{K}$ and $\mathcal{K}'$ are empirical approximations of $\mathcal{T}$. Furthermore, $\mathcal{T}, \mathcal{K}$, and $\mathcal{K}'$ have non-negative eigenvalues which we assume are ordered as $\lambda_1^* \geq \lambda_2^* \geq \cdots \geq 0$, and similarly $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$, and $\lambda_1' \geq \lambda_2' \geq \cdots \geq \lambda_s' \geq 0$. Hence we have eigen-decompositions for these operators as written below.

$$\mathcal{T} = \sum_{i=1}^{\infty} \lambda_i^* \cdot \psi_i^* \otimes \psi_i^*, \qquad \mathcal{K} = \sum_{i=1}^{n} \lambda_i \cdot \psi_i \otimes \psi_i,$$
$$\mathcal{K}' = \sum_{i=1}^{s} \lambda_i' \cdot \psi_i' \otimes \psi_i'.$$

For a fixed $q \leq s$, we define the following preconditioners with the assumption that $\lambda_q, \lambda_q' > 0$ as below,

$$\mathcal{P}_q := \mathcal{I} - \sum_{i=1}^{q-1} \left(1 - \frac{\lambda_q}{\lambda_i}\right) \psi_i \otimes \psi_i \tag{9}$$

$$\mathcal{P}_{s,q} := \mathcal{I} - \sum_{i=1}^{q-1} \left(1 - \frac{\lambda_q'}{\lambda_i'}\right) \psi_i' \otimes \psi_i'. \tag{10}$$

With the above definition, one can verify that

$$\mathcal{P}_{s,q}^{\frac{1}{2}} = \mathcal{I} - \sum_{i=1}^{q-1} \left(1 - \sqrt{\frac{\lambda_q'}{\lambda_i'}}\right) \psi_i' \otimes \psi_i'. \tag{11}$$

3.1 Gradient Descent and preconditioning

Define $\mathrm{b} := \frac{1}{n}\sum_{i=1}^n K(\cdot, x_i) y_i \in \mathcal{H}_K$, and note that $\mathrm{b} = \mathcal{K} f^*$.

Due to the reproducing property of the RKHS, we have $f(x) = \langle f, K(x,\cdot)\rangle_{\mathcal{H}_K}$ for all $f \in \mathcal{H}_K$ and for all $x \in \mathbb{R}^d$, whereby the Fréchet derivative $\nabla_f f(x)$ equals $K(\cdot, x)$ for all $f \in \mathcal{H}_K$ and $x \in \mathbb{R}^d$. Using a similar argument, observe that $\nabla_f \mathsf{L}(f) = \mathcal{K} f - \mathrm{b} = \mathcal{K}(f - f^*)$, and $\mathcal{K} = \nabla_f^2 \mathsf{L}(f)$ is the Hessian of the optimization problem (1).

One step of Gradient Descent (GD) with learning rate η is given by:

$$\begin{aligned} f^t &= f^{t-1} - \eta \nabla_f \mathsf{L}(f^{t-1}) \\ &= (\mathcal{I} - \eta\mathcal{K}) f^{t-1} + \eta \mathrm{b}. \end{aligned} \tag{GD}$$

where we have used the fact that $\nabla_f \mathsf{L}(f) = \sum_{i=1}^n (f(x_i) - y_i) K(x_i, \cdot) = \mathcal{K} f - \mathrm{b}$.

Then, for the purpose of convergence analysis, one can add and subtract f^* and verify that the above equation can be rewritten as,

$$f^t - f^* = (\mathcal{I} - \eta\mathcal{K})(f^{t-1} - f^*)$$

where we have used the fact that $\mathrm{b} = \mathcal{K} f^*$.

The convergence rate of this iteration is governed by the condition number of the operator $\mathcal{K}$. Standard proof techniques for gradient descent, e.g. [Garrigos and Gower, 2023, Theorem 3.6] allow us to show that with the choice $\eta = 1/\kappa(\mathcal{K}) = 1/\kappa_1$ we can find f^t such that

$$\left\| f^t - f^* \right\|_{\mathcal{H}_K}^2 \leq \tau \left\| f^0 - f^* \right\|_{\mathcal{H}_K}^2$$

in $t = \kappa_1 \log\left(\frac{1}{\tau}\right)$ iterations.

Preconditioned gradient descent: Using the preconditioner $\mathcal{P}_q$ with level q, the update equations for preconditioned gradient descent with step size η_q are

$$\begin{aligned} f^t &= f^{t-1} - \eta_q \mathcal{P}_q \nabla_f \mathsf{L}(f^{t-1}) \\ &= (\mathcal{I} - \eta_q \mathcal{P}_q \mathcal{K}) f^{t-1} + \eta \mathcal{P}_q \mathrm{b}. \end{aligned} \tag{PGD}$$

whose convergence rate is determined by the condition number of $\mathcal{P}_q\mathcal{K}$ (since $\mathcal{P}_q$ and $\mathcal{K}$ commute and therefore $\mathcal{P}_q\mathcal{K}$ is self-adjoint). A similar argument as for GD shows that PGD is equivalent to

$$f^t - f^* = (\mathcal{I} - \eta_q \mathcal{P}_q \mathcal{K})(f^{t-1} - f^*).$$

With the choice $\eta_q = 1/\kappa(\mathcal{P}_q\mathcal{K}) = 1/\kappa_q$, the iteration converges within $\kappa_q \log\left(\frac{1}{\tau}\right)$ steps.

