DECENTRALIZED LEARNING FOR OVERPARAMETERIZED PROBLEMS: A MULTI-AGENT KERNEL APPROXIMATION APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

This work develops a novel framework for communication-efficient distributed learning where the models to be learnt are overparameterized. We focus on a class of kernel learning problems (which includes the popular neural tangent kernel (NTK) learning as a special case) and propose a novel *multi-agent kernel approximation* technique that allows the agents to distributedly estimate the full kernel function, and subsequently perform decentralized optimization, without directly exchanging any local data or parameters. The proposed framework is a significant departure from the classical consensus-based approaches, because the agents do not exchange problem parameters, and no consensus is required. We analyze the optimization and the generalization performance of the proposed framework for the ℓ_2 loss. We show that with M agents and N total samples, when certain generalized inner-product kernels (resp. the random features kernel) are used, each agent needs to communicate $\mathcal{O}\big(N^2/M\big)$ bits (resp. $\mathcal{O}\big(N\sqrt{N}/M\big)$ real values) to achieve minimax optimal generalization performance. We validate the theoretical results on 90 UCI benchmarking datasets (with average data size $N \approx 1000$) and show that each agent needs to share a total of $200N/M$ bits (resp. $3N/M$ real values) to closely match the performance of the centralized algorithms, and these numbers are independent of parameter and feature dimensions.

1 INTRODUCTION

Recently, decentralized optimization algorithms have become a mainstay of the optimization research. In decentralized optimization, multiple local agents hold small to moderately sized private datasets. Typically, the agents collaborate by iteratively solving their local problems while sharing some information with other agents. Most of the existing decentralized learning algorithms are deeply rooted in classical consensus-based approaches (Tsitsiklis, 1984), where the agents repetitively share the local model parameters with each other to reach an optimal *consensual* solution. However, the recent trend of using learning models with an extremely large number of parameters (He et al., 2016; Vaswani et al., 2017; Fedus et al., 2021) (i.e., the *overparameterized* regime (Belkin et al., 2019; Neal et al., 2018; Bartlett et al., 2020)) poses significant challenges to the above consensus-based paradigm, mainly because sharing model parameters iteratively becomes excessively expensive as the parameter dimension grows.

To reduce the communication overhead, there are two classical approaches: (i) sharing model parameters intermittently between computations (McMahan et al., 2017; Stich, 2018; Lin et al., 2018), and/or, (ii) compressing model parameters before sharing (Seide et al., 2014; Alistarh et al., 2017; Bernstein et al., 2018). Although these approaches can lead to some communication savings, they require iteratively communicating high-dimensional parameters or might lead to increased variance (McMahan et al., 2017; Alistarh et al., 2017), hence slowing down convergence. If the size of local data is much smaller than that of the parameters, perhaps a more efficient way is to directly share the local data. However this approach raises privacy concerns, thus almost never being used in practice. Therefore, a fundamental question in decentralized learning in the overparameterized regime is:

(Q) For overparameterized learning problems, how to design a decentralized algorithm, such that the agents achieve the *best* optimization/generalization performance while maintaining some level of privacy and only exchanging the *minimum* amount of messages?

In this work, we attempt to partially answer question **(Q)** in the context of distributed kernel learning (Vert et al., 2004). As a departure from the popular consensus-based algorithms, we propose an optimization framework that does not require the local agents to share model parameters or raw data. We focus on kernel learning because: (i) Kernel methods provide an elegant way to model non-linear learning problems with complex data dependencies as simple linear problems (Vert et al., 2004; Hofmann et al., 2008), and (ii) kernel-based methods can be used to capture the behavior of a fully-trained deep network with large width (Jacot et al., 2018; Arora et al., 2019; 2020).

Distributed implementation of kernel learning problems is challenging. Current state-of-the-art algorithms for kernel learning either rely on sharing raw data among agents and/or imposing restrictive assumptions on the number of agents (Zhang et al., 2015; Lin et al., 2017; Koppel et al., 2018; Lin et al., 2020; Hu et al., 2020; Pradhan et al., 2021; Predd et al., 2006). Some recent approaches rely on specific random feature (RF) kernels to alleviate some of the above problems. These algorithms reformulate the (approximate) problem in the parameter domain and solve it by iteratively sharing the (potentially high-dimensional) parameters (Bouboulis et al., 2017; Richards et al., 2020; Xu et al., 2020; Liu et al., 2021). These algorithms suffer from excessive communication overhead, especially in the overparameterized regime where the number of parameters is larger than the data size N. For example, implementing the neural tengent kernel (NTK) with RF kernel requires at least $\mathcal{O}(N^\beta)$, $\beta \geq 8$, random features (parameter dimension) using ReLU activation (Arora et al., 2019; Han et al., 2021). In Table 1, we compare the communication overhead required by each agent to achieve $\mathcal{O}(1/\sqrt{N})$ optimization accuracy[1] for popular distributed GD and SGD algorithms in the overparameterized regime. Note that even when $\beta \geq 2$, the overall communication can be extremely high.

Table 1: Number of (real) values communicated by each agent for optimizing a particular objective via Distributed GD and SGD in the overparameterized regime, i.e., parameter dimension $\mathcal{O}(N^\beta)$ with $\beta > 1$.

Objective	Algorithms	
	Distributed GD	Distributed SGD
Strongly-convex (PL)	$\mathcal{O}(N^\beta)$	$\mathcal{O}(N^{0.5+\beta}/M)$
Convex	$\mathcal{O}(N^{0.5+\beta})$	$\mathcal{O}(N^{1+\beta}/M)$
Non-convex	$\mathcal{O}(N^{0.5+\beta})$	$\mathcal{O}(N^{1+\beta}/M)$

In this work, we propose a novel algorithmic framework for decentralized kernel learning. Below, we list the major contributions of our work.

[GIP Kernel for Distributed Approximation] We define a new kernel suitable for distributed implementation, Generalized inner-product (GIP) kernel, that is fully characterized by the angle between a pair of feature vectors and their respective norms. Many kernels of practical importance including the NTK can be represented as GIP kernel. Further, we propose a *multi-agent kernel approximation* method for estimating the GIP and the popular RF kernels at individual agents.

[One-shot and Iterative Scheme] Based on the proposed kernel approximation, we develop two optimization algorithms, where the first one only needs *one-shot* information exchange, but requires sharing data labels among the agents; the second one needs *iterative* information exchange, but does not need to share the data labels. A key feature of these algorithms is that neither raw data nor (high-dimensional) parameters are exchanged among agents.

[Performance of the Approximation Framework] We theoretically analyze the optimization and the generalization performance of the proposed approximation algorithms for ℓ_2 loss. In particular, we show that GIP requires communicating $\mathcal{O}(N^2/M)$ bits and the RF kernel requires communicating $\mathcal{O}(N\sqrt{N}/M)$ real values per agent to achieve minimax optimal generalization performance. Note that in comparison to Table 1, the proposed algorithms require fewer communications, and importantly, the required communication is independent of the function class and the optimization algorithm. We validate the theoretical results by empirically showing the performance of our approximation algorithms on 90 UCI benchmarking datasets.

[1] To achieve optimal generalization performance it is sufficient to obtain an optimization accuracy of $\epsilon = \mathcal{O}(1/\sqrt{N})$ (Zhou et al., 2021).

Notations: We use $\mathbb{R}$, $\mathbb{R}^d$, and $\mathbb{R}^{n\times m}$ to denote the sets of real numbers, d-dimensional Euclidean space, and real matrices of size $n \times m$, respectively. We use $\mathbb{N}$ to denote the set of natural numbers. $\mathcal{N}(0, \Sigma)$ is multivariate normal distribution with zero mean and covariance Σ. Uniform distribution with support $[a, b]$ is denoted by $\mathcal{U}[a, b]$. $\langle a, b\rangle$ (resp. $\langle a, b\rangle_{\mathcal{H}}$) denotes the inner-product in Eucledean space (resp. Hilbert space $\mathcal{H}$). The inner-product defines the usual norms in corresponding spaces. Norm $\|A\|$ of matrix A denotes the operator norm induced by ℓ_2 vector norm. We denote by $[a]_i$ or $[a]^{(i)}$ the ith element of a vector a. $[A \cdot a]_j^{(i)}$ denotes the $(i \cdot j)^{\text{th}}$ element of vector $A \cdot a$. Moreover, $A^{(:,i)}$ is the ith column of A and $[A]_{mk}$ is the element corresponding to m^{th} row and k^{th} column. Notation $m \in [M]$ denotes $m \in \{1, .., M\}$. Finally, $\mathbb{1}[E]$ is the indicator function of event E.

2 Problem Statement

Given a probability distribution $\pi(x, y)$ over $\mathcal{X} \times \mathbb{R}$, we want to minimize the population loss

$$\mathcal{L}(f) = \mathbb{E}_{x,y\sim\pi(x,y)}[\ell(f(x), y)], \tag{1}$$

where $x \in \mathcal{X}$ and $y \in \mathbb{R}$ denote the features and the labels, respectively. Here, $f : \mathcal{X} \to \mathbb{R}$ is an estimate of the true label y. We consider a distributed system of M agents, with each agent $m \in [M]$ having access to a locally available independently and identically distributed (i.i.d) dataset $\mathcal{N}_m = \{x_m^{(i)}, y_m^{(i)}\}_{i=1}^n$ with[2] $(x_m^{(i)}, y_m^{(i)}) \sim \pi(x, y)$. The total number of samples is $N = nM$. The goal of kernel learning with a kernel function, $k(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, is to find a function $f \in \mathcal{H}$ (where $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) associated with k (Vert et al., 2004)) that minimizes (1). We aim to solve the following (decentralized) empirical risk minimization problem

$$\min_{f\in\mathcal{H}} \left\{ \hat{\mathcal{R}}(f) = \hat{\mathcal{L}}(f) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 = \frac{1}{M}\sum_{m=1}^{M} \hat{\mathcal{L}}_m(f) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \right\}, \tag{2}$$

where $\lambda > 0$ is the regularization parameter and $\hat{\mathcal{L}}_m(f) = \frac{1}{n}\sum_{i\in\mathcal{N}_m} \ell(f(x_m^{(i)}), y_m^{(i)})$ is the local loss for each $m \in [M]$. Problem (2) can be reformulated using the Representer theorem (Schölkopf et al., 2002) with $\hat{\mathcal{L}}_m(\alpha) = \frac{1}{n}\sum_{i\in\mathcal{N}_m} \ell\big([\mathbf{K}\alpha]_m^{(i)}, y_m^{(i)}\big)$, $\forall m \in [M]$, as

$$\min_{\alpha\in\mathbb{R}^N} \left\{ \hat{\mathcal{R}}(\alpha) = \hat{\mathcal{L}}(\alpha) + \frac{\lambda}{2}\|\alpha\|_{\mathbf{K}}^2 = \frac{1}{M}\sum_{m=1}^{M} \hat{\mathcal{L}}_m(\alpha) + \frac{\lambda}{2}\|\alpha\|_{\mathbf{K}}^2 \right\}, \tag{3}$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ is the kernel matrix with elements $k(x_m^{(i)}, x_{\bar{m}}^{(j)})$, $\forall m, \bar{m} \in [M]$, $\forall i \in \mathcal{N}_m$ and $\forall j \in \mathcal{N}_{\bar{m}}$; and $[\mathbf{K}\alpha]_m^{(i)}$ with $m \in [M]$ and $i \in \mathcal{N}_m$ represents the $(m \cdot i)^{\text{th}}$ element of vector $[\mathbf{K}\alpha] \in \mathbb{R}^N$. In contrast to (2), problem (3) is an N-dimensional problem.

The supervised (centralized) learning problem (3) is a classical problem in statistical learning (Caponnetto & De Vito, 2007) and has been popularized recently due to connections with overparameterized neural network training. Specifically, Jacot et al. (2018) showed that the behavior of a fully-connected deep neural network trained with gradient descent (GD) can be described by the NTK in the infinite width limit. Importantly, Arora et al. (2019; 2020) empirically showed that the NTK can match or even outperform the state-of-the-art kernel learning methods. However, it is by no means clear if and how one can efficiently solve (3) in a *decentralized manner*. The key challenge is that, unlike the conventional decentralized learning problems, here each loss term $\ell([\mathbf{K}\alpha]_m^{(i)}, y_m^{(i)})$ is not separable over the agents. Instead, each agent m's local problem is dependent on $k(x_m^{(i)}, x_{\bar{m}}^{(j)})$ with $m \neq \bar{m}$. Importantly, without directly transmitting the data itself (as has been done in Predd et al. (2006); Koppel et al. (2018); Lin et al. (2020)), it is not clear how one can obtain the required $(m \cdot i)^{\text{th}}$ element of $\mathbf{K}\alpha$. One option as adopted in Richards et al. (2020); Xu et al. (2020) is to solve (2) in the primal domain by approximating $k(\cdot, \cdot)$ with $k_P(x, x') = \langle \phi_P(x), \phi_P(x')\rangle$ via a finite dimensional random feature map $\phi_P(\cdot) \in \mathbb{R}^P$ such that $k \approx k_P$ (Rahimi et al., 2007; Rudi & Rosasco, 2017; Li et al., 2019). Since such approaches rely on sharing P-dimensional parameters at every iteration, the overall message exchange can still be extensive, especially in the overparameterized regime (cf. Table 1). Therefore, it is important to identify kernels that are suitable for decentralized implementation and propose efficient algorithms for learning with such kernels.

[2] The techniques presented in this work can be easily extended to unbalanced datasets, i.e., when each agent has a dataset of different size.

3 THE PROPOSED ALGORITHMS

In this section, we define a general class of kernels referred to as the *generalized inner product (GIP) kernels*. By focusing on GIP kernels, we aim to understand the best possible decentralized optimization/generalization performance that can be achieved for solving (3). Surprisingly, one of our proposed algorithm only shares $\mathcal{O}(nN) = \mathcal{O}(N^2/M)$ bits of information, while achieving the minimax optimal generalization performance. Such an algorithm only requires one round of communication, where the messages transmitted are independent of the actual parameter dimension; further, there is no requirement for achieving consensus among the agents. The proposed algorithm represents a significant departure from the *classical* consensus-based decentralized learning algorithms. We first define a class of kernels that we will focus on in this work.

Definition 3.1. [Generalized inner-product (GIP) kernel] We define a GIP kernel as:

$$k(x, x') = g(\psi(x, x'), \|x\|, \|x'\|), \tag{4}$$

where $\psi(x, x') = \arccos(x^T x'/(\|x\|\|x'\|)) \in [0, \pi]$ denotes the angle between the feature vectors x and x'; and $g(\cdot, \|x\|, \|x'\|)$ is assumed to be Lipschitz continuous (cf. Assumption 2).

Remark 1. Note that the GIP kernel is a generalization of the inner-product kernels (Schölkopf et al., 2002), i.e., kernels of the form $k(x, x') = k(\langle x, x'\rangle)$. Clearly, $k(\langle x, x'\rangle)$ can be represented as $k(\langle x, x'\rangle) = g(\psi(x, x'), \|x\|, \|x'\|)$ for some function $g(\cdot)$. Moreover, many kernels of practical interest can be represented as GIP kernels, some examples include NTK (Jacot et al., 2018; Chizat et al., 2019; Arora et al., 2019), arccosine (Cho & Saul, 2009), polynimal, Gaussian, Laplacian, sigmoid, and inner-product kernels (Schölkopf et al., 2002).

The main reason we focus on the GIP kernels for decentralized implementation is that, this class of kernels can be fully specified at each agent if the norms of all the feature vectors and the pairwise angles between them are known at each agent. For example, consider an NTK of a single hidden-layer ReLU neural network: $k(x, x') = x^T x'(\pi - \psi(x, x'))/2\pi$ (Chizat et al., 2019), this kernel can be fully learned with just the knowledge of norms and the pairwise angles of the feature vectors. For many applications of interest (Bietti & Mairal, 2019; Geifman et al., 2020; Pedregosa et al., 2011), normalized feature vectors are utilized for learning. For such problems the GIP kernel at each agent can be computed using the knowledge of the pairwise angles between the feature vectors. We show in Section 3.1 that such kernels can be efficiently estimated by each agent while sharing only a few bits of information. Next, we define the RF kernel that can be learned in a decentralized setting.

Definition 3.2. [Random features (RF) kernel] RF kernel is defined as (Rahimi et al., 2007; Rudi & Rosasco, 2017; Li et al., 2019) as:

$$k(x, x') = \int_{\omega \in \Omega} \zeta(x, \omega) \cdot \zeta(x', \omega) dp(\omega) \tag{5}$$

with (Ω, ω) being the probability space and $\zeta : \mathcal{X} \times \Omega \to \mathbb{R}$.

Remark 2. The RF kernel can be approximated as: $k(\cdot, \cdot) \approx k_P(x, x') = \langle \phi_P(x), \phi_P(x')\rangle$, with $\phi_P(x) = \frac{1}{\sqrt{P}}[\zeta(x, \omega_1), \cdots, \zeta(x, \omega_P)]^T \in \mathbb{R}^P$ and $\{\omega_i\}_{i=1}^P$ drawn i.i.d. from distribution $p(\omega)$. An important example of the RF kernels are the shift-invariant kernels, i.e., kernels of the form $k(x, x') = k(x - x')$ (Rahimi et al., 2007). The RF kernels generalize the random Fourier features construction (Rudin, 2017) for shift-invariant kernels to general kernels. Other than shift-invariant kernels, important examples of the RF kernels include, inner-product, (Kar & Karnick, 2012), homogeneous additive, (Vedaldi & Zisserman, 2012) and NTK (Jacot et al., 2018).

Next, we propose a multi-agent approximation algorithm to effectively learn the GIP and the RF kernels at each agent, as well as the optimization algorithms to efficiently solve the learning problem. Our proposed algorithms will follow an *approximation – optimization* strategy, where the agents first exchange some information so that they can locally *approximate* the full kernel matrix $\mathbf{K}$; then they can independently *optimize* the resulting approximated local problems. Below we list a number of key design issues arising from implementing such an approximation – optimization strategy:

[Kernel approximation] How to accurately approximate the kernel $\mathbf{K}$, locally at each agent? For example, for the class of GIP kernels, how to accurately construct the angles $\psi(x_m^{(i)}, x_{\bar{m}}^{(j)}) := \arccos(\langle x_m^{(i)}, x_{\bar{m}}^{(j)}\rangle/(\|x_m^{(i)}\|\|x_{\bar{m}}^{(j)}\|))$ for a given agent m, where $j \in \mathcal{N}_{\bar{m}}$ and $\bar{m} \neq m$? This is not an easy task, especially when direct local data sharing is not allowed.

Algorithm 1 Approximation: Local Kernel Estimation

1: **for** $m \in [M]$ **do**
2: Draw P i.i.d. random variables $\omega_i \in \mathbb{R}^d$ with $\omega_i \sim p(\omega)$ for $i = 1, \ldots, P$
3: Compute $\zeta(x_m^{(i)}, \omega_j)$ $\forall i \in \mathcal{N}_m$ and $j \in [P]$
4: Define matrix $A_m \in \mathbb{R}^{P \times n}$ with elements $\zeta(x_m^{(i)}, \omega_j)$
5: Communicate A_m to every other agent and receive $A_{\bar{m}}$ with $\bar{m} \neq m$ from other agents
6: Estimate the kernel matrix $\mathbf{K}_P$ locally using (6) for the GIP and (8) for the RF kernel
7: **end for**

[Effective exchange of local information] How shall we design appropriate messages to be exchanged among the agents? Obviously, the type of messages that gets exchanged will dependent on the underlying kernel approximation schemes. Therefore, it is critical that proposed approximation methods are able to utilize as little information from other agents as possible.

