ON THE RELATIVE ERROR OF RANDOM FOURIER FEATURES FOR PRESERVING KERNEL DISTANCE

Kuan Cheng
Peking University
Email: ckkcdh@pku.edu.cn

Shaofeng H.-C. Jiang
Peking University
Email: shaofeng.jiang@pku.edu.cn

Luojian Wei
Peking University
Email: luojianwei@pku.edu.cn

Zhide Wei
Peking University
Email: zhidewei@pku.edu.cn

ABSTRACT

The method of random Fourier features (RFF), proposed in a seminal paper by Rahimi and Recht (NIPS'07), is a powerful technique to find approximate low-dimensional representations of points in (high-dimensional) kernel space, for shift-invariant kernels. While RFF has been analyzed under various notions of error guarantee, the ability to preserve the kernel distance with *relative* error is less understood. We show that for a significant range of kernels, including the well-known Laplacian kernels, RFF cannot approximate the kernel distance with small relative error using low dimensions. We complement this by showing as long as the shift-invariant kernel is analytic, RFF with $\mathrm{poly}(\varepsilon^{-1}\log n)$ dimensions achieves ε-relative error for pairwise kernel distance of n points, and the dimension bound is improved to $\mathrm{poly}(\varepsilon^{-1}\log k)$ for the specific application of kernel k-means. Finally, going beyond RFF, we make the first step towards data-oblivious dimension-reduction for general shift-invariant kernels, and we obtain a similar $\mathrm{poly}(\varepsilon^{-1}\log n)$ dimension bound for Laplacian kernels. We also validate the dimension-error tradeoff of our methods on simulated datasets, and they demonstrate superior performance compared with other popular methods including random-projection and Nyström methods.

1 INTRODUCTION

We study the ability of the random Fourier features (RFF) method (Rahimi & Recht, 2007) for preserving the relative error for the *kernel* distance. Kernel method (Schölkopf & Smola, 2002) is a systematic way to map the input data into a (indefinitely) high dimensional feature space to introduce richer structures, such as non-linearity. In particular, for a set of n data points P, a kernel function $K : P \times P \to \mathbb{R}$ implicitly defines a *feature mapping* $\varphi : P \to \mathcal{H}$ to a feature space $\mathcal{H}$ which is a Hilbert space, such that $\forall x, y, K(x, y) = \langle \varphi(x), \varphi(y) \rangle$. Kernel methods have been successfully applied to classical machine learning (Boser et al., 1992; Schölkopf et al., 1998; Girolami, 2002), and it has been recently established that in a certain sense the behavior of neural networks may be modeled as a kernel (Jacot et al., 2018).

Despite the superior power and wide applicability, the scalability has been an outstanding issue of applying kernel methods. Specifically, the representation of data points in the feature space is only implicit, and solving for the explicit representation, which is crucially required in many algorithms, takes at least $\Omega(n^2)$ time in the worst case. While for many problems such as kernel SVM, it is possible to apply the so-called "kernel trick" to rewrite the objective in terms of $K(x, y)$, the explicit representation is still often preferred, since the representation is compatible with a larger range of solvers/algorithms which allows better efficiency.

In a seminal work (Rahimi & Recht, 2007), Rahimi and Recht addressed this issue by introducing the method of random Fourier features (see Section 2 for a detailed description), to compute an explicit low-dimensional mapping $\varphi' : P \to \mathbb{R}^D$ (for $D \ll n$) such that $\langle \varphi'(x), \varphi'(y) \rangle \approx \langle \varphi(x), \varphi(y) \rangle =$

$K(x, y)$, for shift-invariant kernels (i.e., there exists $K : P \to \mathbb{R}$, such that $K(x, y) = K(x - y)$) which includes widely-used Gaussian kernels, Cauchy kernels and Laplacian kernels.

Towards understanding this fundamental method of RFF, a long line of research has focused on analyzing the tradeoff between the target dimension D and the accuracy of approximating K under certain error measures. This includes additive error $\max_{x,y} |\langle \varphi(x), \varphi(y)\rangle - \langle \varphi'(x), \varphi'(y)\rangle|$ (Rahimi & Recht, 2007; Sriperumbudur & Szabó, 2015; Sutherland & Schneider, 2015), spectral error (Avron et al., 2017; Choromanski et al., 2018; Zhang et al., 2019; Erdélyi et al., 2020; Ahle et al., 2020), and the generalization error of several learning tasks such as kernel SVM and kernel ridge regression (Avron et al., 2017; Sun et al., 2018; Li et al., 2021). A more comprehensive overview of the study of RFF can be found in a recent survey (Liu et al., 2021).