Approximated preconditioner: Preconditioning with an approximated preconditioner $\mathcal{P}_{s,q}$ modifies the update rule to:

$$\begin{aligned} f^t &= f^{t-1} - \eta_{s,q} \mathcal{P}_{s,q} \nabla_f \mathsf{L}(f^{t-1}) \\ &= (\mathcal{I} - \eta_{s,q} \mathcal{P}_{s,q} \mathcal{K}) f^{t-1} + \eta_{s,q} \mathcal{P}_{s,q} \mathrm{b}. \end{aligned} \tag{nPGD}$$

which can similarly be shown to be equivalent to

$$f^t - f^* = (\mathcal{I} - \eta_{s,q} \mathcal{P}_{s,q} \mathcal{K})(f^{t-1} - f^*).$$

Observe that $\mathcal{P}_{s,q}^{\frac{1}{2}}$ is invertible, whereby introducing a change of variables, $g^t = \mathcal{P}_{s,q}^{-\frac{1}{2}} f^t$, and $g^* = \mathcal{P}_{s,q}^{-\frac{1}{2}} f^*$, we can show nPGD is equivalent to

$$g^t - g^* = (\mathcal{I} - \eta_{s,q} \mathcal{P}_{s,q}^{\frac{1}{2}} \mathcal{K} \mathcal{P}_{s,q}^{\frac{1}{2}})(g^{t-1} - g^*)$$

whose convergence rate is determined by the condition number of the self-adjoint operator $\mathcal{P}_{s,q}^{\frac{1}{2}} \mathcal{K} \mathcal{P}_{s,q}^{\frac{1}{2}}$. Specifically with $\eta_{s,q} = 1/\kappa(\mathcal{P}_{s,q}^{\frac{1}{2}} \mathcal{K} \mathcal{P}_{s,q}^{\frac{1}{2}}) = 1/\kappa_{s,q}$, the iteration converges in $\kappa_{s,q} \log\left(\frac{1}{\tau}\right)$ steps.

Remark 1. *We consider $\mathcal{P}_{s,q}^{\frac{1}{2}}\mathcal{K}\mathcal{P}_{s,q}^{\frac{1}{2}}$ to deal with the fact that $\mathcal{P}_{s,q}\mathcal{K}$ is not self-adjoint. This issue does not occur for $\mathcal{P}_q\mathcal{K}$, since $\mathcal{P}_q$ and $\mathcal{K}$ commute whereby $\mathcal{P}_q\mathcal{K}$ is self-adjoint.*

Our main result in the next section provides sufficient conditions on s, under which we can obtain a multiplicative approximation of $\kappa(\mathcal{P}_{s,q}^{\frac{1}{2}}\mathcal{K}\mathcal{P}_{s,q}^{\frac{1}{2}})$ in terms of $\kappa(\mathcal{P}_q\mathcal{K})$.

3.2 Speed-up of PGD over GD

Here we provide a summary of the computational and storage costs of three algorithms — gradient descent (GD), preconditioned gradient descent (PGD), and preconditioned gradient descent with a Nyström approximated preconditioner (nPGD).

The time for running GD is

$$T_{\mathsf{GD}} = \kappa_1 n^2 \log\left(\tfrac{1}{\tau}\right) \tag{12}$$

where n^2 is the time per iteration for calculating $\nabla_f \mathsf{L}(f^t)$ and $\kappa_1 \log\left(\frac{1}{\tau}\right)$ is the number of iterations required to achieve error τ. Note the n^2 complexity arises since we need to compute $\nabla_f \mathsf{L}(f^t) = \mathcal{K}f^t$. If we use the basis expansion $f^t = \sum_{i=1}^n K(x_i,\cdot)\alpha_i$, we have $\mathcal{K}f^t = \sum_{i=1}^n \sum_{j=1}^n K(x_j,\cdot)K(x_i,x_j)\alpha_i$, which requires n^2 steps. One can argue that the quadratic complexity is optimal under our assumptions.

Similarly, the time taken for preconditioned gradient descent is

$$T_{\mathsf{PGD}} = \kappa_q \left(n^2 + 2qn\right) \log\left(\tfrac{1}{\tau}\right), \tag{13}$$

where $2nq$ is the per iteration overhead to apply the preconditioner, i.e. calculating $\nabla_f \mathsf{L}(f^t) \mapsto \mathcal{P}_q \nabla_f \mathsf{L}(f^t)$. Thus the speed-up of PGD over GD is

$$\frac{T_{\mathsf{GD}}}{T_{\mathsf{PGD}}} = \frac{\kappa_1}{\kappa_q + q/n} \approx \frac{\kappa_1}{\kappa_q} = \frac{\lambda_1}{\lambda_q} \approx \frac{\lambda_1^*}{3\lambda_q^*} \tag{14}$$

when $q = O(1)$, and the last approximation holds because of a concentration argument that allows us to control $|\lambda_i - \lambda_i^*|$ for large enough n, see equation (27).

Representing the spectral preconditioner $\mathcal{P}_q \equiv \{\psi_i\}_{i=1}^q$ requires storing q vectors of length n as explained in the following lemma.

Proposition 1. *Let $\boldsymbol{e} = (e_j) \in \mathbb{R}^n$ be an eigenvector of the matrix $(K(x_j,x_k))_{1\le i,j\le n}$, with eigenvalue $n\lambda$, then $\psi = \sum_{j=1}^n e_j K(\cdot,x_j)$ is an eigenfunction of $\mathcal{K}$, with eigenvalue λ.*

Proof. Observe that if ψ is defined as above, $n\mathcal{K}\psi = n\sum_{j=1}^n e_j \mathcal{K}K(\cdot,x_j) = \sum_{j,k=1}^n e_j K(x_k,x_j)K(\cdot,x_k) = \sum_{k=1}^n K(\cdot,x_k)\left(\sum_{j=1}^n K(x_k,x_j)e_j\right)$. The term in the parenthesis equals $n\lambda e_k$, since $\boldsymbol{e}$ is an eigenvector. □

Thus storing the preconditioner $\mathcal{P}_q$ requires storing qn floats. Additionally, the cost of calculating the q eigenvectors needed to represent $\{\psi_i\}_{i=1}^q$ is $O(qn^2)$.