[Iterative or one-shot scheme] It is not clear if such an *approximation – optimization* scheme should be *one shot* or *iterative* – that is, whether it is more beneficial that the agents *iteratively* exchange information and perform local optimization (just as what the classical consensus based algorithms do), or they should do it just once. Once again, this will depend on the underlying information sharing schemes.

Next, we will formally introduce the proposed algorithms. Our presentation follows the *approximation – optimization* strategy outlined above. We first discuss the proposed decentralized kernel approximation algorithm, followed by two different ways of performing decentralized optimization.

3.1 Multi-agent kernel approximation

The kernel $\mathbf{K}$ is approximated locally at each agent using Algorithm 1. Note that in Step 1 of Algorithm 1, each agent randomly samples $\{\omega_i\}_{i=1}^P$ from $p(\omega)$ (cf. (5)). This can be easily established via random seed sharing as in Xu et al. (2020); Richards et al. (2020). In Step 3, each agent shares a locally constructed matrix A_m of size $P \times n$ with elements $\zeta(x_m^{(i)}, \omega_i)$ with all other agents. The choices of distribution $p(\omega)$ and $\zeta(\cdot, \cdot)$ depend on the choice of kernel. Specifically, we have:

[Approximation for GIP kernel] For the GIP kernel, we first assume that the feature vectors are normalized (Pedregosa et al., 2011). We then choose $p(\omega)$ to be any circularly symmetric distribution, for simplicity we choose $p(\omega)$ as $\mathcal{N}(0, I_d)$. Moreover, we use $\zeta(x, \omega) = \mathbb{1}[\omega^T x \geq 0]$ such that A_m is a binary matrix with entries $\{0, 1\}$. Note that such matrices are easy to communicate. Next, we approximate the kernel $\mathbf{K}$ with $\mathbf{K}_P$ as

$$k(x_m^{(i)}, x_{\bar{m}}^{(j)}) \approx k_P(x_m^{(i)}, x_{\bar{m}}^{(j)}) = g(\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)}), \|x_m^{(i)}\|, \|x_{\bar{m}}^{(j)}\|), \tag{6}$$

where $k(x_m^{(i)}, x_{\bar{m}}^{(j)})$ and $k_P(x_m^{(i)}, x_{\bar{m}}^{(j)})$ $\forall i \in \mathcal{N}_m$, $\forall m \in [M]$ and $\forall j \in \mathcal{N}_{\bar{m}}$ and $\forall \bar{m} \in [M]$ are the individual elements of $\mathbf{K}$ and $\mathbf{K}_P$, resp., and $\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)})$ is an approximation of the angle $\psi(x_m^{(i)}, x_{\bar{m}}^{(j)})$ evaluated using A_m, $A_{\bar{m}}$ as

$$\psi(x_m^{(i)}, x_{\bar{m}}^{(j)}) \approx \psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)}) = \left| \pi - 2\pi [A_m^{(:,i)}]^T [A_{\bar{m}}^{(:,j)}]/P \right|, \tag{7}$$

where $A_m^{(:,i)} \in \mathbb{R}^P$ represents the ith column of A_m. This implies that $\mathbf{K}$ can be approximated for the GIP kernel by communicating only nP bits of information per agent. Note that in the general case if the feature vectors are not normalized, then (6) can be evaluated by communicating additional n real values of the norms of the feature vectors by each agent.

[Approximation for RF kernel] For the RF kernel, we approximate $\mathbf{K}$ with $\mathbf{K}_P$ as

$$k(x_m^{(i)}, x_{\bar{m}}^{(j)}) \approx k_P(x_m^{(i)}, x_{\bar{m}}^{(j)}) = \langle \phi_P(x_m^{(i)}), \phi_P(x_{\bar{m}}^{(j)}) \rangle, \tag{8}$$

where $k(x_m^{(i)}, x_{\bar{m}}^{(j)})$ and $k_P(x_m^{(i)}, x_{\bar{m}}^{(j)})$ are elements of $\mathbf{K}$ and $\mathbf{K}_P$, resp., $\phi_P(x_m^{(i)}) = 1/\sqrt{P}[A_m^{(:,i)}]$ and $\phi_P(x_{\bar{m}}^{(j)}) = 1/\sqrt{P}[A_{\bar{m}}^{(:,j)}]$. Note that $\mathbf{K}$ can be approximated for the RF kernel by sharing only nP real values per agent. As discussed earlier, the distribution $p(\omega)$ and the mapping $\zeta(\cdot, \cdot)$ depend

Algorithm 2 Optimization: One-Shot Communication for Kernel Learning

1: **Initialize:** $\alpha_m^1 \in \mathbb{R}^N$, step-sizes $\{\eta_m^t\}_{t=1}^{T_m}$ at each agent $m \in [M]$
2: **for** $m \in [M]$ **do**
3: Using Algorithm 1 construct $\mathbf{K}_P$
4: Communicate $\bar{y}_m = [y_m^{(1)}, \ldots, y_m^{(n)}]^T \in \mathbb{R}^n$
5: Using $\mathbf{K}_P$ and $\bar{y}_m$ construct $\hat{\mathcal{L}}_P(\alpha)$ (cf. (9)) locally using $\hat{\mathcal{L}}_{m,P}(\alpha)$
6: **Option I:** Solve (9) exactly at each agent
7: **Option II:** Solve (9) inexactly using GD at each agent
8: **for** $t = 1$ to T_m
9: `GD Update`: $\alpha_m^{t+1} = \alpha_m^t - \eta_m^t \nabla \hat{\mathcal{R}}_P(\alpha_m^t)$
10: **end for**
11: **end for**
12: **Return:** α_m^{T+1} for all $m \in [M]$

on the type of RF kernel used, for example, for shift-invariant kernels with random Fourier features, we can choose $\zeta(x, \omega) = \sqrt{2}\cos(\omega^T x + b)$ with $\omega \sim p(\omega)$ and $b \sim \mathcal{U}[0, 2\pi]$ (Rahimi et al., 2007).

Now that using Algorithm 1 we have approximated the kernel matrix at all the agents, we are ready to solve (3) approximately.

3.2 The decentralized optimization step

The approximated kernel regression problem (3) with $\mathbf{K}_P$ obtained using Algorithm 1, and local loss $\hat{\mathcal{L}}_{m,P}(\alpha) := \frac{1}{n}\sum_{i \in \mathcal{N}_m} \ell\big(\big[\mathbf{K}_P \alpha\big]_m^{(i)}, y_m^{(i)}\big)$ is

$$\min_{\alpha \in \mathbb{R}^N} \left\{ \hat{\mathcal{R}}_P(\alpha) = \hat{\mathcal{L}}_P(\alpha) + \frac{\lambda}{2}\|\alpha\|_{\mathbf{K}_P}^2 = \frac{1}{M}\sum_{m=1}^{M} \hat{\mathcal{L}}_{m,P}(\alpha) + \frac{\lambda}{2}\|\alpha\|_{\mathbf{K}_P}^2 \right\}. \tag{9}$$

Remark 3. For the approximate problem (9), we would ideally want $\mathbf{K}_P$ constructed using the multi-agent kernel approximation approach to be positive semi-definite (PSD), i.e., the kernel function $k_P(\cdot, \cdot)$ is a positive definite (PD) kernel. From the definition of the approximate RF kernel (8), it is easy to verify that it is PD. However, it is not clear if the approximated GIP kernel is PD. Certainly, for the GIP kernel we expect that as $P \to \infty$ we have $\mathbf{K}_P \to \mathbf{K}$, i.e., asymptotically $\mathbf{K}_P$ is PSD, since $\mathbf{K}$ is PSD. In the Appendix, we introduce a sufficient condition (Assumption 6) that ensures $\mathbf{K}_P$ to be PSD for the GIP kernel. In the following, for simplicity we assume $\mathbf{K}_P$ is PSD.

Decentralized optimization based on one-shot communication: In this setting, we share the information among all the agents in *one-shot*, using the gathered information each agent learns its corresponding minimizer. We assume that each agent can communicate with every other agent either in a decentralized manner (or via a central server) before initialization. This is a common assumption in distributed learning with RF kernels where the agents need to share random seeds before initialization to determine the approximate feature mapping (Richards et al., 2020; Xu et al., 2020). Here, consensus is not enforced as each agent can learn a *local* minimizer which has a good *global property*. The label information is also exchanged among all the agents.

In Algorithm 2, we list the steps of the algorithm. In Step 1, the agents learn $\mathbf{K}_P$ (the local estimate of the kernel matrix) using Algorithm 1. In Step 4, the agents share the labels $\bar{y}_m$ so that each agent can (approximately) reconstruct the loss $\hat{\mathcal{L}}(\alpha)$ (cf. (9)) locally. Then each agent can either choose **Option I** or **Option II** to solve (9). A few important properties of Algorithm 2 are:

[Communication] Each agent communicates in one-shot a total of $\mathcal{O}(nP) = \mathcal{O}(NP/M)$ bits for learning with the GIP kernel, and the same number of real values for the RF kernels. Importantly, in contrast to parameter sharing approaches (cf. Table 1) the required communication is independent of the local optimization algorithm used.

[No parameter sharing] Algorithm 2 does not require the agents to (iteratively) share the high-dimensional model parameters in the original space, nor the kernel space parameter α's (cf. (9)). This is highly desirable, especially for overparameterized problems where either or both of these spaces can have large dimensions.

[No consensus needed] Each agent executes Algorithm 2 independently to learn α_m, without needing to reach any kind of consensus. They are free to choose different initializations, step-sizes, and even regularizers (i.e., λ in (9)). In contrast to the classical learning, where algorithms are designed to guarantee consensus (Koppel et al., 2018; Richards et al., 2020; Xu et al., 2020), our algorithms allow each agent to learn a different function $\hat{f}_{m,P}(\cdot)$.

The proposed framework relies on sharing random matrices A_m's that are random functions of the local features. In the future, it will be interesting to analyze the tradeoff between the optimization performance and privacy leakage (using e.g., notions from differential privacy (Huang et al., 2015)). Note that problem (9) can also be solved by utilizing an iterative distributed gradient tracking algorithm (Qu & Li, 2018), with the benefit that no label sharing is needed; see Appendix C for details.

Remark 4 (Optimization performance). Note that using Algorithm 2, we can solve the approximate problem (9) to arbitrary accuracy using either **Option I** or **Option II**. However, it is by no means clear if the solution obtained by Algorithm 2 will be close to the solution of (3). Therefore, after problem (9) is solved, it is important to understand how close the solutions returned by Algorithm 2 are to the original kernel regression problem (3).

4 Main Results

In this section, we analyze the performance of Algorithm 2. Specifically, we are interested in understanding the training loss and the generalization error incurred due to the kernel approximation (cf. Algorithm 1). For this purpose, we focus on ℓ_2 loss functions for which the kernel regression problem (9) can be solved in closed-form. Specifically, we want to minimize the loss:

$$\mathcal{L}(f) = \frac{1}{2}\mathbb{E}_{x,y\sim\pi(x,y)}[(f(x)-y)^2]. \tag{10}$$

Then we solve the following kernel ridge regression problem with the choice $\hat{\mathcal{L}}(\alpha) = \frac{1}{2N}\|\bar{y}-\mathbf{K}\alpha\|^2$. More specifically, problem (3) becomes:

$$\min_{\alpha\in\mathbb{R}^N}\left\{\hat{\mathcal{R}}(\alpha) = \frac{1}{2N}\|\bar{y}-\mathbf{K}\alpha\|^2 + \frac{\lambda}{2}\|\alpha\|^2_{\mathbf{K}}\right\} \tag{11}$$

where we denote $\bar{y} = [\bar{y}_1^T,\ldots,\bar{y}_M^T]^T \in \mathbb{R}^N$ with $\bar{y}_m = [y_m^{(1)}, y_m^{(2)},\ldots,y_m^{(n)}]^T \in \mathbb{R}^n$. The above problem can be solved in closed form with $\hat{\alpha}^* = [\mathbf{K}+N\cdot\lambda\cdot I]^{-1}\bar{y}$. Note that, problem (11) can be (approximately) solved using Algorithm 2, **Option I** in a distributed setting. The approximated problem at each agent with the kernel $\mathbf{K}_P$ and with the loss function $\hat{\mathcal{L}}_P(\alpha) = \frac{1}{2N}\|\bar{y}-\mathbf{K}_P\alpha\|^2$ is

$$\min_{\alpha\in\mathbb{R}^N}\left\{\hat{\mathcal{R}}_P(\alpha) = \frac{1}{2N}\|\bar{y}-\mathbf{K}_P\alpha\|^2 + \frac{\lambda}{2}\|\alpha\|^2_{\mathbf{K}_P}\right\} \tag{12}$$

with the optimal solution returned by **Option I** in Algorithm 2 as $\hat{\alpha}_P^* = [\mathbf{K}_P+N\cdot\lambda\cdot I]^{-1}\bar{y}$. The goal of this section is to analyze the impact of the approximation on the performance of Algorithm 2. Specifically, we bound the difference between the optimal losses of the exact and the approximated Kernel ridge regression. We begin with some assumptions.

Assumption 1. We assume $|k(x,x')| \le \kappa^2$ and $|k_P(x,x')| \le \kappa^2$ for some $\kappa \ge 1$.

Assumption 2. The function $g(\cdot)$ in (4) used to construct the GIP kernel is G-Lipschitz w.r.t. ψ, i.e., $\exists G \ge 0$ such that: $|g(\psi,z_2,z_3) - g(\hat{\psi},z_2,z_3)| \le G|\psi-\hat{\psi}|, \forall\psi,\hat{\psi}\in[0,\pi]$ and $\forall z_1,z_2\in\mathbb{R}$.

Assumption 3. We assume that the data labels $|y| \le R$ almost surely for some $R > 0$.

Assumption 4. There exists $f_{\mathcal{H}} \in \mathcal{H}$ such that $\mathcal{L}(f_{\mathcal{H}}) = \inf_{h\in\mathcal{H}}\mathcal{L}(h)$.

A few remarks are in order. Note that Assumptions 1, 3 and 4 are standard in the statistical learning theory (Cucker & Zhou, 2007; Caponnetto & De Vito, 2007; Ben-Hur & Weston, 2010; Rudi & Rosasco, 2017). Moreover, for RF kernel Assumption 1 is automatically satisfied if $|\zeta(x,\omega)| \le \kappa$ almost surely (Rudi & Rosasco, 2017; Richards et al., 2020) (cf. (5) and (8)). Assumption 2 is required for estimating the kernel by approximating the pairwise angles between feature vectors. It is easy to verify that the popular kernels including, NTK (14), Arccosine, Gaussian and Polynomial kernels satisfy Assumption 2 with feature vectors belonging to a compact domain. Now we are ready to present the results.

First, we analyze how well Algorithm 1 approximates the exact kernel. Specifically, we are interested in the approximation error as a function of the number of random samples P. We have the following lemma.

Lemma 4.1 (Approximation error)**.** *For $\mathbf{K}_P$ returned by Algorithm 1, the following holds with probability at least $1-\delta$: (i) For the GIP kernel, $\|\mathbf{K}-\mathbf{K}_P\| \leq GN\Big(\sqrt{\frac{32\pi^2}{P}\log\frac{2N}{\delta}}+\frac{8\pi}{3P}\log\frac{2N}{\delta}\Big)$. (ii) Similarly, for the RF kernel, $\|\mathbf{K}-\mathbf{K}_P\| \leq \kappa^2 N\Big(\sqrt{\frac{8}{P}\log\frac{2N}{\delta}}+\frac{4}{3P}\log\frac{2N}{\delta}\Big)$.*

Note that as P increases $\mathbf{K}_P \to \mathbf{K}$, in particular, to achieve an approximation error of $\epsilon > 0$, we need $P = \mathcal{O}(\epsilon^{-2})$. Importantly, Lemma 4.1 plays a crucial role in analyzing the optimization performance of the kernel approximation approach. Next, we state the training loss incurred as a consequence of solving the approximate decentralized problem (12) in Algorithm 2 instead of (11).

Theorem 4.2 (Optimization performance)**.** *Suppose $P \geq \frac{2}{9}\log\frac{2N}{\delta}$, then for both the GIP and the RF kernels, the solution returned by Algorithm 2 (**Option I**) for solving (11) approximately (i.e, (12)), satisfies the following with probability at least $1-\delta$*

$$\big|\hat{\mathcal{L}}_P(\hat{\alpha}_P^*)-\hat{\mathcal{L}}(\hat{\alpha}^*)\big| = \mathcal{O}\Big(\sqrt{\tfrac{1}{P}\log\tfrac{2N}{\delta}}\Big) \quad \textit{and} \quad \big|\hat{\mathcal{R}}_P(\hat{\alpha}_P^*)-\hat{\mathcal{R}}(\hat{\alpha}^*)\big| \leq \mathcal{O}\Big(\sqrt{\tfrac{1}{P}\log\tfrac{2N}{\delta}}\Big).$$

Theorem 4.2 states that as P increases, the optimal training loss achieved by solving approximating solving (12) via Algorithm 2 (**Option I**) will approach the performance of the centralized system (11). Importantly, this behavior is consistent for both the GIP and the RF kernels. The results of Lemma 4.1 and Theorem 4.2 characterize the optimization and approximation performance of the algorithms on the fixed training samples. Of course, it is of interest to analyze how the proposed approximation algorithms will perform on unseen test data. Towards this end, it is essential to analyze the performance of the function $\hat{f}_P$ learned from solving (12) via Algorithm 2, on the population loss (10). We have the following result.

Theorem 4.3 (Generalization performance)**.** *Let us choose $\lambda = 1/\sqrt{N}$ and $N \geq \max\Big\{\frac{4}{3\|K\|^2}, 72\kappa^2\sqrt{N}\log\frac{32\kappa^2\sqrt{N}}{\delta}\Big\}$, also choose $P \geq \max\Big\{8, \frac{512\pi^2G^2}{\|K\|^2}, 288\pi^2G^2N\Big\}\log\frac{16}{\delta}$ for the GIP kernel and $P \geq \max\Big\{8\kappa^2, \frac{32\kappa^2}{\|K\|^2}, 72\kappa^2\sqrt{N}\Big\}\log\frac{128\kappa^2\sqrt{N}}{\delta}$ for the RF kernel. Then with probability at least $1-\delta$, we have for $\hat{f}_P$ returned from Algorithm 2 (**Option I**) for approximately solving (11) (i.e., (12)): $\mathcal{L}(\hat{f}_P) - \inf_{h\in\mathcal{H}}\mathcal{L}(h) = \mathcal{O}\big(1/\sqrt{N}\big)$.*

The proof of Theorem 4.3 utilizes a result similar to Lemma 4.1 but for integral operator defined using kernels $k(\cdot,\cdot)$ and $k_P(\cdot,\cdot)$. For details please see Appendix G. Theorem 4.3 states that with appropriate choice of λ (the regularization parameter), N (the number of overall samples), and P (the messages communicated per agent), the proposed algorithm achieves the minimax optimal generalization performance (Caponnetto & De Vito, 2007). The next corollary precisely states the total communication required per agent to achieve this optimal performance.