We focus on analyzing RFF with respect to the *kernel distance*. Here, the kernel distance of two data points x, y is defined as their (Euclidean) distance in the feature space, i.e.,

$$\mathrm{dist}_\varphi(x, y) = \|\varphi(x) - \varphi(y)\|_2.$$

While previous results on the additive error of $K(x, y)$ (Rahimi & Recht, 2007; Sriperumbudur & Szabó, 2015; Sutherland & Schneider, 2015; Avron et al., 2017) readily implies additive error guarantee of $\mathrm{dist}_\varphi(x, y)$, the *relative* error guarantee is less understood. As far as we know, Chen & Phillips (2017) is the only previous work that gives a relative error bound for kernel distance, but unfortunately, only Gaussian kernel is studied in that work, and whether or not the kernel distance for other shift-invariant kernels is preserved by RFF, is still largely open.

In spirit, this multiplicative error guarantee of RFF, if indeed exists, makes it a kernelized version of Johnson-Lindenstrauss Lemma (Johnson & Lindenstrauss, 1984) which is one of the central result in dimension reduction. This guarantee is also very useful for downstream applications, since one can combine it directly with classical geometric algorithms such as k-means++ (Arthur & Vassilvitskii, 2007), locality sensitive hashing (Indyk & Motwani, 1998) and fast geometric matching algorithms (Raghvendra & Agarwal, 2020) to obtain very efficient algorithms for kernelized k-means clustering, nearest neighbor search, matching and many more.

1.1 Our Contributions

Our main results are characterizations of the kernel functions on which RFF preserves the kernel distance with small relative error using $\mathrm{poly}\log$ target dimensions. Furthermore, we also explore how to obtain data-oblivious dimension-reduction for kernels that cannot be handled by RFF.

As mentioned, it has been shown that RFF with small dimension preserves the *additive* error of kernel distance for all shift-invariant kernels (Rahimi & Recht, 2007; Sriperumbudur & Szabó, 2015; Sutherland & Schneider, 2015). In addition, it has been shown in Chen & Phillips (2017) that RFF indeed preserves the relative error of kernel distance for Gaussian kernels (which is shift-invariant). Hence, by analogue to the additive case and as informally claimed in Chen & Phillips (2017), one might be tempted to expect that RFF also preserves the relative error for general shift-invariant kernels as well.

Lower Bounds. Surprisingly, we show that this is *not* the case. In particular, we show that for a wide range of kernels, including the well-known Laplacian kernels, it requires unbounded target dimension for RFF to preserve the kernel distance with constant multiplicative error. We state the result for a Laplacian kernel in the following, and the full statement of the general conditions of kernels can be found in Theorem 4.1. In fact, what we show is a quantitatively stronger result, that if the input is (Δ, ρ)*-bounded*, then preserving any constant multiplicative error requires $\Omega(\mathrm{poly}(\Delta/\rho))$ target dimension. Here, a point $x \in \mathbb{R}^d$ is (Δ, ρ)-bounded if $\|x\|_\infty \le \Delta$ and $\min_{i:x_i \neq 0} |x_i| \ge \rho$, i.e., the magnitude is (upper) bounded by Δ and the resolution is (lower) bounded by ρ.

Theorem 1.1 (Lower bound; see Remark 4.1)**.** *For every $\Delta \ge \rho > 0$ and some feature mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$ of a Laplacian kernel $K(x, y) = \exp(-\|x - y\|_1)$, if for every $x, y \in \mathbb{R}^d$ that are (Δ, ρ)-bounded, the RFF mapping π for K with target dimension D satisfies $\mathrm{dist}_\pi(x, y) \in (1 \pm \varepsilon) \cdot \mathrm{dist}_\varphi(x, y)$ with constant probability, then $D \ge \Omega(\frac{1}{\varepsilon^2}\frac{\Delta}{\rho})$. This holds even when $d = 1$.*

Upper Bounds. Complementing the lower bound, we show that RFF can indeed preserve the kernel distance within $1 \pm \varepsilon$ error using $\mathrm{poly}(\varepsilon^{-1} \log n)$ target dimensions with high probability,

as long as the kernel function is shift-invariant and analytic, which includes Gaussian kernels and Cauchy kernels. Our target dimension nearly matches (up to the degree of polynomial of parameters) that is achievable by the Johnson-Lindenstrauss transform (Johnson & Lindenstrauss, 1984), which is shown to be tight (Larsen & Nelson, 2017). This upper bound also greatly generalizes the result of Chen & Phillips (2017) which only works for Gaussian kernels (see Section G for a detailed comparison).