Similar to (13), for nPGD we have

$$T_{\mathsf{nPGD}} = \kappa_{s,q}\left(n^2 + 2sn + 2sq\right)\log\left(\tfrac{1}{\tau}\right), \tag{15}$$

where $2ns + 2nq$ is the per iteration overhead in calculating $\nabla_f \mathsf{L}(f^t) \mapsto \mathcal{P}_{s,q}\nabla_f \mathsf{L}(f^t)$. Hence we get that

$$\frac{T_{\mathsf{GD}}}{T_{\mathsf{nPGD}}} = \frac{\kappa_1}{\kappa_{s,q} + s/n + sq/n^2} \approx \frac{\kappa_1}{\kappa_{s,q} + s/n} \tag{16}$$

when $q = O(1)$.

An argument similar to the one following Proposition 1 for PGD shows that representing $\mathcal{P}_{s,q} \equiv \{\psi_i'\}_{i=1}^q$ requires storing eigenvectors of the $s \times s$ matrix $\left(K(x_j',x_k')\right)_{1\le j,k\le s}$, which requires storing qs floats, while computing q eigenvectors requires time $O(qs^2)$.

4 MAIN RESULT

The main result of the paper provides a multiplicative bound on the condition number for the nPGD iterations in terms of the condition number for PGD. To that end, recall the definitions of $\beta(K)$ from (8), and the condition number κ of an operator from (2).

We are ready to state the main result of this paper.

Theorem 2. *Assume that kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is symmetric positive definite, continuous and bounded. Consider the operators $\mathcal{K}$, $\mathcal{P}_q$, and $\mathcal{P}_{s,q}$ defined in equations (5),(9), and (10), respectively. Let $q \ge 1$ be such that $\lambda_q^* > 0$. For any $\varepsilon > 0$ and $\delta \in (0,1)$, we have*

$$\kappa(\mathcal{P}_{s,q}^{\frac{1}{2}}\mathcal{K}\mathcal{P}_{s,q}^{\frac{1}{2}}) \le (1+\varepsilon)^4\,\kappa(\mathcal{P}_q\mathcal{K})$$

with probability at least $1-\delta$ (over the randomness of X_n and X_s) when

$$s \ge \max\left\{c_1 C_{K,q}^2,\ \frac{c_2 C_{K,q}^4 \log^4(n+1)}{\varepsilon^4}\right\}\log\left(\tfrac{4}{\delta}\right), \tag{17}$$

where $C_{K,q} := \frac{\beta(K)}{\lambda_q^*}$, and c_1, c_2 are universal constants.

4.1 Speed-up of nPGD over GD

Consider the fact that the per iteration complexity of all 3 algorithms GD, PGD and nPGD are $O(n^2)$. Consequently the speed-up in PGD and nPGD over GD arises due to fewer number of iterations.

The level of the preconditioner q is implicitly constrained due to (17). For simplicity, we will provide the analysis for $q = O(1)$, i.e., it cannot depend on

n. However, λ_is also depend on n. To keep the dependence on n straightforward we can use the bounds $\frac{\lambda_i^*}{2} \le \lambda_i \le \frac{3\lambda_i^*}{2}$ (see (27)), which hold under the assumptions of Theorem 2. This helps because λ_i^*s are independent of n.

Theorem 2 allows δ to depend on n, but we will assume $\delta = O(1)$ for simplicity. We also have $C_{K,q} = O(1)$ since we assume $q = O(1)$. Consequently $s = \Omega(\frac{\log^4 n}{\varepsilon^4})$ is optimal. Hence, following (16), we can write

$$\frac{T_{\mathsf{GD}}}{T_{\mathsf{nPGD}}} = \frac{\kappa_1}{\kappa_{s,q} + s/n} \approx \frac{\kappa_1}{\kappa_{s,q}} \approx \frac{\kappa_1}{\kappa_q(1+\varepsilon)^4} \approx \frac{\lambda_1^*}{3\lambda_q^*(1+\varepsilon)^4}$$

following a concentration argument similar to (14) that controls $|\lambda_i' - \lambda_i^*|$ under the assumptions on s, see (27). We omit the constant 3 in the denominator since it becomes arbitrarily close to 1 for large n.

5 PROOF OF MAIN RESULT

Recall the definitions of the operators $\mathcal{T}, \mathcal{K}, \mathcal{K}', \mathcal{P}_q$ and $\mathcal{P}_{s,q}$ in equations (4–10).

The proof proceeds in three key steps: (i) We start by stating (in Proposition 5) that $\|\mathcal{K}^{\frac{1}{2}} - \mathcal{K}'^{\frac{1}{2}}\|_{\mathsf{OP}}$ can be made arbitrarily small if s and n are large enough. This follows from a concentration result in [Rosasco et al., 2010] that we state for our case in Corollary 4. (ii) Next, in Proposition 6 we show why $\|\mathcal{P}_q^{\frac{1}{2}} f\|/\|\mathcal{P}_{s,q}^{\frac{1}{2}} f\|$ being close to 1 for all f is sufficient to prove the claim in Theorem 2. (iii) Finally we show that in fact $\|\mathcal{K}^{\frac{1}{2}} - \mathcal{K}'^{\frac{1}{2}}\|_{\mathsf{OP}}$ being small is sufficient for $\|\mathcal{P}_q^{\frac{1}{2}} f\|/\|\mathcal{P}_{s,q}^{\frac{1}{2}} f\|$ to be close to 1.

We first start by showing that X_s, just like X_n, is an i.i.d. sample from ρ, whereby we can apply the same concentration arguments for both.