Corollary 1 (Communication requirements for the GIP and RF kernels)**.** *With the choice of parameters stated in Theorem 4.3, Algorithm 2 for approximately solving (11) (i.e., (12)) with the GIP kernel requires a total of $\mathcal{O}(N^2/M)$ bits and the RF kernel requires a total of $\mathcal{O}(N\sqrt{N}/M)$ real values of communication per agent to achieve minimax optimal generalization performance.*

5 Experiments

In this section we empirically analyze the performance of the proposed *approximation – optimization* approach for decentralized learning on the UCI datasets, which is popular for benchmarking kernel learning problems. We are interested in analyzing the total number of message exchanges required by Algorithm 1 such that the solution returned by Algorithm 2 matches the baseline centralized algorithms. We follow the setup of Fernández-Delgado et al. (2014); Arora et al. (2020) for modelling and hyper-parameter selection. The details of experiments are provided in Appendix A, including the dataset selection, hyper-parameter tuning, and training and test data split, etc. Our focus is to compare the performance of Algorithm 2 on three popular kernels, namely, NTK, Gaussian and the polynomial kernels. We consider a multi-class classification task with problem (3) modeled as kernel SVM with soft margin. We analyze the performance of Algorithm 2 for approximately solving the kernel SVM for the following choices of $P \in \{10^i : i \in [1,6]\}$ (cf. (9)).

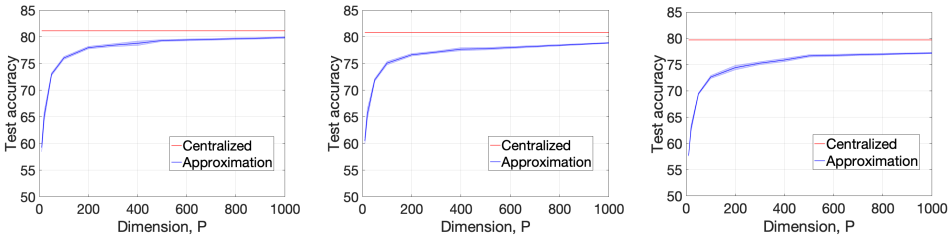
Table 2: Performance of Algorithm 2.

(a) Algorithm 2 vs centralized benchmark.

Algorithm	Polynomial	Gaussian	NTK
Centralized	79.69	80.79	81.13
Approx.	79.33 ± 0.005	80.36 ± 0.003	81.22 ± 0.006

(b) Communication per agent.

# Agents	Polynomial	Gaussian	NTK
5	900	500	440
10	450	250	220
20	225	125	110



(a) NTK. (b) Gaussian kernel. (c) Polynomial kernel.

Figure 1: Performance of Algorithm 2 with P on Kernel SVM for different kernels.

First in Table 2a, we compare the performance of the approximation – optimization scheme of Algorithm 2 to the centralized benchmark. Note from Table 2a that the test accuracy of Algorithm 2 for the classification task matches (and sometimes improves) that of the centralized benchmarks, corroborating the theoretical analysis of Section 4. In Figure 1, we show the effect of P on the test accuracy of the classification problem solved via Algorithm 2 (we only show performance for small P). Note that for all the (approximate) kernel methods, the accuracy increases rapidly in the beginning as P increases with $P \leq 200$ and approaches the centralized benchmarks as P increases further. Importantly, note that with only $P = 1000$, the test accuracy of Algorithm 2 for all the (approximated) kernels is within 2% of the centralized benchmarks. This implies that the proposed approximation – optimization approaches can be utilized to alleviate the communication burden for distributed learning algorithms. To analyze the communication performance of the proposed schemes, in Table 2b we compare the real values[3] shared by each agent to achieve an accuracy of 75%. For example, from Table 2b we note that for learning with approximate NTK, each agent needs to communicate only 220 real values on average for a network with 10 agents. This implies that a total of 2200 real values are shared across the network, which is about 2.5 times the total sample size ($N \approx 965$ on average). In contrast, if the agents share high-dimensional parameters (cf. Table 1) (resp. partial data), then the total real values shared will be 965^{β} for $\beta > 1$ per iteration (resp. $965 \cdot d_f$, where d_f is the dimension of the (shared) features), which will be much higher than our proposed approximation – optimization schemes even for moderate values of d_f.

6 Conclusion

In this work, we developed a novel framework for overparameterized decentralized learning in the context of kernel regression. For this purpose, we defined a general class of kernels, referred to as GIP kernels, that are suitable for distributed implementation. Specifically, we proposed a novel *multi-agent kernel approximation* approach for GIP and RF kernels, that allows the agents to distributedly estimate the full kernel matrix. A key feature of the proposed algorithms is that they allow the agents to perform decentralized optimization, without directly exchanging any local data or parameters. We theoretically analyzed the optimization and the generalization performance of the proposed framework for the ℓ_2 loss. Finally, we validated the theoretical results on 90 UCI benchmarking datasets and showed that the proposed approximation framework closely matches the performance of the centralized approaches while requiring only a small amount of message exchanges. Possible future directions of this work include analyzing and improving the privacy guarantees for the proposed approximation approach.

[3] We use a 64-bit architecture, i.e., communicating B bits is equivalent to communicating $B/64$ real values.

REFERENCES

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, 30:1709–1720, 2017.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Sanjeev Arora, Simon S. Du, Zhiyuan Li, Ruslan Salakhutdinov, Ruosong Wang, and Dingli Yu. Harnessing the power of infinitely wide deep nets on small-data tasks. In *International Conference on Learning Representations*, 2020.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Asa Ben-Hur and Jason Weston. A user's guide to support vector machines. In *Data mining techniques for the life sciences*, pp. 223–239. Springer, 2010.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pp. 560–569. PMLR, 2018.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Pantelis Bouboulis, Symeon Chouvardas, and Sergios Theodoridis. Online distributed learning over networks in rkh spaces using random fourier features. *IEEE Transactions on Signal Processing*, 66(7):1920–1932, 2017.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Youngmin Cho. *Kernel methods for deep learning*. University of California, San Diego, 2012.

Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

Felipe Cucker and Ding Xuan Zhou. *Learning theory: an approximation theory viewpoint*, volume 24. Cambridge University Press, 2007.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181, 2014.

Amnon Geifman, Abhay Yadav, Yoni Kasten, Meirav Galun, David Jacobs, and Basri Ronen. On the similarity between the laplace and neural tangent kernels. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1451–1461. Curran Associates, Inc., 2020.

Insu Han, Haim Avron, Neta Shoham, Chaewon Kim, and Jinwoo Shin. Random features for the neural tangent kernel. *arXiv preprint arXiv:2104.01351*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, 36(3):1171–1220, 2008.

Ting Hu, Qiang Wu, and Ding-Xuan Zhou. Distributed kernel gradient descent algorithm for minimum error entropy principle. *Applied and Computational Harmonic Analysis*, 49(1):229–256, 2020.

Zhenqi Huang, Sayan Mitra, and Nitin Vaidya. Differentially private distributed optimization. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, pp. 1–10, 2015.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. In *Artificial intelligence and statistics*, pp. 583–591. PMLR, 2012.

Alec Koppel, Santiago Paternain, Cédric Richard, and Alejandro Ribeiro. Decentralized online learning with kernels. *IEEE Transactions on Signal Processing*, 66(12):3240–3255, 2018.

Zhu Li, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards a unified analysis of random fourier features. In *International Conference on Machine Learning*, pp. 3905–3914. PMLR, 2019.

Shao-Bo Lin, Xin Guo, and Ding-Xuan Zhou. Distributed learning with regularized least squares. *The Journal of Machine Learning Research*, 18(1):3202–3232, 2017.

Shao-Bo Lin, Di Wang, and Ding-Xuan Zhou. Distributed kernel ridge regression with communications. *J. Mach. Learn. Res.*, 21:93–1, 2020.

Tao Lin, Sebastian U. Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd, 2018.

Yong Liu, Jiankun Liu, and Shuqiang Wang. Effective distributed learning with random features: Improved bounds and algorithms. In *International Conference on Learning Representations*, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Brady Neal, Sarthak Mittal, Aristide Baratin, Vinayak Tantia, Matthew Scicluna, Simon Lacoste-Julien, and Ioannis Mitliagkas. A modern take on the bias-variance tradeoff in neural networks. *arXiv preprint arXiv:1810.08591*, 2018.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Hrusikesha Pradhan, Amrit Singh Bedi, Alec Koppel, and Ketan Rajawat. Adaptive kernel learning in heterogeneous networks. *IEEE Transactions on Signal and Information Processing over Networks*, 2021.

Joel B Predd, Sanjeev R Kulkarni, and H Vincent Poor. Distributed kernel regression: An algorithm for training collaboratively. In *2006 IEEE Information Theory Workshop-ITW'06 Punta del Este*, pp. 332–336. IEEE, 2006.

Guannan Qu and Na Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2018. doi: 10.1109/TCNS.2017.2698261.

Ali Rahimi, Benjamin Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, pp. 5. Citeseer, 2007.

Dominic Richards, Patrick Rebeschini, and Lorenzo Rosasco. Decentralised learning with random features and distributed gradient descent. In *International Conference on Machine Learning*, pp. 8105–8115. PMLR, 2020.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *NIPS*, pp. 3215–3225, 2017.

Walter Rudin. *Fourier analysis on groups*. Courier Dover Publications, 2017.

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

Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Fifteenth Annual Conference of the International Speech Communication Association*. Citeseer, 2014.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Joel A Tropp. User-friendly tools for random matrices: An introduction. Technical report, California Institute Of Technology Pasadena Div Of Engineering and Applied Science, 2012.

John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):480–492, 2012.

Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. A primer on kernel methods. *Kernel methods in computational biology*, 47:35–70, 2004.

Ping Xu, Yue Wang, Xiang Chen, and Tian Zhi. Coke: Communication-censored kernel learning for decentralized non-parametric learning. *arXiv preprint arXiv:2001.10133*, 2020.

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

Pan Zhou, Xiaotong Yuan, Zhouchen Lin, and Steven Hoi. A hybrid stochastic-deterministic mini-batch proximal gradient method for efficient optimization and generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

APPENDIX

In this section, we discuss in detail some important aspects of the proposed kernel approximation framework along with the proofs of Theorems 4.2 and 4.3 in detail.

A ADDITIONAL EXPERIMENTAL DETAILS

The goal of this section is to empirically analyze the performance of the proposed multi-agent kernel approximation algorithm on real datasets. We are interested in the communication required by each agent (and network wide) to achieve performance similar to the baseline centralized algorithms. Specifically, we show that even with small amount of message sharing per agent, the proposed approximation framework performs well. To present the results we use three types of popular GIP kernels, namely, Polynomial, Gaussian and the NTK. We evaluate the performance of the approximation algorithms on the standard UCI dataset, a popular dataset for benchmarking kernel methods Fernández-Delgado et al. (2014). Below, we discuss the dataset and the implementation details.

Dataset and model: We use 90 UCI datasets for benchmarking the multi-agent kernel approximation algorithm against the centralized algorithms for multi-class classification tasks. For the datasets, we follow the pre-processing suggested in Fernández-Delgado et al. (2014). The datasets are chosen such that the sample sizes for each task are smaller than 5000 (Arora et al., 2020). There is an average of approximately 965 samples per task. The samples are randomly split into equal sized training and test sets, and typical 4-fold cross-validation is used for performance comparison (Pedregosa et al., 2011). Note that for testing we utilize Algorithm 4 in Appendix D. For communication comparison, we consider three settings where the data is equally split randomly among agents with number of agents chosen as 5, 10 and 20. Next, we discuss the algorithms and parameter setting.

Algorithms and parameter setting: We evaluate the performance of the poposed kernel-approximation approach on three popular kernels, namely Polynomial, Gaussian and the NTK. For multi-class classification, we utilize kernel SVM with soft margin and regularization hyper-parameter λ tuned from the set $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$. For all the kernels, the hyper-parameters are tuned using the validation method in Fernández-Delgado et al. (2014). For NTK, we choose the kernel corresponding to a single hidden-layer ReLU network. All the presented results are averaged over 3 independent runs of the algorithms.

B NEURAL TANGENT KERNEL

In this section, we discuss the NTK as a motivation to define the GIP kernels.

Neural tangent kernel (NTK): The NTK of an L hidden-layer fully-connected neural network with element wise activation $a : \mathbb{R} \to \mathbb{R}$ is Jacot et al. (2018); Chizat et al. (2019); Arora et al. (2019):

$$k(x, x') = \sum_{h=1}^{L+1} \left(\Sigma^{(h-1)}(x, x') \cdot \prod_{h'=h}^{L+1} \dot{\Sigma}^{(h')}(x, x') \right),$$

where $\Sigma^{(h)}$ and $\dot{\Sigma}^{(h)}$ are defined recursively as:

$$\begin{aligned}
&\Sigma^{(0)}(x, x') = x^T x', \\
&\Lambda^{(h)}(x, x') = \begin{pmatrix} \Sigma^{(h-1)}(x, x) & \Sigma^{(h-1)}(x, x') \\ \Sigma^{(h-1)}(x', x) & \Sigma^{(h-1)}(x', x') \end{pmatrix} \in \mathbb{R}^2, \\
&\Sigma^{(h)}(x, x') = \mathbb{E}_{(u,v)\in\mathcal{N}(0,\Lambda^{(h)})}[a(u)a(v)] \quad \text{and} \quad \dot{\Sigma}^{(h)}(x, x') = \mathbb{E}_{(u,v)\in\mathcal{N}(0,\Lambda^{(h)})}[\dot{a}(u)\dot{a}(v)],
\end{aligned} \tag{13}$$

where $\dot{a}(\cdot)$ is the derivative of a. For ReLU activation the NTK in (13) is a GIP kernel (Han et al., 2021) and can be defined in closed form (Arora et al., 2020; Han et al., 2021). For ease of exposition, we consider the NTK with ReLU activation for a single hidden-layer neural network (Cho & Saul, 2009; Cho, 2012; Chizat et al., 2019)

$$k(x, x') = x^T x' \, \frac{\pi - \psi(x, x')}{2\pi}, \tag{14}$$

Algorithm 3 Decentralized Gradient Tracking for Kernel Learning

1: **Initialize:** $\alpha_m^1 = \alpha^1 \in \mathbb{R}^N$, $\Delta_m^1 = \nabla\hat{\mathcal{R}}_P(\alpha_m^1)$, step-sizes $\{\eta^t\}_{t=1}^T$ at each agent $m \in [M]$
2: **for** $m \in [M]$ **do**
3: Using Algorithm 1 construct $\mathbf{K}_P$
4: Construct local functions $\hat{\mathcal{L}}_{m,P}(\alpha)$ (cf. equation 9)
5: **for** $t = 1$ to T **do**
6: `Update`: $\alpha_m^{t+1} = \sum_{k\in\mathcal{NB}_m}[W]_{mk}\alpha_m^t - \eta^t\Delta_m^t$
7: `Gradient Tracking`: $\Delta_m^{t+1} = \sum_{k\in\mathcal{NB}_m}[W]_{mk}\Delta_k^t + \nabla\hat{\mathcal{R}}_P(\alpha_m^{t+1}) - \nabla\hat{\mathcal{R}}_P(\alpha_m^t)$
8: **end for**
9: **end for**
10: **Return:** α_m^{T+1} for all $m \in [M]$

where $\psi(x, x') = \arccos(x^T x'/\|x\|\|x'\|) \in [0, \pi]$. Note that the NTK (14) depends on the inner-product as well as the angle between x and x', and the former can be constructed by the latter if one also knows the norms of the data samples. The main reason we focus on the GIP kernels for decentralized implementation is that, this class of kernels can be fully specified at each agent if the norms of all the feature vectors as well as the angles between any pair of them are known at each agent. Moreover, for many applications of practical interest (Bietti & Mairal, 2019; Geifman et al., 2020; Pedregosa et al., 2011) normalized feature vectors are utilized for learning, for such problems the GIP kernel at each agent can be fully computed only with the knowledge of the pairwise angles between the (normalized) feature vectors. Further, the RF kernels also exhibit a special structure where they can be approximated with the knowledge of feature mapping $\phi_P(\cdot) \in \mathbb{R}^P$ corresponding to all the feature vectors. The RF kernel approximation is expected to work well for generic kernels (Rahimi et al., 2007; Rudi & Rosasco, 2017), however, they might not be suited for all the problems, especially for overparameterized problems (like NTK in (14)), where P can be very large , i.e., $P = \mathcal{O}(N^\beta)$ with $\beta > 1$ (Han et al., 2021; Arora et al., 2019) (cf. Table 1 and discussion in Section 1). For such overparameterized problems, GIP kernel is an appealing alternative to the RF kernel. The above discussed properties of the GIP (and the RF) kernels enable us to design an efficient decentralized kernel approximation and message exchange scheme, which allows the agents to estimate randomly approximated versions of these kernels, but without directly sharing the (raw) local data.

C AN ALGORITHM BASED ON DECENTRALIZED GRADIENT TRACKING (DGT)

Note that in this setting we utilize DGT to learn α locally. Since, this algorithm is a natural extension of DGT in the functional space, there will be consensus in parameters α_m learned at each agent $m \in [M]$ (Qu & Li, 2018). Note that in contrast to Algorithm 2, DGT does not require sharing labels. For DGT we assume that the agents communicate via a undirected graph represented by a doubly stochastic matrix W. We denote by $\mathcal{NB}_m$ the neighbourhood of agent $m \in [M]$, i.e., $\mathcal{NB}_m = \{k \in [M] : [W]_{mk} \neq 0\}$. The steps of DGT are listed in Algorithm 3. In Step 3, each agent locally computes $\mathbf{K}_P$. We utilize the fact that the once $\mathbf{K}_P$ is known, the loss in (9) is decomposable in $m \in [M]$. This implies that each agent can update the local parameters α_m using Steps 6 and 7. Below we list important properties of Algorithm 3:

[Communication] The total communication required at each agent is $\mathcal{O}(nP + NT)$ values (where the term nP are bits for the GIP and real values for the RF kernels). The DGT algorithm is useful when problem (9) is strongly-convex (or PL), in that case $T = \mathcal{O}(\log N)$ (Qu & Li, 2018) and total communication becomes $\tilde{\mathcal{O}}(nP + N)$. Note that even for non-convex losses $\hat{\mathcal{L}}_{m,P}(\cdot)$, the local objective at each agent can be made strongly-convex with large regularizer, λ.

[Consensus] DGT with appropriately chosen step-sizes $\{\eta^t\}_{t=1}^T$ will guarantee consensus of α_m's, this will ensure functional consensus, i.e., $\hat{f}_{m,P} \approx \hat{f}_{\bar{m},P}$ for $m \neq \bar{m}$ and $m, \bar{m} \in [M]$.

Algorithm 4 Local Kernel Estimation for Testing

1: **Initialize:** Test samples $\{x^{(i)}\}_{i=1}^{N_{\text{test}}}$; $\{\omega_i\}_{i=1}^{P}$ and matrices $\{A_m\}_{m=1}^{M}$ from Algorithm 1
2: Compute $\zeta(x^{(i)}, \omega_j)$ $\forall i \in [N_{\text{test}}]$ and $j \in [P]$
3: Define matrix $A_{\text{test}} \in \mathbb{R}^{P \times N_{\text{test}}}$ with elements $\zeta(x^{(i)}, \omega_j)$
4: Estimate the kernel matrix $\mathbf{K}_{P,\text{test}}$ using (15) for the GIP and (16) for the RF kernel

D TESTING ALGORITHM FOR DECENTRALIZED KERNEL REGRESSION

In this section, we outline how to perform testing once the kernel model has been trained to solve (9). Suppose that there is a set N_{test} test samples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N_{\text{test}}}$. The idea is to utilize the local information at each agent, $\{A_m\}_{m=1}^{M}$, combined with the new data, to construct the testing kernel matrix, based on which the testing process can be carried out.