Theorem 1.2 (Upper bound)**.** *Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a kernel function which is shift-invariant and analytic at the origin, with feature mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$ for some feature space $\mathcal{H}$. For every $0 < \delta \leq \varepsilon \leq 2^{-16}$, every $d, D \in \mathbb{N}$, $D \geq \max\{\Theta(\varepsilon^{-1}\log^3(1/\delta)), \Theta(\varepsilon^{-2}\log(1/\delta))\}$, if $\pi : \mathbb{R}^d \to \mathbb{R}^D$ is an RFF mapping for K with target dimension D, then for every $x, y \in \mathbb{R}^d$,*

$$\Pr[|\operatorname{dist}_\pi(x,y) - \operatorname{dist}_\varphi(x,y)| \leq \varepsilon \cdot \operatorname{dist}_\varphi(x,y)] \geq 1 - \delta.$$

The technical core of our analysis is a moment bound for RFF, which is derived by analysis techniques such as Taylor expansion and Cauchy's integral formula for multi-variate functions. The moment bound is slightly weaker than the moment bound of Gaussian variables, and this is the primary reason that we obtain a bound weaker than that of the Johnson-Lindenstrauss transform. Finally, several additional steps are required to fit this moment bound in Bernstein's inequality, which implies the bound in Theorem 1.2.

Improved Dimension Bound for Kernel k-Means. We show that if we focus on a specific application of kernel k-means, then it suffices to set the target dimension $D = \operatorname{poly}(\varepsilon^{-1}\log k)$, instead of $D = \operatorname{poly}(\varepsilon^{-1}\log n)$, to preserve the kernel k-means clustering cost for every k-partition. This follows from the probabilistic guarantee of RFF in Theorem 1.2 plus a generalization of the dimension-reduction result proved in a recent paper (Makarychev et al., 2019). Here, given a data set $P \subset \mathbb{R}^d$ and a kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, denoting the feature mapping as $\varphi : \mathbb{R}^d \to \mathcal{H}$, the kernel k-means problem asks to find a k-partition $\mathcal{C} := \{C_1, \ldots, C_k\}$ of P, such that $\operatorname{cost}^\varphi(P, \mathcal{C}) = \sum_{i=1}^k \min_{c_i \in \mathcal{H}} \sum_{x \in C_i} \|\varphi(x) - c_i\|_2^2$ is minimized.

Theorem 1.3 (Dimension reduction for clustering; see Theorem 3.1)**.** *For kernel k-means problem whose kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is shift-invariant and analytic at the origin, for every data set $P \subset \mathbb{R}^d$, the RFF mapping $\pi : \mathbb{R}^d \to \mathbb{R}^D$ with target dimension $D \geq O(\frac{1}{\varepsilon^2}(\log^3\frac{k}{\delta} + \log^3\frac{1}{\varepsilon}))$, with probability at least $1 - \delta$, preserves the clustering cost within $1 \pm \varepsilon$ error for every k-partition simultaneously.*

Applying RFF to speed up kernel k-means has also been considered in Chitta et al. (2012), but their error bound is much weaker than ours (and theirs is not a generic dimension-reduction bound). Also, similar dimension-reduction bounds (i.e., independent of n) for kernel k-means were obtained using Nyström methods (Musco & Musco, 2017; Wang et al., 2019), but their bound is $\operatorname{poly}(k)$ which is worse than our $\operatorname{poly}\log(k)$; furthermore, our RFF-based approach is unique in that it is data-oblivious, which enables great applicability in other relevant computational settings such as streaming and distributed computing.

Going beyond RFF. Finally, even though we have proved RFF cannot preserve the kernel distance for every shift-invariant kernels, it does not rule out the existence of other efficient data-oblivious dimension reduction methods for those kernels, particularly for Laplacian kernel which is the primary example in our lower bound. For instance, in the same paper where RFF was proposed, Rahimi and Recht (Rahimi & Recht, 2007) also considered an alternative embedding called "binning features" that can work for Laplacian kernels. Unfortunately, to achieve a relative error of ε, it requires a dimension that depends *linearly* on the magnitude/aspect-ratio of the dataset, which may be exponential in the input size. Follow-up works, such as (Backurs et al., 2019), also suffer similar issues.

We make the first successful attempt towards this direction, and we show that Laplacian kernels do admit an efficient data-oblivious dimension reduction. Here, we focus on the (Δ, ρ)-bounded case, Here, we use a similar setting to our lower bound (Theorem 1.1) where we focus on the (Δ, ρ)-bounded case.

Theorem 1.4 (Oblivious dimension-reduction for Laplacian kernels, see Theorem F.1)**.** *Let K be a Laplacian kernel, and denote its feature mapping as $\varphi : \mathbb{R}^d \to \mathcal{H}$. For every $D \geq \max\{\Theta(\varepsilon^{-1}\log^3(1/\delta)), \Theta(\varepsilon^{-2}\log(1/\delta))\}$, every $\Delta \geq \rho > 0$, there is a mapping $\pi : \mathbb{R}^d \to \mathbb{R}^D$,*

such that for every $0 < \delta \leq \varepsilon \leq 2^{-16}$, $\varepsilon_g > 0$ *and* $x, y \in \mathbb{R}^d$ *that are* (Δ, ρ)*-bounded, it holds that*

$$\Pr[|\operatorname{dist}_\pi(x, y) - \operatorname{dist}_\varphi(x, y)| \leq \varepsilon \cdot \operatorname{dist}_\varphi(x, y)] \geq 1 - \delta - \varepsilon_g.$$

The time for evaluating π *is* $dD \cdot \operatorname{poly}(\log \frac{\Delta}{\rho} \log \varepsilon_g^{-1})$.