Lemma 3. *Let $x_1, \dots, x_n$ be i.i.d. random variables drawn from ρ, a probability distribution over measurable space $(\mathbb{R}^d, \beta)$. Let $i_1, \dots, i_s \in [n]$ be random according to some joint probability distribution such that $i_1, \dots, i_s$ are all distinct almost surely, and such that $i_1, \dots, i_s$ is independent of all x_i. Then $x_{i_1}, \dots, x_{i_s}$ are i.i.d., each distributed as ρ.*

Proof. Let ρ^s denote the product measure on $(\mathbb{R}^d)^s$. Let $A \in (\mathbb{R}^d)^s$ and measurable. Conditioning on $i_1, \dots, i_s$:

$$\begin{aligned}
&\mathbb{P}\big((x_{i_1}, \dots, x_{i_s}) \in A\big) \\
&= \mathbb{E}\big(\mathbb{P}((x_{i_1}, \dots, x_{i_s}) \in A \mid i_1, \dots, i_s)\big) \\
&\overset{(a)}{=} \mathbb{E}\big(\rho^s(A)\big) = \rho^s(A),
\end{aligned}$$

where (a) holds because $i_1, \dots, i_s$ are distinct and independent of $x_1, \dots, x_n$. □

We can now apply a concentration inequality to bound error between $\mathcal{K}$ and $\mathcal{T}$, and between $\mathcal{K}'$ and $\mathcal{T}$.

Corollary 4 (of [Rosasco et al., 2010, Theorem 7])**.** *With probability at least $1 - \frac{\delta}{2}$, with respect to the randomness of X_n, we have*

$$\|\mathcal{K} - \mathcal{T}\|_{\mathsf{OP}} \le 2\beta(K)\sqrt{\frac{2}{n}\log\left(\tfrac{4}{\delta}\right)}. \tag{18}$$

Similarly, with probability at least $1 - \frac{\delta}{2}$, with respect to the randomness of X_s, we have

$$\|\mathcal{K}' - \mathcal{T}\|_{\mathsf{OP}} \le 2\beta(K)\sqrt{\frac{2}{s}\log\left(\tfrac{4}{\delta}\right)}. \tag{19}$$

The result in [Rosasco et al., 2010, Theorem 7] was stated in terms of $\|\cdot\|_{\mathsf{HS}}$. The above corollary follows by using the fact that $\|\cdot\|_{\mathsf{OP}} \le \|\cdot\|_{\mathsf{HS}}$. Additionally, we substitute their τ for $\log\left(\frac{4}{\delta}\right)$ for the sake of readability.

Proposition 5. *With probability at least $1 - \delta$ (with respect to the randomness in X_s and X_n),*

$$\left\|\mathcal{K}^{\frac{1}{2}} - \mathcal{K}'^{\frac{1}{2}}\right\|_{\mathsf{OP}} \le 2\sqrt{\beta(K)\sqrt{\frac{2}{s}\log\left(\tfrac{4}{\delta}\right)}}.$$

Proof. Observe that we have the bound,

$$\left\|\mathcal{K}^{\frac{1}{2}} - \mathcal{K}'^{\frac{1}{2}}\right\|_{\mathsf{OP}} \le \sqrt{\|\mathcal{K} - \mathcal{K}'\|_{\mathsf{OP}}} \tag{20}$$

following [Bhatia, 2013, Theorem X.1, page 290]. Next, by triangle inequality, we get that with probability $1 - \delta$ (since we use a union bound on neither inequality failing, each with failure probability $\frac{\delta}{2}$),

$$\begin{aligned}
&\|\mathcal{K} - \mathcal{K}'\|_{\mathsf{OP}} \le \|\mathcal{T} - \mathcal{K}'\|_{\mathsf{OP}} + \|\mathcal{K} - \mathcal{T}\|_{\mathsf{OP}} \\
&\overset{(a)}{\le} 2\beta(K)\sqrt{2\log\left(\tfrac{4}{\delta}\right)}\left(\tfrac{1}{\sqrt{s}} + \tfrac{1}{\sqrt{n}}\right) \le 4\beta(K)\sqrt{\tfrac{2}{s}\log\left(\tfrac{4}{\delta}\right)}.
\end{aligned}$$

where (a) is because of Corollary 4 and the last inequality holds since $s \le n$. □

Recall definition of κ in equation (2).

Proposition 6. *Suppose there exists $\gamma > 0$ such that for all $f \in \mathcal{H}_K$ we have*

$$(1+\gamma)^{-1}\|\mathcal{P}_q^{\frac{1}{2}} f\| \le \|\mathcal{P}_{s,q}^{\frac{1}{2}} f\| \le (1+\gamma)\|\mathcal{P}_q^{\frac{1}{2}} f\|. \tag{21}$$

Then $\kappa(\mathcal{P}_{s,q}^{\frac{1}{2}} \mathcal{K} \mathcal{P}_{s,q}^{\frac{1}{2}}) \le (1+\gamma)^4 \kappa(\mathcal{P}_q \mathcal{K})$.

Proof. We start by establishing

$$\kappa(\mathcal{P}_{s,q}^{\frac{1}{2}} \mathcal{K} \mathcal{P}_{s,q}^{\frac{1}{2}}) = \kappa(\mathcal{P}_{s,q}^{\frac{1}{2}} \mathcal{K}^{\frac{1}{2}})^2. \tag{22}$$

This follows from the following observation: For a finite rank operator $\mathcal{A} \in L(\mathcal{H}_K)$ we have

$$\kappa(\mathcal{A}\mathcal{A}^*) = \kappa(\mathcal{A})^2 \tag{23}$$

(which can be seen by expressing $\kappa(\cdot)$ in terms of singular values as $\kappa(\mathcal{A}) = \sigma_1(\mathcal{A})/\sigma_{\mathrm{rank}(\mathcal{A})}(\mathcal{A})$ and standard matrix arguments, see discussion after Equation (2)). Then set $\mathcal{A} = \mathcal{P}_{s,q}^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}}$ in (23) to get (22).