Specifically, we use Algorithm 4 to construct the kernel matrix $\mathbf{K}_{P,\text{test}} \in \mathbb{R}^{N_{\text{test}} \times N}$ as

1. **GIP kernel:** For the GIP kernel, we consider the same setting as (6) and construct $\mathbf{K}_{P.\text{test}}$ as

$$k_P(x^{(i)}, x_m^{(j)}) = g(\psi_P(x^{(i)}, x_m^{(j)}), \|x^{(i)}\|, \|x_m^{(j)}\|), \tag{15}$$

   where $k_P(x^{(i)}, x_m^{(j)})$ $\forall i \in [N_{\text{test}}]$ and $\forall j \in \mathcal{N}_m$, $\forall m \in [M]$ are the individual elements of $\mathbf{K}_{P,\text{test}}$ and where $\psi_P(x^{(i)}$ is defined using A_{test} and A_m as:

$$\psi_P(x^{(i)}, x_m^{(j)}) = \left| \pi - 2\pi \frac{[A_{\text{test}}^{(:,i)}]^T [A_m^{(:,j)}]}{P} \right|,$$

   where $A^{(:,i)} \in \mathbb{R}^P$ represents the ith column of matrix A.

2. **RF kernel:** Similarly, for the RF kernel, we construct $\mathbf{K}_{P.\text{test}}$ as

$$k_P(x^{(i)}, x_m^{(j)}) = \langle \phi_P(x^{(i)}), \phi_P(x_m^{(j)})) \rangle, \tag{16}$$

   where again $k_P(x^{(i)}, x_m^{(j)})$ are the individual elements of $\mathbf{K}_{P,\text{test}}$, similar to (8), we have $\phi_P(x^{(i)}) = 1/\sqrt{P}[A_{\text{test}}^{(:,i)}]$ and $\phi_P(x_m^{(j)}) = 1/\sqrt{P}[A_m^{(:,j)}]$.

Once we have $\mathbf{K}_{P,\text{test}}$, then the testing is performed as:

$$\hat{f}_{P,m}(x^{(i)}) = \left[\mathbf{K}_{P,\text{test}} \cdot \alpha_m^{(T+1)}\right]^{(i)}, \tag{17}$$

where and $\left[\mathbf{K}_{P,\text{test}} \cdot \alpha_m^{(T+1)}\right]^{(i)}$ represents the ith element of the vector $[\mathbf{K}_{P,\text{test}} \cdot \alpha_m^{(T+1)}] \in \mathbb{R}^{N_{\text{test}}}$. Using (17) we can evaluate the test loss

$$\hat{\mathcal{L}}_{P,\text{test}}(\alpha_m^{T+1}) = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \ell\Big(\left[\mathbf{K}_{P,\text{test}} \cdot \alpha_m^{(T+1)}\right]^{(i)}, y^{(i)} \Big). \tag{18}$$

Next, we present the performance of the multi-agent kernel approximation framework proposed in this work. Note that the algorithms presented above work for arbitrary loss functions. However, in the sequel we focus on the ℓ_2 loss and provide training and generalization loss incurred because of the multi-agent kernel approximation.

Remark 5 (Optimization performance of GD (Nesterov, 2003) and DGT (Qu & Li, 2018)). Algorithms 2 and 3 can solve the approximated kernel problem (9) to arbitrary accuracy. In particular, for strongly-convex and smooth objectives, both GD and DGT converge at a linear rate. On the other hand, for convex and smooth objectives both GD and DGT converge sub-linearly with rate $\mathcal{O}(1/T)$.

E NOTATIONS

First, we define some notations below.

- Recall that the features and the labels are generated as $(x, y) \sim \pi(x, y)$ where $\pi(x, y)$ defines the joint probability distribution on $\mathcal{X} \times \mathbb{R}$.
- π_x denotes the marginal distribution of the features on $\mathcal{X}$.
- $\pi(y|x)$ denotes the conditional distribution of the label y given the feature x on $\mathbb{R}$.
- We define by $L_2(\mathcal{X}, \pi_x)$ as the space of square integrable functions with Lebesgue measure defined by π_x.
- The inner-product and the norm on space $L_2(\mathcal{X}, \pi_x)$ is defined by

$$\text{Inner-product: } \langle f, g\rangle_{\pi_x} = \int_{x\in\mathcal{X}} f(x)g(x)d\pi_x(x),$$
$$\text{Norm: } \|f\|_{\pi_x} = \left(\langle f, f\rangle_{\pi_x}\right)^{1/2}, \ \ \forall f, g \in L_2(X, \pi_x).$$

- We define the RKHS induced by mapping k and k_P as $\mathcal{H}$ and $\mathcal{H}_P$, respectively, as

$$\mathcal{H} = \overline{\text{span}\{k(x, \cdot) : x \in \mathcal{X}\}}, \qquad \text{and} \qquad \mathcal{H}_P = \overline{\text{span}\{k_P(x, \cdot) : x \in \mathcal{X}\}}.$$

  with inner-products defined as $\langle k(x, \cdot), k(x', \cdot)\rangle_{\mathcal{H}} = k(x, x')$ in $\mathcal{H}$ and $\langle k_P(x, \cdot), k_P(x', \cdot)\rangle_{\mathcal{H}_P} = k_P(x, x')$ in $\mathcal{H}_P$. We also note that $\mathcal{H}_P = \mathbb{R}^P$ for the RF kernels.

Next, we define some operators which shall be utilized in bounding the optimization and generalization losses.

- We define $K : L_2(\mathcal{X}, \pi_x) \to L_2(\mathcal{X}, \pi_x)$ and $K_P : L_2(\mathcal{X}, \pi_x) \to L_2(\mathcal{X}, \pi_x)$ as

$$(Kh)(x) = \int_{z\in\mathcal{X}} k(x, z)h(z)d\pi_z \qquad \text{and} \qquad (K_P h)(x) = \int_{z\in\mathcal{X}} k_P(x, z)h(z)d\pi_z.$$

- Next, we define $\Phi : \mathcal{H} \to L_2(\mathcal{X}, \pi_x)$ and $\Phi_P : \mathcal{H}_P \to L_2(\mathcal{X}, \pi_x)$ as

$$(\Phi\beta)(x) = \langle \phi(x), \beta\rangle \qquad \text{and} \qquad (\Phi_P\beta)(x) = \langle \phi_P(x), \beta\rangle$$

- Moreover, using the definitions of Φ and Φ_P we define the covariances $C : \mathcal{H} \to \mathcal{H}$ and $C_P : \mathcal{H}_P \to \mathcal{H}_P$ as

$$C = \Phi^*\Phi \qquad \text{and} \qquad C_P = \Phi_P^*\Phi_P,$$

  Also, note that from the definition of Φ we have $K = \Phi\Phi^*$ and $K_P = \Phi_P\Phi_P^*$.
- We define the finite sample version of the operators Φ^* and Φ_P^* as $\hat{\Phi}^* : \mathbb{R}^N \to \mathcal{H}$ and $\hat{\Phi}_P^* : \mathbb{R}^N \to \mathcal{H}_P$, respectively as

$$\hat{\Phi}^*\hat{y} = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} \phi(x_i)\hat{y}_i \qquad \text{and} \qquad \hat{\Phi}_P^*\hat{y} = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} \phi_P(x_i)\hat{y}_i.$$

  where $\hat{y} \in \mathbb{R}^N$.
- From the definition of $\hat{\Phi}$ and $\hat{\Phi}_P$
  - We define $\hat{C} : \mathcal{H} \to \mathcal{H}$ and $\hat{C}_P : \mathcal{H}_P \to \mathcal{H}_P$ as

$$\hat{C} = \hat{\Phi}^*\hat{\Phi} \qquad \text{and} \qquad \hat{C}_P = \hat{\Phi}_P^*\hat{\Phi}_P.$$

  - Moreover, we define $\hat{K} : \mathbb{R}^N \to \mathbb{R}^N$ and $\hat{K}_P : \mathbb{R}^N \to \mathbb{R}^N$ as

$$\hat{K} = \hat{\Phi}\hat{\Phi}^* \qquad \text{and} \qquad \hat{K}_P = \hat{\Phi}_P\hat{\Phi}_P^*.$$

- For an operator A, we define the operator A_λ as $A_\lambda = A + \lambda I$. We also denote by $\lambda_{\max}(A)$ as the largest eigen value of A. We denote by $\|A\|$ the operator norm and $\|A\|_{HS}$ the Hilbert-Schmidt norm of a linear operator A.

Remark 6. Note from the definition of operators $\hat{K}$ and $\hat{K}_P$ we have

$$\hat{K} = \frac{1}{N}\mathbf{K}, \qquad \text{and} \qquad \hat{K}_P = \frac{1}{N}\mathbf{K}_P.$$

We will utilize this fact and the operators defined above in the proofs for bounding the optimization and generalization errors for the kernel methods.

F OPTIMIZATION PERFORMANCE OF MULTI-AGENT KERNEL APPROXIMATION FRAMEWORK: PROOF OF THEOREM 4.2

In this section, we present the complete proof of Theorem 4.2. Specifically, the main technical results in this section is to analyze the impact of the multi-agent kernel approximation framework proposed in this work on the training performance for the proposed algorithms. For this purpose, we bound difference between the optimal losses for the exact and the approximated kernel ridge regression problems (11) and (12). Towards this end, we first provide a few technical lemmas.

First, we present two concentration inequalities to bound the distance between the empirical mean of a sequence of random matrices (and random variables) from their mean. These results will be later used in Proposition F.4 to bound the operator norms of difference between the kernel matrix $\mathbf{K}$ and its approximation $\mathbf{K}_P$. First, we present the Bernstein's inequality for random matrices.

Proposition F.1 (Bernstein's inequality for random matrices, Theorem 6.1.1 in Tropp (2012))**.** *Consider a sequence X_ℓ for $\ell \in \mathbb{N}$ of independent, random, Hermitian matrices of dimension N. Assume that*

$$\mathbb{E}[X_\ell] = 0 \quad \textit{and} \quad \|X_\ell\| \leq C \quad \textit{almost surely}.$$

Define a random matrix as $Y := \sum_\ell X_\ell$, and denote its variance as $\sigma^2 := \|\mathbb{E}[Y^2]\|$. Then for all $\epsilon \geq 0$, we have

$$\mathbb{P}[\|Y\| \geq \epsilon] \leq 2N \exp\left(\frac{-\epsilon^2}{2(\sigma^2 + C \cdot \epsilon/3)}\right).$$

Moreover, with probability at least $1-\delta$ for $\delta \in (0,1]$ we have:

$$\|Y\| \leq \sqrt{2\sigma^2 \log \frac{2N}{\delta}} + \frac{2C}{3} \log \frac{2N}{\delta}.$$

In the next proposition, we present the Bernstein's inequality to bound the distance between the empirical mean of a sequence of random variables and their mean.

Proposition F.2 (Bernstein's inequality for random variables)**.** *Consider a sequence $\{x_\ell\}_{\ell \in \mathbb{N}}$ of independent, random variables such that the following holds:*

$$\mathbb{E}[x_\ell] = 0 \quad \textit{and} \quad |x_\ell| \leq C \quad \textit{almost surely}.$$

Define the random variable $y := \sum_\ell x_\ell$, and denote its variance as $\sigma^2 := \mathbb{E}[y^2]$. Then for all $\epsilon \geq 0$, we have:

$$\mathbb{P}[|y| \geq \epsilon] \leq 2 \exp\left(\frac{-\epsilon^2}{2(\sigma^2 + C \cdot \epsilon/3)}\right).$$

Next, using the above propositions we derive a union bound on the estimate $\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)})$ in (7) from its mean $\psi(x_m^{(i)}, x_{\bar{m}}^{(j)})$ for all training data samples, i.e., for all $i \in \mathcal{N}_m$ and $j \in \mathcal{N}_{\bar{m}}$ with $m, \bar{m} \in [M]$. Recall from Algorithm 1 and (7) that we have

$$\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)}) = \left|\pi - 2\pi \frac{[A_m^{(:,i)}]^T [A_{\bar{m}}^{(:,j)}]}{P}\right|, \tag{19}$$

where $A_m^{(:,i)} \in \mathbb{R}^P$. Specifically, the matrix A_m for the GIP kernel for the mth agent is

$$A_m = \begin{bmatrix} \zeta(x_m^{(1)}, \omega_1) & \zeta(x_m^{(2)}, \omega_1) & \cdots & \zeta(x_m^{(n)}, \omega_1) \\ \zeta(x_m^{(1)}, \omega_2) & \zeta(x_m^{(2)}, \omega_2) & \cdots & \zeta(x_m^{(n)}, \omega_2) \\ \vdots & \vdots & \ddots & \vdots \\ \zeta(x_m^{(1)}, \omega_P) & \zeta(x_m^{(2)}, \omega_P) & \cdots & \zeta(x_m^{(n)}, \omega_P) \end{bmatrix} \in \mathbb{R}^{P \times n},$$

where we choose $\zeta(x, \omega) = 1[\omega^T x \geq 0]$ and with $\{\omega_\ell\}_{\ell=1}^L$ chosen uniformly randomly from a circularly symmetric distribution. Please see discussion in Section 3.1 for details.

Below, we present the union bound.

Proposition F.3 (Union Bound: GIP kernel)**.** *Let* $\psi(x_m^{(i)}, x_{\bar{m}}^{(j)}) = \arccos\left(\frac{\langle x_m^{(i)}, x_{\bar{m}}^{(j)}\rangle}{\|x_m^{(i)}\|\|x_{\bar{m}}^{(j)}\|}\right)$ *be the angle between two feature vectors* $x_m^{(i)}$ *and* $x_{\bar{m}}^{(j)}$*, and let* $\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)})$ *be an estimate of* $\psi(x_m^{(i)}, x_{\bar{m}}^{(j)})$ *as defined in (19). Then with probability at least* $1-\delta$*, the following holds:*

$$\left|\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)}) - \psi(x_m^{(i)}, x_{\bar{m}}^{(j)})\right| \leq \sqrt{\frac{32\pi^2}{P}\log\frac{2N}{\delta}} + \frac{8\pi}{3P}\log\frac{2N}{\delta},$$

$\forall i \in \mathcal{N}_m$, $\forall j \in \mathcal{N}_{\bar{m}}$ *and* $\forall m, \bar{m} \in [M]$.

Proof. Note from the definition of $\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)})$ in (19) for the GIP kernel, we have

$$\begin{aligned}\left|\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)}) - \psi(x_m^{(i)}, x_{\bar{m}}^{(j)})\right| &\overset{(i)}{=} \left|\left|\pi - 2\pi\frac{[A_m^{(:,i)}]^T[A_{\bar{m}}^{(:,j)}]}{P}\right| - \left|\pi - 2\pi\mathbb{E}[\zeta(x_m^{(i)},\omega)\zeta(x_{\bar{m}}^{(j)},\omega)]\right|\right| \\ &\overset{(ii)}{\leq} 2\pi\left|\frac{[A_m^{(:,i)}]^T[A_{\bar{m}}^{(:,j)}]}{P} - \mathbb{E}[\zeta(x_m^{(i)},\omega)\zeta(x_{\bar{m}}^{(j)},\omega)]\right| \end{aligned} \tag{20}$$

with $\zeta(x,\omega) = 1[\omega^T x \geq 0]$ and where (i) uses $\mathbb{E}[\zeta(x_m^{(i)},\omega)\zeta(x_{\bar{m}}^{(j)},\omega)] = \frac{\pi - \psi(x_m^{(i)}, x_{\bar{m}}^{(j)})}{2\pi}$ for $\omega \sim \mathcal{N}(0, I)$ Chizat et al. (2019); Cho & Saul (2009); Cho (2012), and (ii) follows from the reverse triangle inequality. Next, note from the definition of A_m, we have

$$\frac{[A_m^{(:,i)}]^T[A_{\bar{m}}^{(:,j)}]}{P} = \frac{1}{P}\sum_{\ell=1}^{P}\zeta(x_m^{(i)},\omega_\ell)\zeta(x_{\bar{m}}^{(j)},\omega_\ell) \tag{21}$$

Let us define the random variable z_ℓ as

$$z_\ell := \frac{1}{P}\Big[\zeta(x_m^{(i)},\omega_\ell)\zeta(x_{\bar{m}}^{(j)},\omega_\ell) - \mathbb{E}\big[\zeta(x_m^{(i)},\omega)\zeta(x_{\bar{m}}^{(j)},\omega)\big]\Big].$$

Note from the definition of z_ℓ that we have

$$\mathbb{E}[z_\ell] = 0 \;\text{ and }\; |z_\ell| \leq \frac{1}{P}, \;\; \forall \ell \in [P].$$

To proceed, let us define the sum of z_ℓ's as $Z(x_m^{(i)}, x_{\bar{m}}^{(j)}) = \sum_{\ell=1}^{L} z_\ell$. Noticing that we can relate $Z(x_m^{(i)}, x_{\bar{m}}^{(j)})$ to the quantity (20) we want to bound in as

$$\begin{aligned} Z(x_m^{(i)}, x_{\bar{m}}^{(j)}) &= \sum_{\ell=1}^{P} z_\ell \\ &= \frac{1}{P}\sum_{\ell=1}^{P}\Big[\zeta(x_m^{(i)},\omega_\ell)\zeta(x_{\bar{m}}^{(j)},\omega_\ell) - \mathbb{E}\big[\zeta(x_m^{(i)},\omega)\zeta(x_{\bar{m}}^{(j)},\omega)\big]\Big] \\ &\overset{(i)}{=} \frac{[A_m^{(:,i)}]^T[A_{\bar{m}}^{(:,j)}]}{P} - \mathbb{E}\big[\zeta(x_m^{(i)},\omega)\zeta(x_{\bar{m}}^{(j)},\omega)\big]. \end{aligned}$$

where the equality (i) follows from (21). Moreover, note that this term is zero mean which follows from the fact that each z_ℓ is zero mean in the first equality. Further, we bound the variance of $Z(x_m^{(i)}, x_{\bar{m}}^{(j)})$ as

$$\sigma^2 = \mathbb{E}[(Z(x_m^{(i)}, x_{\bar{m}}^{(j)}))^2] = \mathbb{E}\Big[\Big(\sum_{\ell=1}^{P} z_\ell\Big)^2\Big] \overset{(i)}{=} \sum_{\ell=1}^{P}\mathbb{E}[z_\ell^2] \overset{(ii)}{\leq} \frac{1}{P},$$

where (i) comes from the facts that (a): each z_ℓ for $\ell \in [P]$ is zero mean, and (b): z_ℓ and z_k are independent for all $\ell \neq k$; (ii) uses the fact that $z_\ell^2 \leq 1/P^2$ for all $\ell \in [P]$.

Applying Proposition H.1 to the random variable $Z(x_m^{(i)}, x_{\bar{m}}^{(j)})$, we get

$$\mathbb{P}\left[\left|Z(x_m^{(i)}, x_{\bar{m}}^{(j)})\right| \geq \epsilon\right] \leq 2\exp\left(\frac{-\epsilon^2}{2\big((1/P) + (\epsilon/3P)\big)}\right).$$

Taking the union bound over all the training samples, i.e., $\forall i \in \mathcal{N}_m, \forall j \in \mathcal{N}_{\bar{m}}$ with $\forall m, \bar{m} \in [M]$, we get

$$\mathbb{P}\Big[\bigcup_{m,\bar{m}\in[M]}\bigcup_{i\in\mathcal{N}_m, j\in\mathcal{N}_{\bar{m}}}\Big\{\big|Z(x_m^{(i)}, x_{\bar{m}}^{(j)})\big| \geq \epsilon\Big\}\Big] \leq \sum_{m=1}^{M}\sum_{\bar{m}=1}^{M}\sum_{i\in\mathcal{N}_m}\sum_{j\in\mathcal{N}_{\bar{m}}} \mathbb{P}\Big[\big|Z(x_m^{(i)}, x_{\bar{m}}^{(j)})\big| \geq \epsilon\Big]$$
$$\leq 2N^2 \exp\left(\frac{-\epsilon^2}{2\big((1/P) + (\epsilon/3P)\big)}\right).$$

Note from above that we have that with probability at least $1-\delta$

$$|Z(x_m^{(i)}, x_{\bar{m}}^{(j)})| = \left|\frac{[A_m^{(:,i)}]^T[A_{\bar{m}}^{(:,j)}]}{P} - \mathbb{E}\big[\zeta(x_m^{(i)}, \omega)\zeta(x_{\bar{m}}^{(j)}, \omega)\big]\right|$$
$$\leq \sqrt{\frac{8}{P}\log\frac{2N}{\delta}} + \frac{4}{3P}\log\frac{2N}{\delta}, \quad \forall i \in \mathcal{N}_m, \forall j \in \mathcal{N}_{\bar{m}} \text{ with } \forall m, \bar{m} \in [M].$$

Substituting the above in (20) completes the proof.. □

Next, we provide a key technical result, which bounds the operator norm of the difference between the kernel, $\mathbf{K}$ and it's approximation $\mathbf{K}_P$. We present the results for both th GIP and the RF kernel. The following proposition is utilized in the proof of Theorem 4.2.

Proposition F.4. *The following holds with probability at least* $1-\delta$

1. *For GIP kernel, we have*
$$\|\mathbf{K} - \mathbf{K}_P\| \leq G \cdot N \cdot \left(\sqrt{\frac{32\pi^2}{P}\log\frac{2N}{\delta}} + \frac{8\pi}{3P}\log\frac{2N}{\delta}\right).$$
2. *For RF kernel, we have*
$$\|\mathbf{K} - \mathbf{K}_P\| \leq \kappa^2 \cdot N \cdot \left(\sqrt{\frac{8}{P}\log\frac{2N}{\delta}} + \frac{4}{3P}\log\frac{2N}{\delta}\right).$$

Proof. First, we consider the GIP kernel.

(i): GIP kernel: Note from the definition of approximated GIP kernel in (6) that the individual elements of $\mathbf{K}_P \in \mathbb{R}^{N\times N}$ are $g(\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)}), \|x_m^{(i)}\|, \|x_{\bar{m}}^{(j)}\|)$. Similarly, from (4) individual elements of $\mathbf{K} \in \mathbb{R}^{N\times N}$ are $g(\psi(x_m^{(i)}, x_{\bar{m}}^{(j)}), \|x_m^{(i)}\|, \|x_{\bar{m}}^{(j)}\|)$, $\forall i \in \mathcal{N}_m, \forall j \in \mathcal{N}_{\bar{m}}$ with $\forall m, \bar{m} \in [M]$. Next, we use the notation $B = \big[b_{ij}\big] \in \mathbb{R}^{N\times N}$ to denote the matrix with individual elements b_{ij}, i.e.,

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NN} \end{bmatrix} = \big[b_{ij}\big] \quad \forall i, j \in [N]. \tag{22}$$

Then we have with probability at least $1-\delta$

$$\|\mathbf{K} - \mathbf{K}_P\| \overset{(i)}{=} \left\|\Big[g(\psi(x_m^{(i)}, x_{\bar{m}}^{(j)}), \|x_m^{(i)}\|, \|x_{\bar{m}}^{(j)}\|) - g(\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)}), \|x_m^{(i)}\|, \|x_{\bar{m}}^{(j)}\|)\Big]\right\|$$
$$\overset{(ii)}{\leq} \left\|\Big[\Big|g(\psi(x_m^{(i)}, x_{\bar{m}}^{(j)}), \|x_m^{(i)}\|, \|x_{\bar{m}}^{(j)}\|) - g(\psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)}), \|x_m^{(i)}\|, \|x_{\bar{m}}^{(j)}\|)\Big|\Big]\right\|$$
$$\overset{(iii)}{\leq} G \cdot \left\|\Big[\Big|\psi(x_m^{(i)}, x_{\bar{m}}^{(j)}) - \psi_P(x_m^{(i)}, x_{\bar{m}}^{(j)})\Big|\Big]\right\|$$
$$\overset{(iv)}{\leq} G \cdot \left(\sqrt{\frac{32\pi^2}{P}\log\frac{2N}{\delta}} + \frac{8\pi}{3P}\log\frac{2N}{\delta}\right) \cdot \left\|\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}\right\|$$
$$\overset{(v)}{\leq} G \cdot N \cdot \left(\sqrt{\frac{32\pi^2}{P}\log\frac{2N}{\delta}} + \frac{8\pi}{3P}\log\frac{2N}{\delta}\right),$$

where equality (i) uses the definition of the GIP kernel and its approximation in (4) and (6), resp.; inequality (ii) results from the fact that for L_p-operator norm with p an even integer we have for any matrix B

$$\left\| \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NN} \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} |b_{11}| & |b_{12}| & \cdots & |b_{1N}| \\ \vdots & \vdots & \ddots & \vdots \\ |b_{N1}| & |b_{N2}| & \cdots & |b_{NN}| \end{bmatrix} \right\|,$$

inequality (iii) results from the Lipschitzness of function $g(\cdot, z_1, z_2)$ in Assumption 2 and the fact that for any matrices A and B with positive entries, we have $\|B\| \leq \|A + B\|$; inequality (iv) utilizes Proposition F.3 and again the fact that for any matrices A and B with positive entries, we have $\|B\| \leq \|A + B\|$; Finally, (v) uses the fact that the L_2-operator norm (maximum eigenvalue) of the all one matrix of dimension $N \times N$ is N.

Therefore, the first result has been proven.

(ii): RF kernel: First, we note that the matrix $\mathbf{K}_P$ can be decomposed as

$$\mathbf{K}_P = \frac{1}{P} \sum_{\ell=1}^{L} \mathbf{K}_\ell,$$

where $\mathbf{K}_\ell \in \mathbb{R}^{N \times N}$ consists of individual elements of the form $\zeta(x_m^{(i)}, \omega_\ell) \cdot \zeta(x_{\bar{m}}^{(j)}, \omega_\ell)$, $\forall i \in \mathcal{N}_m$, $\forall j \in \mathcal{N}_{\bar{m}}$ with $\forall m, \bar{m} \in [M]$. Recall that $\{\omega_\ell\}_{\ell=1}^P$ are drawn i.i.d from distribution $p(\omega)$. Please see discussion in Section 3.1 on RF kernel approximation. Then using the matrix notation defined in (22) we we write $\mathbf{K}_\ell$ as

$$\mathbf{K}_\ell := \left[\zeta(x_m^{(i)}, \omega_\ell) \cdot \zeta(x_{\bar{m}}^{(j)}, \omega_\ell) \right] \in \mathbb{R}^{N \times N} \quad \forall i \in \mathcal{N}_m, \forall j \in \mathcal{N}_{\bar{m}} \text{ with } \forall m, \bar{m} \in [M].$$

We apply Proposition F.1 on the sequence of random matrices defined by $[\mathbf{K}_\ell - \mathbf{K}]/P$. Note that we have:

$$\mathbb{E}\left[\frac{\mathbf{K}_\ell - \mathbf{K}}{P} \right] = 0 \qquad \text{and} \qquad \left\| \frac{\mathbf{K}_\ell - \mathbf{K}}{P} \right\| \leq \frac{2N \cdot \kappa^2}{P},$$

where the first equality follows from the definition of the RF kernel in (5) and the definition of $\mathbf{K}_\ell$, and the second inequality follows from Assumption 1. To apply Proposition F.1, we need to upper bound the variance of the sum $\sum_{\ell=1}^P \frac{\mathbf{K}_\ell - \mathbf{K}}{P} = \mathbf{K}_P - \mathbf{K}$. Towards this end, we obtain:

$$\begin{aligned}
\sigma^2\big(\mathbf{K}_P - \mathbf{K}\big) &:= \big\| \mathbb{E}[(\mathbf{K}_P - \mathbf{K})^2] \big\| \\
&\overset{(i)}{=} \frac{1}{P^2} \Big\| \sum_{\ell=1}^{L} \mathbb{E}[(\mathbf{K}_\ell - \mathbf{K})^2] \Big\| \\
&\overset{(ii)}{=} \frac{1}{P} \big\| \mathbb{E}[(\mathbf{K}_\ell - \mathbf{K})^2] \big\| \\
&\overset{(iii)}{\leq} \frac{1}{P} \Big\| \Big| \mathbb{E}[(\mathbf{K}_\ell - \mathbf{K})^2] \Big| \Big\| \\
&\overset{(iv)}{\leq} \frac{4\kappa^4 N}{P} \left\| \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \right\| \\
&\overset{(v)}{\leq} \frac{4\kappa^4 N^2}{P},
\end{aligned}$$

where (i) follows from the fact that $\mathbf{K}_\ell - \mathbf{K}$ are zero mean and independent across $\ell \in [P]$; (ii) utilizes that $\mathbf{K}_\ell - \mathbf{K}$ are i.i.d. across $\ell \in [P]$; for inequality (iii), we used the notation that for a matrix B, $|B|$ denotes the matrix with absolute values of B and combined it with the fact that for L_p-operator norm with p as an even integer we have for any matrix B that

$$\left\| \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NN} \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} |b_{11}| & |b_{12}| & \cdots & |b_{1N}| \\ \vdots & \vdots & \ddots & \vdots \\ |b_{N1}| & |b_{N2}| & \cdots & |b_{NN}| \end{bmatrix} \right\|.$$

Inequality, (iv) results from the use of Assumption 1, specifically, using Assumption 1 verify that each element of $\mathbb{E}[(\mathbf{K}_\ell - \mathbf{K})^2]$ is bounded by $4\kappa^4 N$ and combine with the fact that for any matrices A and B with positive entries, we have $\|B\| \leq \|A + B\|$; finally, (v) utilizes that for all one matrix of dimension $N \times N$ the L_2 operator norm (maximum eigen-value) is equal to N.

Therefore, we have the result. Next, using Proposition F.1 with $X_\ell = \frac{\mathbf{K}_\ell - \mathbf{K}}{P}$, we have

$$\|\mathbf{K}_P - \mathbf{K}\| \leq \sqrt{\frac{8\kappa^4 N^2}{P} \log \frac{2N}{\delta}} + \frac{4N\kappa^2}{3P} \log \frac{2N}{\delta} = \kappa^2 \cdot N \cdot \left(\sqrt{\frac{8}{P} \log \frac{2N}{\delta}} + \frac{4}{3P} \log \frac{2N}{\delta} \right).$$

with probability at least $1 - \delta$.

Therefore, the proof is complete. □

Finally, we state our main result.

Theorem F.5. *For any fixed $\lambda > 0$, the following holds with probability at least $1 - \delta$*

1. *For the GIP kernel, we have:*

$$\left| \hat{\mathcal{L}}_P(\hat{\alpha}_P^*) - \hat{\mathcal{L}}(\hat{\alpha}^*) \right| \leq \frac{2R^2 \cdot G}{\lambda} \cdot \left(\sqrt{\frac{32\pi^2}{P} \log \frac{2N}{\delta}} + \frac{8\pi}{3P} \log \frac{2N}{\delta} \right).$$

   Moreover, we have:

$$\left| \hat{\mathcal{R}}_P(\hat{\alpha}_P^*) - \hat{\mathcal{R}}(\hat{\alpha}^*) \right| \leq \frac{3R^2 \cdot G}{\lambda} \cdot \left(\sqrt{\frac{32\pi^2}{P} \log \frac{2N}{\delta}} + \frac{8\pi}{3P} \log \frac{2N}{\delta} \right).$$

2. *For the RF kernel, we have:*

$$\left| \hat{\mathcal{L}}_P(\hat{\alpha}_P^*) - \hat{\mathcal{L}}(\hat{\alpha}^*) \right| \leq \frac{2R^2 \cdot \kappa^2}{\lambda} \cdot \left(\sqrt{\frac{8}{P} \log \frac{2N}{\delta}} + \frac{4}{3P} \log \frac{2N}{\delta} \right).$$

   Moreover, we have

$$\left| \hat{\mathcal{R}}_P(\hat{\alpha}_P^*) - \hat{\mathcal{R}}(\hat{\alpha}^*) \right| \leq \frac{3R^2 \cdot \kappa^2}{\lambda} \cdot \left(\sqrt{\frac{8}{P} \log \frac{2N}{\delta}} + \frac{4}{3P} \log \frac{2N}{\delta} \right).$$

Proof. Recall from the notations in Appendix E that we have

$$\hat{K} = \frac{1}{N}\mathbf{K}, \qquad \text{and} \qquad \hat{K}_P = \frac{1}{N}\mathbf{K}_P. \tag{23}$$

and we defined

$$\hat{K}_\lambda = \hat{K} + \lambda \cdot I, \quad \hat{K}_{P,\lambda} = \hat{K}_P + \lambda \cdot I, \tag{24}$$

where I is the identity matrix of some appropriate size. It is easy to observe that the optimal solutions to problems (11) and (12) are given by the following closed-form solutions

$$\hat{\alpha}^* = [\mathbf{K} + N \cdot \lambda \cdot I]^{-1}\bar{y} = \frac{1}{N} \cdot \hat{K}_\lambda^{-1} \cdot \bar{y}, \quad \hat{\alpha}_P^* = [\mathbf{K}_P + N \cdot \lambda \cdot I]^{-1}\bar{y} = \frac{1}{N} \cdot \hat{K}_{P,\lambda}^{-1} \cdot \bar{y}. \tag{25}$$

From the definition of the loss function for the ℓ_2 loss, we obtain:

$$\begin{aligned}
\left| \hat{\mathcal{L}}_P(\hat{\alpha}_P^*) - \hat{\mathcal{L}}(\hat{\alpha}^*) \right| &\overset{(i)}{=} \frac{1}{2N} \Big| \, \|\bar{y} - \hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1} \cdot \bar{y}\|^2 - \|\bar{y} - \hat{K} \cdot \hat{K}_\lambda^{-1} \cdot \bar{y}\|^2 \, \Big| \\
&\overset{(ii)}{=} \frac{1}{2N} \Big| \left\langle \big[\hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1} - \hat{K} \cdot \hat{K}_\lambda^{-1}\big]\bar{y}, \big[\hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1} + \hat{K} \cdot \hat{K}_\lambda^{-1}\big]\bar{y} \right\rangle \\
&\qquad\qquad - 2\left\langle \bar{y}, \big[\hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1} - \hat{K} \cdot \hat{K}_\lambda^{-1}\big]\bar{y} \right\rangle \Big| \\
&\overset{(iii)}{\leq} \frac{1}{2N} \Bigg[\underbrace{\left\| \big[\hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1} + \hat{K} \cdot \hat{K}_\lambda^{-1}\big]\bar{y} \right\|}_{\text{term(A)}} \underbrace{\left\| \big[\hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1} - \hat{K} \cdot \hat{K}_\lambda^{-1}\big]\bar{y} \right\|}_{\text{term(B)}} \\
&\qquad\qquad + 2\|\bar{y}\| \underbrace{\left\| \big[\hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1} - \hat{K} \cdot \hat{K}_\lambda^{-1}\big]\bar{y} \right\|}_{\text{term(B)}} \Bigg]
\end{aligned} \tag{26}$$

where (i) utilizes (24) and (25); (ii) uses the following

$$\|A \cdot z - z\|^2 - \|B \cdot z - z\|^2 = \|A \cdot z\|^2 - \|B \cdot z\|^2 - 2\langle z, [A - B]z\rangle$$
$$= \langle [A + B]z, [A - B]z\rangle - 2\langle z, [A - B]z, \rangle,$$

and (iii) results from the application of Cauchy-Schwartz and triangle inequality.

Next, we consider the term(A) and term(B) in (26), and provide upper bounds for them. Let us first analyze term(B). we have:

$$\begin{aligned}
\big[\hat{K} \cdot \hat{K}_\lambda^{-1} - \hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1}\big]\bar{y} &= \big[\big(\hat{K} + \lambda \cdot I - \lambda \cdot I\big)\hat{K}_\lambda^{-1} - \big(\hat{K}_P + \lambda \cdot I - \lambda \cdot I\big)\hat{K}_{P,\lambda}^{-1}\big]\bar{y} \\
&\overset{(i)}{=} \big[\big(I - \lambda \cdot \hat{K}_\lambda^{-1}\big) - \big(I - \lambda \cdot \hat{K}_{P,\lambda}^{-1}\big)\big]\bar{y} \\
&= \lambda \cdot \big[\hat{K}_{P,\lambda}^{-1} - \hat{K}_\lambda^{-1}\big]\bar{y} \\
&\overset{(ii)}{=} \lambda \cdot \hat{K}_{P,\lambda}^{-1}\big[\hat{K}_\lambda - \hat{K}_{P,\lambda}\big]\hat{K}_\lambda^{-1} \cdot \bar{y} \\
&\overset{(iii)}{=} \lambda \cdot \hat{K}_{P,\lambda}^{-1}\big[\hat{K} - \hat{K}_P\big]\hat{K}_\lambda^{-1} \cdot \bar{y} \qquad (27)
\end{aligned}$$

where (i) follows from the definition of $\hat{K}_\lambda$ and $\hat{K}_{P,\lambda}$; (ii) results from $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$ for invertible matrices A and B and (iii) again results from the definition of $\hat{K}_\lambda$ and $\hat{K}_{P,\lambda}$. This implies that we have:

$$\begin{aligned}
\big\|\big[\hat{K} \cdot \hat{K}_\lambda^{-1} - \hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1}\big]\bar{y}\big\| &\overset{(i)}{\leq} \lambda \cdot \|\hat{K}_{P,\lambda}^{-1}\| \cdot \|\hat{K} - \hat{K}_P\big\| \cdot \|\hat{K}_\lambda^{-1}\| \cdot \|\bar{y}\| \\
&\overset{(ii)}{\leq} \frac{\sqrt{N} \cdot R}{\lambda}\, \|\hat{K} - \hat{K}_P\big\| \\
&\overset{(iii)}{\leq} \frac{R}{\sqrt{N} \cdot \lambda}\, \|\mathbf{K} - \mathbf{K}_P\big\|
\end{aligned}$$

where inequality (i) results from the Cauchy-Schwartz inequality; (ii) uses the fact that $\|\hat{K}_{P,\lambda}^{-1}\| \leq 1/\lambda$, $\|\hat{K}_\lambda^{-1}\| \leq 1/\lambda$ and $\|\bar{y}\| \leq \sqrt{N} \cdot R$, and (iii) utilizes the definition of $\hat{K}$ and $\hat{K}_P$.