A direct application of (21) and the observation that $\mathrm{Null}(\mathcal{P}_{s,q}^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}}) = \mathrm{Null}(\mathcal{P}_q^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}})$ (using the fact that $\mathcal{P}_{s,q}^{\frac{1}{2}}$ and $\mathcal{P}_q^{\frac{1}{2}}$ are invertible) in the definition of $\kappa(\cdot)$ (Equation (2)) gives

$$\kappa(\mathcal{P}_{s,q}^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}}) \le (1+\gamma)^2\kappa(\mathcal{P}_q^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}}). \tag{24}$$

To conclude, notice that $\mathcal{P}_q^{\frac{1}{2}}$ and $\mathcal{K}^{\frac{1}{2}}$ commute. Thus, $\kappa(\mathcal{P}_q\mathcal{K}) = \kappa(\mathcal{P}_q^{\frac{1}{2}}\mathcal{P}_q^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}}) = \kappa(\mathcal{P}_q^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}}\mathcal{P}_q^{\frac{1}{2}}) = \kappa(\mathcal{P}_q^{\frac{1}{2}}\mathcal{K}^{\frac{1}{2}})^2$ (using (23) again). The claim follows from this and Equations (22) and (24). □

We are now ready to complete the proof of Theorem 2.

Proof. (of Theorem 2). In Section 6 we provide a general framework that bounds $\|\mathcal{P}_q^{\frac{1}{2}}f\|/\|\mathcal{P}_{s,q}^{\frac{1}{2}}f\|$ close to 1, via a bound on $\|\mathcal{K}-\mathcal{K}'\|_{\mathsf{OP}}$. In the special case, in Lemma 7 if we set $\mathcal{V}$ as $\mathcal{K}^{\frac{1}{2}}$, and $\mathcal{V}'$ as $\mathcal{K}'^{\frac{1}{2}}$, and the observation that $\mathcal{P}_q^{\frac{1}{2}} = \sqrt{\lambda_q}h_{\sqrt{\lambda_q}}(\mathcal{K}^{\frac{1}{2}})^{-1}$, and $\mathcal{P}_{s,q}^{\frac{1}{2}} = \sqrt{\lambda'_q}h_{\sqrt{\lambda'_q}}(\mathcal{K}'^{\frac{1}{2}})^{-1}$, we get the following,

$$(1+\zeta)^{-1}\|\mathcal{P}_q^{\frac{1}{2}}f\| \le \|\mathcal{P}_{s,q}^{\frac{1}{2}}f\| \le (1+\zeta)\,\|\mathcal{P}_q^{\frac{1}{2}}f\|, \tag{25}$$

$$\zeta := 2C\log(n+1)\cdot\frac{\|\mathcal{K}^{\frac{1}{2}}-\mathcal{K}'^{\frac{1}{2}}\|_{\mathsf{OP}}}{\sqrt{\lambda_q}}\left(1+\frac{\|\mathcal{K}^{\frac{1}{2}}\|_{\mathsf{OP}}}{\sqrt{\lambda'_q}}\right) \tag{26}$$

for some universal constant C. Note that we used the fact that $C\log(\mathrm{rank}(\mathcal{K})+\mathrm{rank}(\mathcal{K}')+1) \le C\log(2n+1) \le 2C\log(n+1)$.

Observe that Proposition 6 says if we can find a suitable upper bound for ζ, we are done. Indeed, we will show that choosing s as in equation (17) leads to $\zeta \le \varepsilon$.

First, we assert that for sufficiently large values of n and s, we can eliminate the dependence on samples and relate everything to the operator $\mathcal{T}$. Note that by Weyl's inequality we have $\left|\lambda_q - \lambda_q^*\right| \le \|\mathcal{K}-\mathcal{T}\|_{\mathsf{OP}}$ (and similarly for λ'_q). Hence via equation (18) we have,

$$\lambda_q \ge \lambda_q^* - \|\mathcal{K}-\mathcal{T}\|_{\mathsf{OP}} \ge \lambda_q^* - 2\beta(K)\sqrt{\tfrac{2}{n}\log\left(\tfrac{4}{\delta}\right)}.$$

and similarly via equation (19) we have, $\lambda'_q \ge \lambda_q^* - 2\beta(K)\sqrt{\frac{2}{s}\log\left(\frac{4}{\delta}\right)}$. Using $n \ge s \ge c_1 C_{K,q}^2\log\left(\frac{4}{\delta}\right)$, where $c_1 := 32$ and $C_{K,q} = \beta(K)/\lambda_q^*$, we can write,

$$\frac{\lambda_q^*}{2} \le \lambda'_q \le \frac{3\lambda_q^*}{2}, \quad \text{and} \quad \frac{\lambda_q^*}{2} \le \lambda_q \le \frac{3\lambda_q^*}{2}. \tag{27}$$

which hold together with probability at least $1-\delta$, (via a union bound on the two events in Corollary 4).

We find an upper bound for ζ from (26) as following,

$$\begin{aligned}
\tfrac{\zeta}{2C\log(1+n)} &\overset{(a)}{\le} \sqrt{\frac{4\beta(K)}{\lambda_q}\sqrt{\frac{2}{s}\log\left(\tfrac{4}{\delta}\right)}}\left(1+\frac{\|\mathcal{K}^{\frac{1}{2}}\|_{\mathsf{OP}}}{\sqrt{\lambda'_q}}\right)\\
&\overset{(b)}{\le} \sqrt{\frac{4\beta(K)}{\lambda_q}\sqrt{\frac{2}{s}\log\left(\tfrac{4}{\delta}\right)}}\left(1+\sqrt{\frac{\beta(K)}{\lambda'_q}}\right)\\
&\overset{(c)}{\le} \sqrt{\frac{4\beta(K)}{\lambda_q}\sqrt{\frac{2}{s}\log\left(\tfrac{4}{\delta}\right)}}\left(1+\sqrt{2}\sqrt{\frac{\beta(K)}{\lambda_q^*}}\right)\\
&\overset{(d)}{\le} \sqrt{\frac{4\beta(K)}{\lambda_q}\sqrt{\frac{2}{s}\log\left(\tfrac{4}{\delta}\right)}}\left(2\sqrt{\frac{\beta(K)}{\lambda_q^*}}\right)\\
&\overset{(e)}{\le} \frac{4\sqrt{2}\beta(K)}{\lambda_q^*}\left(\frac{2\log\left(\frac{4}{\delta}\right)}{s}\right)^{\frac{1}{4}} \overset{(f)}{\le} \frac{\varepsilon}{2C\log(1+n)}
\end{aligned}$$