Next, we utilize high probability bounds for the term $\|\mathbf{K} - \mathbf{K}_P\|$, which characterize the effect of sampling using P observations. Towards this end, we invoke Proposition F.4, which provides two high probability bounds for this term, one for the GIP, and one for the RF kernel. Specifically, for the former kernel, we have the following

$$\big\|\big[\hat{K} \cdot \hat{K}_\lambda^{-1} - \hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1}\big]\bar{y}\big\| \leq \frac{R \cdot G \cdot \sqrt{N}}{\lambda}\bigg(\sqrt{\frac{32\pi^2}{P}\log\frac{2N}{\delta}} + \frac{8\pi}{3P}\log\frac{2N}{\delta}\bigg),$$

with probability at least $1 - \delta$. Similarly, we have for the RF kernel

$$\big\|\big[\hat{K} \cdot \hat{K}_\lambda^{-1} - \hat{K}_P \cdot \hat{K}_{P,\lambda}^{-1}\big]\bar{y}\big\| \leq \frac{R \cdot \kappa^2 \cdot \sqrt{N}}{\lambda}\bigg(\sqrt{\frac{8}{P}\log\frac{2N}{\delta}} + \frac{4}{3P}\log\frac{2N}{\delta}\bigg),$$

with probability at least $1 - \delta$. This completes the analysis for term(B) in (26). Next, we consider the term(A) in (26). We have:

$$\begin{aligned}
\Big\|\big[\hat{K}_P\hat{K}_{P,\lambda}^{-1} + \hat{K}\hat{K}_\lambda^{-1}\big]\bar{y}\Big\| &\leq \big\|\big[\hat{K}_P\hat{K}_{P,\lambda}^{-1}\big]\bar{y}\big\| + \big\|\big[\hat{K}\hat{K}_\lambda^{-1}\big]\bar{y}\big\| \\
&\leq 2\|\bar{y}\| \leq 2\sqrt{N} \cdot R.
\end{aligned}$$

Now substituting the bounds for term(A) and term(B) into (26), we get the following bounds:

For the GIP kernel:

$$\Big|\hat{\mathcal{L}}_P(\hat{\alpha}_P^*) - \hat{\mathcal{L}}(\hat{\alpha}^*)\Big| \leq \frac{2R^2 \cdot G}{\lambda} \cdot \bigg(\sqrt{\frac{32\pi^2}{P}\log\frac{2N}{\delta}} + \frac{8\pi}{3P}\log\frac{2N}{\delta}\bigg). \qquad (28)$$

For the RF kernel:

$$\Big|\hat{\mathcal{L}}_P(\hat{\alpha}_P^*) - \hat{\mathcal{L}}(\hat{\alpha}^*)\Big| \leq \frac{2R^2 \cdot \kappa^2}{\lambda} \cdot \bigg(\sqrt{\frac{8}{P}\log\frac{2N}{\delta}} + \frac{4}{3P}\log\frac{2N}{\delta}\bigg). \qquad (29)$$

Next, we bound the loss incurred because of multi-agent kernel approximation for the oveall objective for both the kernels. Note from the definition of the $\hat{\mathcal{R}}(\alpha)$ and $\hat{\mathcal{R}}_P(\alpha)$

$$\begin{aligned}\left|\hat{\mathcal{R}}(\hat{\alpha}^*) - \hat{\mathcal{R}}(\hat{\alpha}_P^*)\right| &= \left|\hat{\mathcal{L}}(\hat{\alpha}^*) - \hat{\mathcal{L}}(\hat{\alpha}_P^*) + \frac{\lambda}{2}\|\hat{\alpha}^*\|_{\mathbf{K}}^2 - \frac{\lambda}{2}\|\hat{\alpha}_P^*\|_{\mathbf{K}_P}^2\right| \\ &\leq \left|\hat{\mathcal{L}}(\hat{\alpha}^*) - \hat{\mathcal{L}}(\hat{\alpha}_P^*)\right| + \frac{\lambda}{2}\left|\|\hat{\alpha}^*\|_{\mathbf{K}}^2 - \|\hat{\alpha}_P^*\|_{\mathbf{K}_P}^2\right|, \end{aligned} \tag{30}$$

which follows from the use of triangle inequality. Note that we have already bounded the first term. In the following, we bound the second term of (30) above.

$$\begin{aligned}&\frac{\lambda}{2}\cdot\left|\|\hat{\alpha}^*\|_{\mathbf{K}}^2 - \|\hat{\alpha}_P^*\|_{\mathbf{K}_P}^2\right| \overset{(i)}{=} \frac{\lambda}{2N}\cdot\left|\bar{y}^T\left[\hat{K}_\lambda^{-1}\cdot\hat{K}\cdot\hat{K}_\lambda^{-1} - \hat{K}_{P,\lambda}^{-1}\cdot\hat{K}_P\cdot\hat{K}_{P,\lambda}^{-1}\right]\bar{y}\right| \\ &\overset{(ii)}{\leq} \frac{\lambda}{2N}\cdot\|\bar{y}\|^2\cdot\left\|\hat{K}_\lambda^{-1}\cdot\hat{K}\cdot\hat{K}_\lambda^{-1} - \hat{K}_{P,\lambda}^{-1}\cdot\hat{K}_P\cdot\hat{K}_{P,\lambda}^{-1}\right\| \\ &\overset{(iii)}{\leq} \frac{\lambda}{2}\cdot R^2\cdot\Bigg[\underbrace{\left\|\hat{K}_\lambda^{-1}\cdot\hat{K}\cdot\hat{K}_\lambda^{-1} - \hat{K}_{P,\lambda}^{-1}\cdot\hat{K}\cdot\hat{K}_\lambda^{-1}\right\|}_{\text{term(C)}} + \underbrace{\left\|\hat{K}_{P,\lambda}^{-1}\cdot\hat{K}\cdot\hat{K}_\lambda^{-1} - \hat{K}_{P,\lambda}^{-1}\cdot\hat{K}_P\cdot\hat{K}_{P,\lambda}^{-1}\right\|}_{\text{term(D)}}\Bigg],\end{aligned} \tag{31}$$

where (i) follows from the definition of $\hat{\alpha}^*$ and $\hat{\alpha}_P^*$ along with the notation in (23) and (24); (ii) results from Cauchy-Schwartz inequality, and (iii) uses Assumption 3 and the triangle inequality. Next, we bound term(C) and term(D) in (31). First, from the definition of term(C), we have

$$\begin{aligned}\text{term(C)} = \left\|\hat{K}_\lambda^{-1}\cdot\hat{K}\cdot\hat{K}_\lambda^{-1} - \hat{K}_{P,\lambda}^{-1}\cdot\hat{K}\cdot\hat{K}_\lambda^{-1}\right\| &\overset{(i)}{\leq} \left\|\hat{K}_\lambda^{-1} - \hat{K}_{P,\lambda}^{-1}\right\|\cdot\left\|\hat{K}\cdot\hat{K}_\lambda^{-1}\right\| \\ &\overset{(ii)}{\leq} \left\|\hat{K}_\lambda^{-1}\right\|\cdot\left\|\hat{K}_P - \hat{K}\right\|\cdot\left\|\hat{K}_{P,\lambda}^{-1}\right\|\cdot\left\|\hat{K}\cdot\hat{K}_\lambda^{-1}\right\| \\ &\overset{(iii)}{\leq} \frac{1}{\lambda^2}\cdot\left\|\hat{K}_P - \hat{K}\right\| \\ &\overset{(iv)}{\leq} \frac{1}{N\cdot\lambda^2}\cdot\left\|\mathbf{K}_P - \mathbf{K}\right\|,\end{aligned}$$

where (i) results from the Cauchy-Schwartz inequality; (ii) utilizes $A^{-1} - B^{-1} = A^{-1}(B-A)B^{-1}$ for invertible matrices A and B along with Cauchy-Schwartz inequality; (iii) uses that fact that we have $\|\hat{K}_\lambda^{-1}\| \leq 1/\lambda$, $\|\hat{K}_{P,\lambda}^{-1}\| \leq 1/\lambda$ and $\|\hat{K}\cdot\hat{K}_\lambda^{-1}\| \leq 1$, and (iv) results from the notation in (23). Next, we bound term(D) as

$$\begin{aligned}\text{term(D)} = \left\|\hat{K}_{P,\lambda}^{-1}\cdot K\cdot\hat{K}_\lambda^{-1} - \hat{K}_{P,\lambda}^{-1}\cdot\hat{K}_P\cdot\hat{K}_{P,\lambda}^{-1}\right\| &\overset{(i)}{\leq} \left\|\hat{K}_{P,\lambda}^{-1}\right\|\cdot\left\|\hat{K}\cdot\hat{K}_\lambda^{-1} - \hat{K}_P\cdot\hat{K}_{P,\lambda}^{-1}\right\| \\ &\overset{(ii)}{\leq} \lambda\cdot\left\|\hat{K}_{P,\lambda}^{-1}\right\|^2\cdot\left\|\hat{K}_P - \hat{K}\right\|\cdot\left\|\hat{K}_\lambda^{-1}\right\| \\ &\overset{(iii)}{\leq} \frac{1}{\lambda^2}\cdot\left\|\hat{K}_P - \hat{K}\right\| \\ &\overset{(iv)}{\leq} \frac{1}{N\cdot\lambda^2}\cdot\left\|\mathbf{K}_P - \mathbf{K}\right\|,\end{aligned}$$

where inequality (i) follows from Cauchy-Schwartz inequality; (ii) results from (27) and the application of Cauchy-Schwartz inequality; (iii) follows from the fact that we have $\|\hat{K}_\lambda^{-1}\| \leq 1/\lambda$ and $\|\hat{K}_{P,\lambda}^{-1}\| \leq 1/\lambda$, and (iv) results from the notation defined in (23).

Substituting the upper bounds of term(C) and term(D) in (31), we get

$$\frac{\lambda}{2}\left|\|\hat{\alpha}^*\|_{\mathbf{K}}^2 - \|\hat{\alpha}_P^*\|_{\mathbf{K}_P}^2\right| \leq \frac{R^2}{N\cdot\lambda}\cdot\left\|\mathbf{K}_P - \mathbf{K}\right\|.$$

Finally, using Lemma F.4, we have for the GIP kernel

$$\frac{\lambda}{2}\left|\|\hat{\alpha}^*\|_{\mathbf{K}}^2 - \|\hat{\alpha}_P^*\|_{\mathbf{K}_P}^2\right| \leq \frac{R^2\cdot G}{\lambda}\cdot\left(\sqrt{\frac{32\pi^2}{P}\log\frac{2N}{\delta}} + \frac{8\pi}{3P}\log\frac{2N}{\delta}\right),$$

Substituting, the above expression in (30) and combining with the bound in (28), we get the statement of the theorem.

Similarly, we have for the RF kernel

$$\frac{\lambda}{2}\Big|\|\hat{\alpha}^*\|_{\mathbf{K}}^2 - \|\hat{\alpha}_P^*\|_{\mathbf{K}_P}^2\Big| \leq \frac{R^2\cdot\kappa^2}{\lambda}\cdot\left(\sqrt{\frac{8}{P}\log\frac{2N}{\delta}} + \frac{4}{3P}\log\frac{2N}{\delta}\right).$$

Finally, substituting the above expression in (30) and combining with the bound in (29), we get the statement of the theorem.

Therefore, the proof is complete. □

Finally, the proof of Theorem 4.2 is a simple consequence of Theorem F.5.

G Generalization performance of multi-agent kernel approximation framework: Proof of Theorem 4.3

To prove generalization bounds we extensively use the notations defined in Appendix E.

Definition G.1. We define by f_π as the MMSE estimator for f as

$$f_\pi(x) = \mathbb{E}_{y\sim\pi(y|x)}[y|x].$$

We make the following assumption.

Assumption 5. There exists $f_{\mathcal{H}} \in \mathcal{H}$ s.t. $\mathcal{L}(f_{\mathcal{H}}) = \inf_{h\in\mathcal{H}}\mathcal{L}(h)$. From the definition of MMSE estimator above we have $f_\pi = f_{\mathcal{H}}$. Moreover, this assumption implies that there exists $h \in L_2(\mathcal{X}, \pi_x)$ s.t. we have

$$f_{\mathcal{H}}(x) = (K^{1/2}h)(x). \tag{32}$$

for some $\|h\|_{\pi_x} \leq R$.

Note that Assumption 5 is same as Assumption 4 in the main paper. The existence of $f_{\mathcal{H}}$ ensures that (32) holds for some $h \in L_2(\mathcal{X}, \pi_x)$. Moreover, without loss of generality we assume $\|h\|_{\pi_x}$ to be bounded by R (same R as in Assumption 3).

Note that for the ℓ_2 loss defined in (10) and under Assumption 5, we have for any $f \in L_2(\mathcal{X}, \pi_x)$ (Caponnetto & De Vito, 2007)

$$\mathcal{L}(f) - \inf_{h\in\mathcal{H}}\mathcal{L}(h) = \mathcal{L}(f) - \mathcal{L}(f_\pi) = \|f - f_\pi\|_{\pi_x}^2. \tag{33}$$

Now the goal is to bound the generalization error with f replaced by the $\hat{f}_P$ (learned by solving the approximate kernel problem stated in (12)). Using the notations defined in Appendix E, we have

$$\begin{aligned}\hat{f}_P &= \Phi_P\hat{\Phi}_P^*\Big[\big[\hat{K}_P + \lambda\cdot I\big]^{-1}\hat{y}\Big] = \Phi_P\hat{\Phi}_P^*\Big[\big[\hat{\Phi}_P\hat{\Phi}_P^* + \lambda\cdot I\big]^{-1}\hat{y}\Big]\\ &= \Phi_P\Big[\big[\hat{\Phi}_P^*\hat{\Phi}_P + \lambda\cdot I\big]^{-1}\hat{\Phi}_P^*\hat{y}\Big] = \Phi_P\cdot\hat{C}_{P,\lambda}^{-1}\cdot\hat{\Phi}_P^*\cdot\hat{y}.\end{aligned}$$

where we have defined $\hat{y} = \frac{1}{\sqrt{N}}\bar{y}$.

Using the above we have

$$\hat{f}_P - f_\pi = \underbrace{\Phi_P\cdot\hat{C}_{P,\lambda}^{-1}\cdot\hat{\Phi}_P^*\cdot\hat{y} - \Phi_P\cdot\hat{C}_{P,\lambda}^{-1}\cdot\Phi_P^*\cdot f_\pi}_{\text{term(A)}} \tag{34}$$

$$+ \underbrace{\Phi_P\cdot\hat{C}_{P,\lambda}^{-1}\cdot\Phi_P^*\cdot f_\pi - \Phi_P\cdot C_{P,\lambda}^{-1}\cdot\Phi_P^*\cdot f_\pi}_{\text{term(B)}} \tag{35}$$

$$+ \underbrace{\Phi_P\cdot C_{P,\lambda}^{-1}\cdot\Phi_P^*\cdot f_\pi - K\cdot K_\lambda^{-1}\cdot f_\pi}_{\text{term(C)}} \tag{36}$$

$$+ \underbrace{K\cdot K_\lambda^{-1}\cdot f_\pi - f_\pi}_{\text{term(D)}}. \tag{37}$$

Next, we bound each term separately below for both the GIP and the RF kernels. We note that for the RF kernel we follow the proofs developed in Rudi & Rosasco (2017) with some minor corrections.

Bound for term(A) First, we consider term(A) given in (34).

Lemma G.1. *For any $\delta \in (0,1]$, with $\lambda \leq \frac{3}{4}\|K\|$ and $N \geq \max\left\{\frac{72\kappa^2}{\lambda}, \frac{2\lambda}{9}\right\}\log\frac{4\kappa^2}{\lambda\cdot\delta}$. Moreover, with $P \geq \max\left\{8, \frac{512\pi^2 G^2}{\|K\|^2}\right\}\log\frac{2}{\delta}$ for the GIP kernel and with $P \geq \max\left\{8\kappa^2, \frac{32\kappa^2}{\|K\|^2}\right\}\log\frac{2}{\delta}$ for the RF kernel, we have with probability at least $1-3\delta$*

$$\|\text{term(A)}\|_{\pi_x} = \left\|\Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot \hat{\Phi}_P^* \cdot \hat{y} - \Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot \Phi_P^* \cdot f_\pi\right\|_{\pi_x} \leq \frac{6R\cdot\kappa}{\sqrt{\lambda}\cdot N}\log\frac{2}{\delta} + \sqrt{\frac{18R^2\cdot\kappa^2}{\lambda\cdot N}\log\frac{2}{\delta}}.$$

Proof. From the definition of term(A) in (34), we have

$$\begin{aligned}
\text{term(A)} &= \Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot \hat{\Phi}_P^* \cdot \hat{y} \;-\; \Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot \Phi_P^* \cdot f_\pi \\
&= \Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot \left[\hat{\Phi}_P^* \cdot \hat{y} \;-\; \Phi_P^* \cdot f_\pi\right] \\
&\overset{(i)}{=} \underbrace{\left(\Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot C_{P,\lambda}^{1/2}\right)}_{\text{term(A1)}} \cdot \underbrace{\left(C_{P,\lambda}^{-1/2} \cdot \left[\hat{\Phi}_P^* \cdot \hat{y} \;-\; \Phi_P^* \cdot f_\pi\right]\right)}_{\text{term(A2)}}.
\end{aligned} \tag{38}$$

where (i) follows from $C_{P,\lambda}^{1/2} \cdot C_{P,\lambda}^{-1/2} = I$. Next, we individually bound term(A1) and term(A2) below.

First, we bound term(A1) using Proposition 8 in Rudi & Rosasco (2017), we define $\beta_1 := \lambda_{\max}\left[C_{P,\lambda}^{-1/2} \cdot \left(C_P - \hat{C}_P\right) \cdot C_{P,\lambda}^{-1/2}\right]$

$$\begin{aligned}
\|\text{term(A1)}\| &= \|\Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot C_{P,\lambda}^{1/2}\| \overset{(i)}{\leq} \|\Phi_P \cdot \hat{C}_{P,\lambda}^{-1/2}\| \cdot \|\hat{C}_{P,\lambda}^{-1/2} \cdot C_{P,\lambda}^{1/2}\| \\
&\overset{(ii)}{\leq} \|\hat{C}_{P,\lambda}^{-1/2} \cdot C_{P,\lambda}^{1/2}\|^2 \overset{(iii)}{\leq} \frac{1}{1-\beta_1},
\end{aligned}$$

where (i) utilizes the Cauchy-Schwartz inequality; (ii) follows from the fact that:

$$\|\Phi_P \cdot \hat{C}_{P,\lambda}^{-1/2}\| = \underbrace{\|\Phi_P \cdot C_{P,\lambda}^{-1/2}\|}_{\leq 1} \cdot \|C_{P,\lambda}^{1/2} \cdot \; \hat{C}_{P,\lambda}^{-1/2}\| \leq \|\hat{C}_{P,\lambda}^{-1/2} \cdot C_{P,\lambda}^{1/2}\|,$$

finally, (iii) utilizes Proposition 8 in Rudi & Rosasco (2017). Next, we bound term β_1. Using Lemma G.2, with the choice of $\lambda \leq \|C_P\|$ and with $N \geq \max\left\{\frac{72\kappa^2}{\lambda}, \frac{2\lambda}{9}\right\}\log\frac{4\kappa^2}{\lambda\delta}$, we have with probability at least $1-\delta$

$$\beta_1 = \lambda_{\max}\left[C_{P,\lambda}^{-1/2} \cdot \left(C_P - \hat{C}_P\right) \cdot C_{P,\lambda}^{-1/2}\right] \leq \frac{1}{3},$$

Finally, we need to ensure that the random variable $\|C_P\| \geq \lambda$ with high probability for both the GIP and the RF kernel.

Note from Proposition G.7 that for: (i) the GIP kernel with $P \geq \max\left\{8, \frac{512\pi^2 G^2}{\|K\|^2}\right\}\log\frac{2}{\delta}$ and (ii) for the RF kernel with $P \geq \max\left\{8\kappa^2, \frac{32\kappa^2}{\|K\|^2}\right\}\log\frac{2}{\delta}$, we have with probability at least $1-\delta$

$$\|K - K_P\| \leq \frac{1}{4}\|K\|.$$

Next, choosing λ s.t. we have $\lambda \leq \frac{3}{4}\|K\|$. Note that the choice of λ and the choice of P implies that we have with probability at least $1-\delta$

$$\begin{aligned}
\|C_P\| &= \|\Phi_P^* \cdot \Phi_P\| = \|\Phi_P \cdot \Phi_P^*\| = \|K_P\| \\
&\geq \left|\|K\| - \|K - K_P\|\right| \geq \|K\| - \|K - K_P\| \geq \frac{3}{4}\|K\| \geq \lambda.
\end{aligned}$$

Now combining the two facts that $\beta_1 \leq 1/3$ holds with probability $1-\delta$ conditioned on $\|C_P\| \geq \lambda$ and $\|C_P\| \geq \lambda$ holds with probability at least $1-\delta$, it is easy to verify that we have $\beta_1 \leq 1/3$ with probability at least $1-2\delta$. This further implies that we have with probability at least $1-2\delta$

$$\|\text{term}(A1)\| \leq \frac{3}{2}. \tag{39}$$

Next, we consider term(A2) in (38)[4]

$$\text{term(A2)} = C_{P,\lambda}^{-1/2} \cdot \big[\hat{\Phi}_P^* \cdot \hat{y} \; - \; \Phi_P^* \cdot f_\pi\big] = C_{P,\lambda}^{-1/2} \cdot \left[\frac{1}{N}\sum_{i=1}^{N}\big(\phi_P(x_i)\cdot y_i - \Phi^* \cdot f_\pi\big)\right]$$

We modify Lemma 6 in Rudi & Rosasco (2017) to bound this term. First, note that the random variable $\phi_P(x_i)\cdot y_i - \Phi_P^* \cdot f_\pi$ is zero mean. Next, we apply Bernstein's inequality stated in Proposition H.2. Consider the random vector z_i defined as

$$z_i = C_{P,\lambda}^{-1/2} \cdot \phi_P(x_i) \cdot y_i,$$

with mean $\mathbb{E}[z_i] = \mu = C_{P,\lambda}^{-1/2} \cdot \Phi_L^* \cdot f_\pi$. Note that z_i are i.i.d. with the moments of random variable $z = C_{L,\lambda}^{-1/2} \cdot \phi_P(x) \cdot y$ bounded as

$$\begin{aligned}
&\mathbb{E}\|z_i - \mu\|_{\mathcal{H}_P}^p = \mathbb{E}\|z_i - \mathbb{E}[z]\|_{\mathcal{H}_P}^p \\
&\overset{(i)}{\leq} \mathbb{E}_z\mathbb{E}_{z_i}\|z_i - z\|_{\mathcal{H}_P}^p \overset{(ii)}{\leq} 2^{p-1}\mathbb{E}_z\mathbb{E}_{z_i}\big(\|z_i\|_{\mathcal{H}_P}^p + \|z\|_{\mathcal{H}_P}^p\big) = 2^p\mathbb{E}\|z\|_{\mathcal{H}_P}^p,
\end{aligned}$$

where (i) follows from the Jensen's inequality and (ii) utilizes the convexity of $\|\cdot\|_{\mathcal{H}_P}^p$. Next, we bound $\mathbb{E}\|z\|_{\mathcal{H}_P}^p$ as

$$\begin{aligned}
&\mathbb{E}\|z\|_{\mathcal{H}_P}^p = \mathbb{E}\big\|C_{P,\lambda}^{-1/2} \cdot \phi_P(x) \cdot y\big\|_{\mathcal{H}_P}^p \\
&\overset{(i)}{\leq} \mathbb{E}\big[\|C_{P,\lambda}^{-1/2} \cdot \phi_P(x)\|_{\mathcal{H}_P}^p \cdot |y|^p\big] \overset{(ii)}{\leq} R^p \cdot \mathbb{E}\big[\big\|C_{P,\lambda}^{-1/2} \cdot \phi_P(x)\big\|_{\mathcal{H}_P}^p\big] \overset{(iii)}{\leq} \left(\frac{R\cdot\kappa}{\sqrt{\lambda}}\right)^p,
\end{aligned}$$

where (i) utilizes Cauchy-Schwartz inequality; (ii) follows from Assumption 3 and, (iii) results from the following

$$\big\|C_{P,\lambda}^{-1/2} \cdot \phi_P(x)\big\|_{\mathcal{H}_P}^2 \leq \frac{1}{\lambda}\cdot\|\phi_P(x)\|_{\mathcal{H}_P}^2 = \frac{1}{\lambda}\cdot\langle\phi_P(x),\phi_P(x)\rangle_{\mathcal{H}_P} = \frac{1}{\lambda}\cdot k_P(x,x) \leq \frac{\kappa^2}{\lambda},$$

where we utilized the definition of $k_P(\cdot,\cdot)$ and Assumption 1. Combining with the above we get

$$\mathbb{E}\|z_i - \mu\|_{\mathcal{H}_P}^p \leq \left(\frac{2R\cdot\kappa}{\sqrt{\lambda}}\right)^p.$$

This implies that we can apply Bernstein's inequality stated in Proposition H.2 with $\sigma = B = 2R\kappa/\sqrt{\lambda}$, therefore, we get with probability at least $1-\delta$

$$\|\text{term(A2)}\|_{\mathcal{H}_P} \leq \frac{4R\cdot\kappa}{\sqrt{\lambda}\cdot N}\log\frac{2}{\delta} + \sqrt{\frac{8R^2\cdot\kappa^2}{\lambda\cdot N}\log\frac{2}{\delta}}. \tag{40}$$

Finally, substituting the bound for term (A1) derived in (39) and the bound for term (A2) derived in (40) in (38) and noting the fact that (39) holds with probability at least $1-2\delta$ and (40) holds with probability at least $1-\delta$, we have with probability at least $1-3\delta$

$$\|\text{term(A)}\|_{\pi_x} \leq \frac{6R\cdot\kappa}{\sqrt{\lambda}\cdot N}\log\frac{2}{\delta} + \sqrt{\frac{18R^2\cdot\kappa^2}{\lambda\cdot N}\log\frac{2}{\delta}}.$$

Therefore, the lemma is proved. □

Lemma G.2. *For some $\delta \in (0,1]$ and with $\lambda \leq \|C_P\|$ we have with probability at least $1-\delta$*

$$\lambda_{\max}\big[C_{P,\lambda}^{-1/2}\cdot\big(C_P - \hat{C}_P\big)\cdot C_{P,\lambda}^{-1/2}\big] \leq \frac{2}{3N}\log\frac{4\kappa^2}{\lambda\cdot\delta} + \sqrt{\frac{2\kappa^2}{\lambda\cdot N}\log\frac{4\kappa^2}{\lambda\cdot\delta}}.$$

Proof. The proof follows from Proposition 6 in Rudi & Rosasco (2017). □

Next, we bound term(B).

[4] For ease of presentation, we relabel the set $\{(x_m^{(i)}, y_m^{(i)}) : \forall i \in \mathcal{N}_m, \forall m \in [M]\}$ with $\{(x_i, y_i)\}_{i=1}^N$.

Bound for term(B) Let us consider term(B) in (35), we have

Lemma G.3. *For $\delta \in (0,1]$, with $P \geq \max\left\{8, \frac{288\pi^2 G^2}{\lambda^2}\right\} \log \frac{2}{\delta}$ for the GIP kernel and with $P \geq \max\left\{\frac{2\lambda}{9}, \frac{72\kappa^2}{\lambda}\right\} \log \frac{4\kappa^2}{\lambda \cdot \delta}$ for the RF kernel, we have with probability at least $1-4\delta$*

$$\|\text{term(B)}\|_{\pi_x} = \left\|\Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot \Phi_P^* \cdot f_\pi - \Phi_P \cdot C_{P,\lambda}^{-1} \cdot \Phi_P^* \cdot f_\pi\right\| \leq \frac{12R \cdot \kappa^2}{\sqrt{\lambda} \cdot N} \log \frac{2}{\delta} + \sqrt{\frac{36R^2 \cdot \kappa^4}{\lambda \cdot N} \log \frac{2}{\delta}}.$$

Proof. Consider term(B), note that from Lemma 3 in Rudi & Rosasco (2017), we have

$$\begin{aligned}\|\text{term(B)}\|_{\pi_x} &= \left\|\Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot \Phi_P^* \cdot f_\pi - \Phi_P \cdot C_{P,\lambda}^{-1} \cdot \Phi_P^* \cdot f_\pi\right\|_{\pi_x} \\ &\leq R \cdot \underbrace{\left\|K_{P,\lambda}^{-1/2} \cdot K^{1/2}\right\|}_{\text{term(B1)}} \cdot \underbrace{\left\|\Phi_P \cdot \hat{C}_{P,\lambda}^{-1} \cdot C_{P,\lambda}^{1/2}\right\|}_{\text{term(B2)}} \cdot \underbrace{\left\|C_{P,\lambda}^{-1/2} \cdot \left(C_P - \hat{C}_P\right)\right\|}_{\text{term(B3)}}. \end{aligned} \tag{41}$$

Next, we bound each term individually.

First, we consider term(B1) in (41) above, we define $\beta_2 \coloneqq \lambda_{\max}\left[K_\lambda^{-1/2} \cdot \left(K - K_P\right) \cdot K_\lambda^{-1/2}\right]$, then we have

$$\|\text{term(B1)}\| = \left\|K_{P,\lambda}^{-1/2} \cdot K^{1/2}\right\| \overset{(i)}{\leq} \left\|K_{P,\lambda}^{-1/2} \cdot K_\lambda^{1/2}\right\| \overset{(ii)}{\leq} \left(\frac{1}{1-\beta_2}\right)^{1/2},$$

where (i) follows from the fact that $\|K^{1/2}\| \leq \|K_\lambda^{1/2}\|$ and (ii) utilizes Proposition 8 in Rudi & Rosasco (2017). Note that from Lemma G.4 for (i) the GIP kernel with $P \geq \max\left\{8, \frac{288\pi^2 G^2}{\lambda^2}\right\} \log \frac{2}{\delta}$ and for (ii) the RF kernel with $P \geq \max\left\{\frac{2\lambda}{9}, \frac{72\kappa^2}{\lambda}\right\} \log \frac{4\kappa^2}{\lambda \cdot \delta}$, we have with probability at least $1-\delta$

$$\beta_2 \leq \frac{1}{3}.$$

This implies that with probability at least $1-\delta$, we have

$$\|\text{term(B1)}\| \leq 2. \tag{42}$$

Next, note from the definition of term(B2) in (41) that it is same as term(A1) in (38). Therefore, using the same bound as derived in (39), we have with probability at least $1-2\delta$

$$\|\text{term(B2)}\| \leq \frac{3}{2}. \tag{43}$$

Next, we bound term(B3) in (41). Using Equation (28) of Lemma 7 in Rudi & Rosasco (2017), we have with probability at least $1-\delta$

$$\|\text{term(B3)}\| \leq \frac{4\kappa^2}{\sqrt{\lambda} \cdot N} \log \frac{2}{\delta} + \sqrt{\frac{4\kappa^4}{\lambda \cdot N} \log \frac{2}{\delta}}. \tag{44}$$

Finally, substituting the bounds derived in (42), (43) and (44) in (41) and utilizing the fact that (42) holds with probability at least $1-\delta$, (43) holds with probability at least $1-2\delta$ and (44) holds with probability at least $1-\delta$, we get with probability at least $1-4\delta$

$$\|\text{term(B)}\|_{\pi_x} \leq \frac{12R \cdot \kappa^2}{\sqrt{\lambda} \cdot N} \log \frac{2}{\delta} + \sqrt{\frac{36R^2 \cdot \kappa^4}{\lambda \cdot N} \log \frac{2}{\delta}}.$$

Therefore, the lemma is proved. □

Lemma G.4. *For some $\delta \in (0,1]$, then with probability at least $1-\delta$ for*

1. *We have for the GIP kernel*

$$\lambda_{\max}\left[K_\lambda^{-1/2} \cdot \left(K - K_P\right) \cdot K_\lambda^{-1/2}\right] \leq \frac{8\pi \cdot G}{\lambda \cdot P} \log \frac{2}{\delta} + \sqrt{\frac{8\pi^2 \cdot G^2}{\lambda^2 \cdot P} \log \frac{2}{\delta}}.$$

2. *We have for the RF kernel (Proposition 6 in Rudi & Rosasco (2017))*

$$\lambda_{\max}\big[K_\lambda^{-1/2}\cdot\big(K-K_P\big)\cdot K_\lambda^{-1/2}\big]\leq\frac{2}{3P}\log\frac{4\kappa^2}{\lambda\cdot\delta}+\sqrt{\frac{2\kappa^2}{\lambda\cdot P}\log\frac{4\kappa^2}{\lambda\cdot\delta}}.$$

Proof. For the GIP kernel, we have

$$\lambda_{\max}\big[K_\lambda^{-1/2}\cdot\big(K-K_P\big)\cdot K_\lambda^{-1/2}\big]\overset{(i)}{\leq}\|K_\lambda^{-1/2}\cdot\big(K-K_P\big)\cdot K_\lambda^{-1/2}\|\overset{(ii)}{\leq}\frac{1}{\lambda}\cdot\|K-K_P\|.$$

where (i) follows from the definition of the operator norm and (ii) uses the fact that $\|K_\lambda^{-1/2}\|\leq 1/\sqrt{\lambda}$. Next, utilizing the bound for $\|K-K_P\|$ from Proposition G.7, we have the proof of the lemma. □

Next, we consider term(C).

Bound for term(C) We consider term(C) in (36).

Lemma G.5. *For $\delta\in(0,1]$, with probability at least $1-\delta$*

1. *For the GIP kernel, we have*

$$\|\text{term(C)}\|_{\pi_x}=\big\|\Phi_P\cdot C_{P,\lambda}^{-1}\cdot\Phi_P^*\cdot f_\pi-K\cdot K_\lambda^{-1}\cdot f_\pi\big\|\leq\frac{8\pi R\cdot G}{\sqrt{\lambda}\cdot P}\log\frac{2}{\delta}+\sqrt{\frac{8\pi^2R^2\cdot G^2}{\lambda\cdot P}\log\frac{2}{\delta}}.$$

2. *For the RF kernel, with $P\geq\max\left\{\frac{2\lambda}{9},\frac{72\kappa^2}{\lambda}\right\}\log\frac{4\kappa^2}{\lambda\cdot\delta}$ we have from Lemma 8 in Rudi & Rosasco (2017)*

$$\begin{aligned}\|\text{term(C)}\|_{\pi_x}&=\big\|\Phi_P\cdot C_{P,\lambda}^{-1}\cdot\Phi_P^*\cdot f_\pi-K\cdot K_\lambda^{-1}\cdot f_\pi\big\|\\&\leq\frac{4R\cdot(\lambda+\kappa^2)}{3\sqrt{\lambda}\cdot P}\log\frac{16\kappa^2}{\lambda\cdot\delta}+\sqrt{\frac{8R^2\cdot\kappa^2}{P}\log\frac{16\kappa^2}{\lambda\cdot\delta}}.\end{aligned}$$

Proof. Let us consider the first term of term(C) for the GIP kernel, we have from the definition of $C_{P,\lambda}$

$$\begin{aligned}&\Phi_P\cdot C_{P,\lambda}^{-1}\cdot\Phi_P^*\cdot f_\pi=\Phi_P\cdot\big(C_P+\lambda I\big)^{-1}\cdot\Phi_P^*\cdot f_\pi\overset{(i)}{=}\Phi_P\cdot\big(\Phi_P^*\cdot\Phi_P+\lambda\cdot I\big)^{-1}\cdot\Phi_P^*\cdot f_\pi\\&\overset{(ii)}{=}\Phi_P\cdot\Phi_P^*\cdot\big(\Phi_P\cdot\Phi_P^*+\lambda\cdot I\big)^{-1}\cdot f_\pi\overset{(iii)}{=}K_P\cdot\big(K_P+\lambda\cdot I\big)^{-1}\cdot f_\pi\overset{(iv)}{=}K_P\cdot K_{P,\lambda}^{-1}\cdot f_\pi\end{aligned}$$

where (i) follows from the definition of C_P; (ii) uses the fact that for any continuous spectral function and compact operator Z, we have $F(Z^*Z)Z^*=Z^*F(ZZ^*)$; (iii) uses the definition of K_P and finally, (iv) results from the notation of $K_{P,\lambda}$. Therefore, we have

$$\begin{aligned}\text{term(C)}&=K_P\cdot K_{P,\lambda}^{-1}\cdot f_\pi-K\cdot K_\lambda^{-1}\cdot f_\pi\\&=\big(K_P\cdot K_{P,\lambda}^{-1}-K\cdot K_\lambda^{-1}\big)\cdot f_\pi\\&\overset{(i)}{=}\lambda\cdot\big(K_{P,\lambda}^{-1}-K_\lambda^{-1}\big)\cdot f_\pi\\&\overset{(ii)}{=}\lambda\cdot K_{P,\lambda}^{-1}\big(K-K_P\big)K_\lambda^{-1}\cdot f_\pi\\&\overset{(iii)}{=}\lambda\cdot K_{P,\lambda}^{-1}\cdot\big(K-K_P\big)\cdot K_\lambda^{-1}\cdot K^{1/2}\cdot h\\&=\big(\lambda\cdot K_{P,\lambda}^{-1}\big)\cdot\big(K-K_P\big)\cdot K_\lambda^{-1/2}\cdot\big(K_\lambda^{-1/2}\cdot K^{1/2}\big)\cdot h\end{aligned}$$

where (i) uses the identity $A\cdot(A+\lambda\cdot I)^{-1}=I-\lambda(A+\lambda\cdot I)^{-1}$ for bounded positive operators; (ii) results from $A^{-1}-B^{-1}=A^{-1}\cdot(B-A)\cdot B^{-1}$ for invertible bounded positive operators, and (iii) utilizes Assumption 5. Taking the norm on both sides and utilizing Cauchy-Schwartz inequality, we get

$$\|\text{term(C)}\|_{\pi_x}\leq\big\|\lambda\cdot K_{P,\lambda}^{-1}\big\|\cdot\big\|K-K_P\big\|\cdot\big\|K_\lambda^{-1/2}\big\|\cdot\big\|K_\lambda^{-1/2}K^{1/2}\big\|\cdot\|h\|_{\pi_x}\leq\frac{R}{\sqrt{\lambda}}\cdot\|K-K_P\|,$$

where the last inequality follows from $\|\lambda \cdot K_{P,\lambda}^{-1}\| \leq 1$, $\|K_\lambda^{-1/2}\| \leq 1/\sqrt{\lambda}$, $\|K_\lambda^{-1/2} \cdot K^{1/2}\| \leq 1$ and $\|h\|_{\pi_x} \leq R$ from Assumption 5.