where (a) is due to Proposition 5, (c) and (e) apply inequalities (27), and (b) is justified by the following,

$$\|\mathcal{K}^{\frac{1}{2}}\|_{\mathsf{OP}} = \|\mathcal{K}\|_{\mathsf{OP}}^{\frac{1}{2}} \le \|\mathcal{K}\|_{\mathsf{HS}}^{\frac{1}{2}} \le \mathrm{trace}(\mathcal{K})^{\frac{1}{2}} \le \sqrt{\beta(K)},$$

where in we have used the monotonicity of the Schatten norm, and inequality (7). Similarly, inequality (d) holds because $\lambda_q^* \le \mathrm{trace}(\mathcal{T}) \le \beta(K)$, again following (7). Finally, (f) holds because for $c_2 := 2^{16}C^4$, we have assumed $s \ge c_2 C_{K,q}^4\frac{\log^4 n}{\varepsilon^4}\log\left(\frac{4}{\delta}\right)$, whereby we can conclude $\zeta \le \varepsilon$.

In conclusion, we have established that with probability at least $1-\delta$, under the assumptions in the statement of Theorem 2, equation (25) holds. Proposition 6 (with $\gamma \leftarrow \zeta$), along with $\zeta \le \varepsilon$, proves the claim. □

6 FROM ADDITIVE TO MULTIPLICATIVE APPROXIMATION

In this section, we show that when two finite rank operators $\mathcal{V}$ and $\mathcal{V}'$ are close in the operator norm, their corresponding preconditioners are similarly close *in a multiplicative sense.* The results in this section do not use properties of RKHS and hold for any separable Hilbert space $\mathcal{H}$.

We will prove the following general lemma.

Lemma 7. *Suppose $\mathcal{V}, \mathcal{V}' \in L(\mathcal{H})$ are finite rank, self-adjoint positive semidefinite operators. Eigenvalues of $\mathcal{V}$ are ordered as $\nu_1 \ge \nu_2 \ge \cdots$ and those of $\mathcal{V}'$ are $\nu'_1 \ge \nu'_2 \ge \cdots$. Further assume that $q \in \mathbb{N}$ is such that $\nu_q, \nu'_q > 0$. Define $\mathcal{C}, \mathcal{C}' \in L(\mathcal{H})$ given by $\mathcal{C} = \nu_q h_{\nu_q}(\mathcal{V})^{-1}$ and $\mathcal{C}' = \nu'_q h_{\nu'_q}(\mathcal{V}')^{-1}$. Then*

$$(\forall f \ne 0) \quad \frac{\|\mathcal{C}f\|}{\|\mathcal{C}'f\|} \in [(1+\varepsilon')^{-1}, 1+\varepsilon']$$

where $\varepsilon' = C\log\left(\operatorname{rank}(\mathcal{V}) + \operatorname{rank}(\mathcal{V}') + 1\right)\frac{\|\mathcal{V}-\mathcal{V}'\|_{\mathsf{OP}}}{\nu_q}(1+\frac{\|\mathcal{V}'\|_{\mathsf{OP}}}{\nu'_q})$ *for some universal constant* C.

The first step, Lemma 8, states that it suffices to find an upper bound for $\|\mathcal{V}-\mathcal{V}'\|_{\mathsf{OP}}$ and a lower bound for the eigenvalues of $\mathcal{C}$ and $\mathcal{C}'$.

Lemma 8. *Let* $\mathcal{A},\mathcal{B}\in L(\mathcal{H})$ *be such that for all* $f\in\mathcal{H}$, $\|\mathcal{A}f\|\geq\lambda\|f\|$ *and* $\|\mathcal{B}f\|\geq\lambda\|f\|$. *Additionally, assume* $\|\mathcal{A}-\mathcal{B}\|_{\mathsf{OP}}\leq\varepsilon$. *Then for all* $f\in\mathcal{H}$ *we have* $(1+\frac{\varepsilon}{\lambda})^{-1}\|\mathcal{B}f\|\leq\|\mathcal{A}f\|\leq(1+\frac{\varepsilon}{\lambda})\|\mathcal{B}f\|$.

Proof. By symmetry it is enough to prove the right hand side inequality: $\|\mathcal{A}f\| = \|\mathcal{A}f-\mathcal{B}f+\mathcal{B}f\| \leq \|\mathcal{B}f\|+\varepsilon\|f\|\leq\|\mathcal{B}f\|(1+\frac{\varepsilon}{\lambda})$. □

The next step, Lemma 9, is the observation that the quality of the multiplicative approximation in Lemma 8 is invariant under taking inverses of the operators. It turns out that it will be easier to bound $\left\|\mathcal{C}^{-1}-\mathcal{C}'^{-1}\right\|_{\mathsf{OP}}$ and apply Lemma 8 to $\mathcal{C}^{-1}$ and $\mathcal{C}'^{-1}$. As a bonus, in that case we can take $\lambda=1$.

Lemma 9. *Let* $\mathcal{A},\mathcal{B}\in L(\mathcal{H})$ *be self-adjoint and invertible. Let* $c\geq 1$. *The following statements are equivalent:*

1. $(\forall f\neq 0)\frac{\|\mathcal{A}f\|}{\|\mathcal{B}f\|}\in[c^{-1},c]$.
2. $\left\|\mathcal{A}\mathcal{B}^{-1}\right\|_{\mathsf{OP}},\left\|\mathcal{B}\mathcal{A}^{-1}\right\|_{\mathsf{OP}}\in[c^{-1},c]$.
3. $\left\|\mathcal{B}^{-1}\mathcal{A}\right\|_{\mathsf{OP}},\left\|\mathcal{A}^{-1}\mathcal{B}\right\|_{\mathsf{OP}}\in[c^{-1},c]$.
4. $(\forall f\neq 0)\frac{\left\|\mathcal{A}^{-1}f\right\|}{\left\|\mathcal{B}^{-1}f\right\|}\in[c^{-1},c]$.