Finally, using Proposition G.7 we have for the generalized inner-product kernel with probability at least $1-\delta$

$$\|\text{term(C)}\|_{\pi_x} \leq \frac{8\pi R \cdot G}{\sqrt{\lambda} \cdot P} \log \frac{2}{\delta} + \sqrt{\frac{8\pi^2 R^2 \cdot G^2}{\lambda \cdot P} \log \frac{2}{\delta}}.$$

Therefore, the lemma is proved. □

Let us next consider term(D).

Bound for term(D) We consider term(D) in equation 37.

Lemma G.6. *We have*

$$\|\text{term(D)}\|_{\pi_x} = \|K \cdot K_\lambda^{-1} \cdot f_\pi - f_\pi\|_{\pi_x} \leq R \cdot \sqrt{\lambda}.$$

Proof. We have

$$\begin{aligned}
&\text{term(D)} = K \cdot K_\lambda^{-1} \cdot f_\pi - f_\pi = \big(K \cdot K_\lambda^{-1} - I\big) \cdot f_\pi \\
&\overset{(i)}{=} -\lambda \cdot K_\lambda^{-1} \cdot f_\pi \overset{(ii)}{=} -\lambda \cdot K_\lambda^{-1} \cdot K^{1/2} \cdot h = -\big(\lambda \cdot K_\lambda^{-1/2}\big) \cdot \big(K_\lambda^{-1/2} \cdot K^{1/2}\big) \cdot h,
\end{aligned}$$

where (i) uses the identity $A \cdot (A + \lambda \cdot I)^{-1} = I - \lambda(A + \lambda \cdot I)^{-1}$ and (ii) follows from Assumption 4. Taking the norm on both sides and applying Cauchy-Schwartz inequality, we get

$$\|\text{term(D)}\|_{\pi_x} \leq \|\lambda \cdot K_\lambda^{-1/2}\| \cdot \|K_\lambda^{-1/2} \cdot K^{1/2}\| \cdot \|h\|_{\pi_x} \leq R \cdot \sqrt{\lambda},$$

where the last inequality results from $\|\lambda^{1/2} \cdot K_\lambda^{-1/2}\| \leq 1$, $\|K_\lambda^{-1/2} \cdot K^{1/2}\| \leq 1$ and $\|h\|_{\pi_x} \leq R$ follows from Assumption 5. □

Next, we combine the four terms to bound $\|\hat{f}_P - f_\pi\|_{\pi_x}$.

Proposition G.7. *The following holds with probability at least* $1-\delta$

1. *For the GIP kernel, we have*

$$\|K - K_P\| \leq \frac{8\pi G}{P} \log \frac{2}{\delta} + \sqrt{\frac{8\pi^2 G^2}{P} \log \frac{2}{\delta}}.$$

2. *For the RF kernel, we have*

$$\|K - K_P\| \leq \frac{4\kappa^2}{P} \log \frac{2}{\delta} + \sqrt{\frac{2\kappa^2}{P} \log \frac{2}{\delta}}.$$

Proof. We first prove (i), note from the definition of $K - K_P$ and the norm in $L_2(\mathcal{X}, \pi_x)$-space we have

$$\begin{aligned}
\|K - K_P\| &= \sup_{\|h\|_{\pi_x}=1} \Big\| \int_{z \in \mathcal{X}} \big(k(x,z) - k_P(x,z)\big) h(z) d\pi_z \Big\|_{\pi_x} \\
&= \sup_{\|h\|_{\pi_x}=1} \Big\| \int_{z \in \mathcal{X}} \big(g(\psi(x,z), \|x\|, \|z\|) - g(\psi_P(x,z), \|x\|, \|z\|)\big) h(z) d\pi_z \Big\|_{\pi_x} \\
&= \sup_{\|h\|_{\pi_x}=1} \Big\| \int_{z \in \mathcal{X}} \big|g(\psi(x,z), \|x\|, \|z\|) - g(\psi_P(x,z), \|x\|, \|z\|)\big| h(z) d\pi_z \Big\|_{\pi_x} \\
&\leq G \sup_{\|h\|_{\pi_x}=1} \Big\| \int_{z \in \mathcal{X}} \big|\psi(x,z) - \psi_P(x,z)\big| h(z) d\pi_z \Big\|_{\pi_x}
\end{aligned}$$

Recall using $\zeta(x,\omega) = \mathbb{1}[\omega^T x \geq 0]$ for the GIP kernel, we have from the definition of $\psi(x,z)$ and $\psi_P(x,z)$

$$\psi(x,z) = \Big|\pi - 2\pi\mathbb{E}\big[\zeta(x,\omega)\zeta(z,\omega)\big]\Big| \quad \text{and} \quad \psi_P(x,z) = \Big|\pi - \frac{2\pi}{P}\sum_{\ell=1}^{P}\zeta(x,\omega_\ell)\zeta(z,\omega_\ell)\Big|.$$

Substituting in the above, we get

$$\begin{aligned}
&\|K - K_P\| \\
&\leq G \sup_{\|h\|_{\pi_x}=1} \left\| \int_{z\in\mathcal{X}} \left| \Big|\pi - 2\pi\mathbb{E}\big[\zeta(x,\omega)\cdot\zeta(z,\omega)\big]\Big| - \Big|\pi - \frac{2\pi}{P}\sum_{\ell=1}^{P}\zeta(x,\omega_\ell)\cdot\zeta(z,\omega_\ell)\Big| \right| \cdot h(z)d\pi_z \right\|_{\pi_x} \\
&\leq 2\pi G \sup_{\|h\|_{\pi_x}=1} \left\| \int_{z\in\mathcal{X}} \left| \mathbb{E}\big[\zeta(x,\omega)\cdot\zeta(z,\omega)\big] - \frac{1}{P}\sum_{\ell=1}^{P}\zeta(x,\omega_\ell)\cdot\zeta(z,\omega_\ell) \right| \cdot h(z)d\pi_z \right\|_{\pi_x} \\
&= 2\pi G \sup_{\|h\|_{\pi_x}=1} \left\| \int_{z\in\mathcal{X}} \left(\mathbb{E}\big[\zeta(x,\omega)\cdot\zeta(z,\omega)\big] - \frac{1}{P}\sum_{\ell=1}^{P}\zeta(x,\omega_\ell)\cdot\zeta(z,\omega_\ell) \right) \cdot h(z)d\pi_z \right\|_{\pi_x} \\
&= 2\pi G \cdot \big\|\Omega - \Omega_P\big\| \\
&\leq 2\pi G \cdot \big\|\Omega - \Omega_P\big\|_{HS}.
\end{aligned}$$

Finally, we bound $\big\|\Omega - \Omega_P\big\|$ using Proposition H.2. Using the notation $\zeta_{\omega_\ell} = \zeta(\cdot,\omega_\ell)$, we define the random variable $\Theta_\ell = \Omega - \zeta_{\omega_\ell}\otimes\zeta_{\omega_\ell}$, this implies that from the definition of $\Omega - \Omega_P$, we have from Lemma 1 in Rudi & Rosasco (2017)

$$\frac{1}{P}\sum_{\ell=1}^{P}\Theta_\ell = \Omega - \Omega_P \qquad \text{with} \qquad \mathbb{E}[\Theta_\ell] = 0 \quad \forall \ell \in [P].$$

Next, note that Θ_ℓ are random vectors belonging to hilbert space of Hilbert-Schmidt operators on $L_2(\mathcal{X},\pi_x)$. Therefore, we can apply Proposition H.2. Moreover, from the definition of Ω and Ω_P, $\|\Omega - \zeta_{\omega_\ell}\otimes\zeta_{\omega_\ell}\|_{HS} \leq 2$ and $\mathbb{E}\|\Theta_\ell\|_{HS}^2 \leq 1$. Therefore, from the application of Proposition H.2, we have with probability at least $1-\delta$

$$\|\Omega - \Omega_P\|_{HS} \leq \frac{4}{P}\log\frac{2}{\delta} + \sqrt{\frac{2}{P}\log\frac{2}{\delta}}.$$

Substituting in the expression above we get the result of statement (i).

The proof of (ii) follows from Lemma 9 in Rudi & Rosasco (2017). □

Combining bounds on term(A), term(B), term(C) and term(D) Here, we combine the results of Lemmas G.1, G.3, G.5 and G.6 to bound $\|\hat{f}_P - f_\pi\|_{\pi_x}$. We get the following theorem.

Theorem G.8. *For the two classes of kernels with $\lambda \leq \frac{3}{4}\|K\|$, the following is satisfied with probability at least $1-\delta$*

1. *For the GIP kernel, with*

$$N \geq \max\Big\{\frac{72\kappa^2}{\lambda}, \frac{2\lambda}{9}\Big\}\log\frac{32\kappa^2}{\lambda\cdot\delta}$$

and with

$$P \geq \max\Big\{8, \frac{512\pi^2G^2}{\|K\|^2}, \frac{288\pi^2G^2}{\lambda^2}\Big\}\log\frac{16}{\delta},$$

we have

$$\|\hat{f}_P - f_\pi\|_{\pi_x} \leq \sqrt{\frac{216R^2\cdot\kappa^4}{\lambda\cdot N}\log\frac{16}{\delta}} + \sqrt{\frac{32\pi^2R^2\cdot G^2}{\lambda\cdot P}\log\frac{16}{\delta}} + R\cdot\sqrt{\lambda}.$$

2. *For the RF kernel, with*

$$N \geq \max\Big\{\frac{72\kappa^2}{\lambda}, \frac{2\lambda}{9}\Big\}\log\frac{32\kappa^2}{\lambda\cdot\delta}$$

and with

$$P \geq \max\left\{8\kappa^2, \frac{32\kappa^2}{\|K\|^2}, \frac{2\lambda}{9}, \frac{72\kappa^2}{\lambda}, \frac{2(\lambda+\kappa^2)^2}{9\kappa^2}\right\}\log\frac{128\kappa^2}{\lambda\cdot\delta},$$

we have

$$\|\hat{f}_P - f_\pi\|_{\pi_x} \leq \sqrt{\frac{216R^2\cdot\kappa^4}{\lambda\cdot N}\log\frac{16}{\delta}} + \sqrt{\frac{32R^2\cdot\kappa^2}{P}\log\frac{128\kappa^2}{\lambda\cdot\delta}} + R\cdot\sqrt{\lambda}.$$

Finally, using the above Theorem G.8 and (33), we bound the generalization error.

Theorem G.9. *For the two classes of kernels with $\lambda \leq \frac{3}{4}\|K\|$, the following is satisfied with probability at least $1-\delta$*

1. *For the GIP kernel, with*

$$N \geq \max\left\{\frac{72\kappa^2}{\lambda}, \frac{2\lambda}{9}\right\}\log\frac{32\kappa^2}{\lambda\cdot\delta}$$

and with

$$P \geq \max\left\{8, \frac{512\pi^2G^2}{\|K\|^2}, \frac{288\pi^2G^2}{\lambda^2}\right\}\log\frac{16}{\delta},$$

we have

$$\mathcal{L}(f) - \inf_{h\in\mathcal{H}}\mathcal{L}(h) \leq \frac{648R^2\cdot\kappa^4}{\lambda\cdot N}\log\frac{16}{\delta} + \frac{96\pi^2R^2\cdot G^2}{\lambda\cdot P}\log\frac{16}{\delta} + 3R^2\cdot\lambda.$$

2. *For the RF kernel, with*

$$N \geq \max\left\{\frac{72\kappa^2}{\lambda}, \frac{2\lambda}{9}\right\}\log\frac{32\kappa^2}{\lambda\cdot\delta}$$

and with

$$P \geq \max\left\{8\kappa^2, \frac{32\kappa^2}{\|K\|^2}, \frac{2\lambda}{9}, \frac{72\kappa^2}{\lambda}, \frac{2(\lambda+\kappa^2)^2}{9\kappa^2}\right\}\log\frac{128\kappa^2}{\lambda\cdot\delta},$$

we have

$$\mathcal{L}(f) - \inf_{h\in\mathcal{H}}\mathcal{L}(h) \leq \frac{648R^2\cdot\kappa^4}{\lambda\cdot N}\log\frac{16}{\delta} + \frac{96R^2\cdot\kappa^2}{P}\log\frac{128\kappa^2}{\lambda\cdot\delta} + 3R^2\cdot\lambda.$$

Finally, we choose the parameters to get the result of Theorem 4.3.

Choosing parameters: Here, we optimally choose the parameters λ, P, N such that all the conditions in Theorem G.8 and G.9 are satisfied and the proposed approximation approach achieves the (minimax) optimal generalization performance.

- **GIP kernel:** For the GIP kernel, we choose the parameters as follows:
  - Regularization parameter: $\lambda = 1/\sqrt{N}$.
  - Number of overall samples:

$$N \geq \max\left\{\frac{4}{3\|K\|^2}, 72\kappa^2\cdot\sqrt{N}\cdot\log\frac{32\kappa^2\cdot\sqrt{N}}{\delta}\right\}$$

  - Number of bits communicated:

$$P \geq \max\left\{8, \frac{512\pi^2G^2}{\|K\|^2}, 288\pi^2G^2\cdot N\right\}\log\frac{16}{\delta}$$

- **RF kernel:** For the RF kernel we choose the parameters as follows:
  - Regularization parameter: $\lambda = 1/\sqrt{N}$.
  - Number of overall samples:

$$N \geq \max\left\{\frac{4}{3\|K\|^2}, 72\kappa^2\cdot\sqrt{N}\cdot\log\frac{32\kappa^2\cdot\sqrt{N}}{\delta}\right\}$$

– Number of real values communicated:

$$P \geq \max\left\{8\kappa^2, \frac{32\kappa^2}{\|K\|^2}, 72\kappa^2 \cdot \sqrt{N}\right\} \log \frac{128\kappa^2 \cdot \sqrt{N}}{\delta}$$

Now with the above choice of parameters we have Theorem 4.3 stated beolw.

Theorem G.10. *With the choice of parameters stated above, we have with probability at least* $1-\delta$ *for both the GIP and RF kernels that*

$$\mathcal{L}(f) - \inf_{h \in \mathcal{H}} \mathcal{L}(h) = \mathcal{O}\left(\frac{1}{\sqrt{N}}\right).$$

H CONCENTRATION INEQUALITIES

Proposition H.1. *For* $x_1, \ldots, x_N$*, a sequence of zero-mean i.i.d. random variables with* $x_i \in \mathbb{R}$ *for all* $i \in [N]$*. If there exists* $T, S \in \mathbb{R}$ *s.t.* $x_i \leq T$ *a.s. and* $\mathbb{E}x_i^2 \leq S$ *for all* $i \in [N]$*. Then with probability at least* $1-\delta$ *we have*

$$\frac{1}{N}\sum_{i=1}^{N} x_i \leq \frac{2T\log\frac{1}{\delta}}{3N} + \sqrt{\frac{2S\log\frac{1}{\delta}}{N}}.$$

Proposition H.2. *For* $x_1, \ldots, x_N$*, a sequence of i.i.d. random vectors on a separable Hilbert space* $\mathcal{H}$*. Assume* $\mu = \mathbb{E}x_i$ *and let* $\sigma, B \geq 0$ *s.t.*

$$\mathbb{E}\|x_i - \mu\|_{\mathcal{H}}^p \leq \frac{1}{2}p!\sigma^2 B^{p-2} \qquad \forall p \geq 2,$$

for any $i \in [N]$*. Then we have with probability at least* $1-\delta$

$$\left\|\frac{1}{N}\sum_{i=1}^{N} x_i - \mu\right\|_{\mathcal{H}} \leq \frac{2B\log\frac{2}{\delta}}{N} + \sqrt{\frac{2\sigma^2\log\frac{2}{\delta}}{N}}.$$

Proposition H.3. *Let* $\mathcal{F}$ *be a separable Hilbert space and let* $X_1, \ldots, X_N$ *be a sequence of i.i.d. self-adjoint positive random operators on* $\mathcal{F}$*. Assume that* $\mathbb{E}[X_i] = 0$ *and* $\lambda_{\max}(X_i) \leq T$ *a.s. for some* $T > 0$ *for all* $i \in [N]$*. Let* S *be such that* $\mathbb{E}(X_i)^2 \leq S$*. Then with probability at least* $1-\delta$ *we have*

$$\lambda_{\max}\left(\frac{1}{N}\sum_{i=1}^{N} X_i\right) \leq \frac{2T\beta}{3N} + \sqrt{\frac{2\|S\|\beta}{N}},$$

with $\beta = \log\frac{2TrS}{\|S\|\cdot\delta}$.

I SUFFICIENT CONDITION FOR POSITIVE DEFINITE GIP KERNEL

Assumption 6. We assume that there exists a mapping from $\{x_i, \ldots, x_N\} \mapsto \{\tilde{x}_i, \ldots, \tilde{x}_N\}$ with $x_i \in \mathbb{R}^d$ and $\tilde{x}_i \in \mathbb{R}^D$ for some $D \in \mathbb{N}$ such that the following is satisfied:

1. $\|\tilde{x}_i\| = \|x_i\|$ for all $i \in [N]$.
2. $\psi(\tilde{x}_i, \tilde{x}_j) = \psi_P(x_i, x_j)$ for all $i, j \in [N]$ where $\psi_P(x_i, x_j)$ is defined in (7).

The above assumption implies that there exists a mapping that transforms the feature vectors such that their norms are preserved and the pairwise angles between the transformed set of vectors are same as the approximated angles for the GIP kernel (cf. (7)).

Proposition I.1. *Under Assumption 6, the kernel* $\mathbf{K}_P$ *approximated using (6) for the GIP kernel is positive semi-definite.*

Proof. From the definition of the approximated GIP kernel (6) and Assumption 6, we have

$$k_P(x_i, x_j) = g(\psi_P(x_i, x_j), \|x_i\|, \|x_j\|) = g(\psi_P(\tilde{x}_i, \tilde{x}_j), \|\tilde{x}_i\|, \|x_j\|) = k(\tilde{x}_i, \tilde{x}_j).$$

Using the above fact, we have for the approximated GIP kernel for any vector $v \in \mathbb{R}^N$

$$\begin{aligned} v^T \mathbf{K}_P v &= \sum_{i,j=1}^{N} [v]_i [v]_j k_P(x_i, x_j) \\ &= \sum_{i,j=1}^{N} [v]_i [v]_j k(\tilde{x}_i, \tilde{x}_j) \\ &\overset{(i)}{=} \left\langle \sum_{i=1}^{N} [v]_i \phi(x_i), \sum_{j=1}^{N} [v]_j \phi(x_j) \right\rangle \\ &= \left\| \sum_{i=1}^{N} [v]_i \phi(x_i) \right\| \geq 0, \end{aligned}$$

where (i) follows from Mercer's theorem, i.e., for positive definite kernel k, there exists $\phi(\cdot)$ such that $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. Moreover, we used the fact that the GIP kernel k is well defined irrespective of the dimension of the feature vectors.

Therefore, we have the proof. □