Proof. (1 ⇔ 2) In 1, use substitution $f=\mathcal{B}^{-1}(g)$ to get $(\forall g\neq 0)\frac{\|\mathcal{A}\mathcal{B}^{-1}(g)\|}{\|g\|}\in[c^{-1},c]$. Taking sup we get $\left\|\mathcal{A}\mathcal{B}^{-1}\right\|_{\mathsf{OP}}\in[c^{-1},c]$. Similarly, in 1 use substitution $f=\mathcal{A}^{-1}g$ and take inf to get $\left\|\mathcal{B}^{-1}\mathcal{A}\right\|_{\mathsf{OP}}\in[c^{-1},c]$. The same argument in reverse shows the equivalence.

(2 ⇔ 3) Follows immediately from the fact that the operator norm is invariant under taking adjoint.

(3 ⇔ 4) This follows from 1 ⇔ 2 with $\mathcal{A}^{-1}$ in the role of $\mathcal{A}$ and $\mathcal{B}^{-1}$ in the role of $\mathcal{B}$. □

Proposition 10. *For any two operators* $\mathcal{S},\mathcal{T}\in L(\mathcal{H})$ *where* $\mathcal{H}$ *is an* k*-dimensional Hilbert space we have*

$$\|h_0(\mathcal{S})-h_0(\mathcal{T})\|_{\mathsf{OP}}\leq C\log(k+1)\|\mathcal{S}-\mathcal{T}\|_{\mathsf{OP}}.$$

where $C>0$ *is a universal constant.*

Proof. For an operator $\mathcal{S}$, denote by $|\mathcal{S}|$ the operator $(\mathcal{S}^*\mathcal{S})^{\frac{1}{2}}$. [Bhatia, 2010, Theorem 4.2] states that $\||\mathcal{S}|-|\mathcal{T}|\|_{\mathsf{OP}}\leq O(\log k)\|\mathcal{S}-\mathcal{T}\|_{\mathsf{OP}}\leq c\log(k+1)\|\mathcal{S}-\mathcal{T}\|_{\mathsf{OP}}$ for some universal constant $c>0$. Thus we have $h_0(\mathcal{S})=(\mathcal{S}+|\mathcal{S}|)/2$ whereby,

$$\begin{aligned}&\|h_0(\mathcal{S})-h_0(\mathcal{T})\|_{\mathsf{OP}} = \|(\mathcal{S}+|\mathcal{S}|)/2-(\mathcal{T}+|\mathcal{T}|)/2\|_{\mathsf{OP}}\\ &\leq\frac{1}{2}(\|\mathcal{S}-\mathcal{T}\|_{\mathsf{OP}}+\||\mathcal{S}|-|\mathcal{T}|\|_{\mathsf{OP}})\\ &\leq C\log(k+1)\|\mathcal{S}-\mathcal{T}\|_{\mathsf{OP}}.\end{aligned}$$ □

Instead of [Bhatia, 2010, Theorem 4.2], it may be possible to obtain a result similar to Proposition 10 by using [Kato, 1973, result I], which gives a bound where the dependence on k is replaced by a dependence on the operator norms of $\mathcal{S}$ and $\mathcal{T}$.

Proof. (of Lemma 7) Let $V=\operatorname{im}(\mathcal{V})+\operatorname{im}(\mathcal{V}')$, i.e., V is the subspaces generated by taking the linear combination of these two subspaces of $\mathcal{H}$. Below, if $\mathcal{A}\in L(\mathcal{H})$ is such that $\mathcal{A}|_V$ has image in V, then (by a slight abuse of notation) we consider $\mathcal{A}|_V$ as an element of $L(V)$ (even though in general $\mathcal{A}|_V: V\to\mathcal{H}$). We have

$$\begin{aligned}&\left\|\mathcal{C}^{-1}-\mathcal{C}'^{-1}\right\|_{\mathsf{OP}} = \left\|\nu_q^{-1}h_{\nu_q}(\mathcal{V})-\nu_q'^{-1}h_{\nu'_q}(\mathcal{V}')\right\|_{\mathsf{OP}}\\ &=\left\|\nu_q^{-1}h_{\nu_q}(\mathcal{V}|_V)-\nu_q'^{-1}h_{\nu'_q}(\mathcal{V}'|_V)\right\|_{\mathsf{OP}}\\ &=\left\|\nu_q^{-1}h_0\big((\mathcal{V}-\nu_q\mathcal{I})|_V\big)-\nu_q'^{-1}h_0\big((\mathcal{V}'-\nu'_q\mathcal{I})|_V\big)\right\|_{\mathsf{OP}}\\ &=\left\|h_0\big((\nu_q^{-1}\mathcal{V}-\mathcal{I})|_V\big)-h_0\big((\nu_q'^{-1}\mathcal{V}'-\mathcal{I})|_V\big)\right\|_{\mathsf{OP}}\\ &\overset{(a)}{\leq} C\log(\dim(V)+1)\left\|\nu_q^{-1}\mathcal{V}-\nu_q'^{-1}\mathcal{V}'\right\|_{\mathsf{OP}}\\ &\overset{(b)}{\leq} C\log(\dim(V)+1)\left(\left\|\frac{\mathcal{V}-\mathcal{V}'}{\nu_q}\right\|_{\mathsf{OP}}+\left\|\frac{\nu'_q-\nu_q}{\nu_q\nu'_q}\mathcal{V}'\right\|_{\mathsf{OP}}\right)\\ &\leq\tfrac{C}{\nu_q}\log(\dim(V)+1)\left(\|\mathcal{V}-\mathcal{V}'\|_{\mathsf{OP}}+\left|\nu'_q-\nu_q\right|\frac{\|\mathcal{V}'\|_{\mathsf{OP}}}{\nu'_q}\right)\\ &\leq\tfrac{C}{\nu_q}\log(\dim(V)+1)\|\mathcal{V}-\mathcal{V}'\|_{\mathsf{OP}}\left(1+\tfrac{1}{\nu'_q}\|\mathcal{V}'\|_{\mathsf{OP}}\right).\end{aligned}$$

Note that (a) holds because of Proposition 10, and (b) follows from a triangle inequality. From Lemma 8 (with $\lambda=1$) we get

$$(\forall f\neq 0)\quad\frac{\left\|\mathcal{C}^{-1}f\right\|}{\left\|\mathcal{C}'^{-1}f\right\|}\in[(1+\varepsilon')^{-1},1+\varepsilon'].$$

Lemma 9 and the observation that $\dim(V)\leq\operatorname{rank}(\mathcal{V})+\operatorname{rank}(\mathcal{V}')$ complete the proof. □

7 CONCLUSION

In this paper we analyzed the trade-offs of an approximation method for preconditioning used in fast algorithms for training kernel models. Our analysis provides sufficient conditions on the approximation properties of the approximated preconditioner. This analysis guides

the design of practical preconditioners using in implementations such as EigenPro2 [Ma and Belkin, 2019] and EigenPro3 [Abedsoltan et al., 2023] based on preconditioned gradient descent.

Acknowledgements

A.A. and M.B. are supported by the National Science Foundation (NSF) and the Simons Foundation for the Collaboration on the Theoretical Foundations of Deep Learning (https://deepfoundations.ai/) through awards DMS-2031883 and #814639 and the TILOS institute (NSF CCF-2112665). P.P. was supported by a Simons Postdoctoral Fellowship via HDSI at UCSD. L.R. is supported by the National Science Foundation under Grant CCF-2006994 and acknowledges support by HDSI, UCSD.

References

[Abedsoltan et al., 2023] Abedsoltan, A., Belkin, M., and Pandit, P. (2023). Toward large kernel models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

[Arora et al., 2019] Arora, S., Du, S. S., Li, Z., Salakhutdinov, R., Wang, R., and Yu, D. (2019). Harnessing the power of infinitely wide deep nets on small-data tasks. *arXiv preprint arXiv:1910.01663*.

[Avron et al., 2017] Avron, H., Clarkson, K. L., and Woodruff, D. P. (2017). Faster kernel ridge regression using sketching and preconditioning. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1116–1138.

[Beaglehole et al., 2023] Beaglehole, D., Radhakrishnan, A., Pandit, P., and Belkin, M. (2023). Mechanism of feature learning in convolutional neural networks. *arXiv preprint arXiv:2309.00570*.

[Bhatia, 2010] Bhatia, R. (2010). Modulus of continuity of the matrix absolute value. *Indian Journal of Pure and Applied Mathematics*, 41:99–111.

[Bhatia, 2013] Bhatia, R. (2013). *Matrix analysis*, volume 169. Springer Science & Business Media.

[Bietti and Bach, 2020] Bietti, A. and Bach, F. (2020). Deep equals shallow for relu networks in kernel regimes. *arXiv preprint arXiv:2009.14397*.

[Camoriano et al., 2016] Camoriano, R., Angles, T., Rudi, A., and Rosasco, L. (2016). Nytro: When subsampling meets early stopping. In *Artificial Intelligence and Statistics*, pages 1403–1411. PMLR.

[Charlier et al., 2021] Charlier, B., Feydy, J., Glaunes, J. A., Collin, F.-D., and Durif, G. (2021). Kernel operations on the gpu, with autodiff, without memory overflows. *The Journal of Machine Learning Research*, 22(1):3457–3462.

[Cutajar et al., 2016] Cutajar, K., Osborne, M., Cunningham, J., and Filippone, M. (2016). Preconditioning kernel matrices. In *International conference on machine learning*, pages 2529–2538. PMLR.

[Gardner et al., 2018] Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31.

[Garrigos and Gower, 2023] Garrigos, G. and Gower, R. M. (2023). Handbook of convergence theorems for (stochastic) gradient methods. *arXiv preprint arXiv:2301.11235*.

[Jacot et al., 2018] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

[Kato, 1973] Kato, T. (1973). Continuity of the map $S \to |S|$ for linear operators. *Proceedings of the Japan Academy*, 49(3):157 – 160.

[Ma and Belkin, 2017] Ma, S. and Belkin, M. (2017). Diving into the shallows: a computational perspective on large-scale shallow learning. *Advances in neural information processing systems*, 30.

[Ma and Belkin, 2019] Ma, S. and Belkin, M. (2019). Kernel machines that adapt to gpus for effective large batch training. *Proceedings of Machine Learning and Systems*, 1:360–373.

[Matthews et al., 2017] Matthews, A. G. d. G., Van Der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). Gpflow: A gaussian process library using tensorflow. *J. Mach. Learn. Res.*, 18(40):1–6.

[Meanti et al., 2020] Meanti, G., Carratino, L., Rosasco, L., and Rudi, A. (2020). Kernel methods through the roof: handling billions of points efficiently. *Advances in Neural Information Processing Systems*, 33:14410–14422.

[Radhakrishnan et al., 2022] Radhakrishnan, A., Beaglehole, D., Pandit, P., and Belkin, M. (2022). Feature learning in neural networks and kernel machines that recursively learn features. *arXiv preprint arXiv:2212.13881*.

[Rosasco et al., 2010] Rosasco, L., Belkin, M., and De Vito, E. (2010). On learning with integral operators. *Journal of Machine Learning Research*, 11(2).

[Rudi et al., 2017] Rudi, A., Carratino, L., and Rosasco, L. (2017). Falkon: An optimal large scale kernel method. *Advances in neural information processing systems*, 30.

[Shalev-Shwartz et al., 2007] Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007). Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning*, pages 807–814.

[Shankar et al., 2020] Shankar, V., Fang, A., Guo, W., Fridovich-Keil, S., Ragan-Kelley, J., Schmidt, L., and Recht, B. (2020). Neural kernels without tangents. In *International conference on machine learning*, pages 8614–8623. PMLR.

[Williams and Seeger, 2000] Williams, C. and Seeger, M. (2000). Using the nyström method to speed up kernel machines. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press.