Our target dimension only depends on $\log \frac{\Delta}{\rho}$ which may be interpreted as the precision of the input. Hence, as an immediate corollary, for any n-points dataset with precision $1/\operatorname{poly}(n)$, we have an embedding with target dimension $D = \operatorname{poly}(\varepsilon^{-1} \log n)$, where the success probability is $1 - 1/\operatorname{poly}(n)$ and the overall running time of embedding the n points is $O(n \operatorname{poly}(d\varepsilon^{-1} \log n))$.

Our proof relies on the fact that every ℓ_1 metric space can be embedded into a squared ℓ_2 metric space isometrically. We explicitly implement an approximate version of this embedding (Kahane, 1981), and eventually reduce our problem of Laplacian kernels to Gaussian kernels. After this reduction, we use the RFF for Gaussian kernels to obtain the final mapping. However, since the embedding to squared ℓ_2 is only of very high dimension, to implement this whole idea efficiently, we need to utilize the special structures of the embedding, combined with an application of space bounded pseudo-random generators (PRGs) (Nisan, 1992).

Even though our algorithm utilizes the special property of Laplacian kernels and eventually still partially use the RFF for Gaussian kernels, it is still of conceptual importance. It opens up the direction of exploring general methods for Johnson-Lindenstrauss style dimension reduction for shift-invariant kernels. Furthermore, the lower bound suggests that the Johnson-Lindenstrauss style dimension reduction for general shift-invariant kernels has to be *not* differentiable, which is a fundamental difference to RFF. This requirement of "not analytical" seems very counter-intuitive, but our construction of the mapping for Laplacian kernels indeed provides valuable insights on how the non-analytical mapping behaves.

Experiments and Comparison to Other Methods. Apart from RFF, the Nyström and the random-projection methods are alternative popular methods for kernel dimension reduction. In Section 6, we conduct experiments to compare their empirical dimension-error tradeoffs with that of our methods on a simulated dataset. Since we focus on the error, we use the "ideal" implementation of both methods that achieve the best accuracy, so they are only in favor of the two baselines – for Nyström, we use SVD on the kernel matrix, since Nyström methods can be viewed as fast and approximate low-rank approximations to the kernel matrix; for random-projection, we apply the Johnson-Lindenstrauss transform on the explicit representations of points in the feature space. We run two experiments to compare each of RFF (on a Gaussian kernel) and our new algorithm in Theorem 1.4 (on a Laplacian kernel) with the two baselines respectively. Our experiments indicate that the Nyström method is indeed incapable of preserving the kernel distance in relative error, and more interestingly, our methods perform the best among the three, even better than the Johnson-Lindenstrauss transform which is the optimal in the worst case.

1.2 Related Work

Variants of the vanilla RFF, particularly those that use information in the input data set and/or sample random features non-uniformly, have also been considered, including leverage score sampling random Fourier features (LSS-RFF) (Rudi et al., 2018; Liu et al., 2020; Erdélyi et al., 2020; Li et al., 2021), weighted random features (Rahimi & Recht, 2008; Avron et al., 2016; Chang et al., 2017; Dao et al., 2017), and kernel alignment (Shahrampour et al., 2018; Zhen et al., 2020).

The RFF-based methods usually work for shift-invariant kernels only. For general kernels, techniques that are based on low-rank approximation of the kernel matrix, notably Nyström method (Williams & Seeger, 2000; Gittens & Mahoney, 2016; Musco & Musco, 2017; Oglic & Gärtner, 2017; Wang et al., 2019) and incomplete Cholesky factorization (Fine & Scheinberg, 2001; Bach & Jordan, 2002; Chen et al., 2021; Jia et al., 2021)) were developed. Moreover, specific sketching techniques were known for polynomial kernels (Avron et al., 2014; Woodruff & Zandieh, 2020; Ahle et al., 2020; Song et al., 2021), a basic type of kernel that is not shift-invariant.

2 PRELIMINARIES

Random Fourier Features. RFF was first introduced by Rahimi and Recht (Rahimi & Recht, 2007). It is based on the fact that, for shift-invariant kernel $K : \mathbb{R}^d \to \mathbb{R}$ such that $K(0) = 1$ (this can be assumed w.l.o.g. by normalization), function $p : \mathbb{R}^d \to \mathbb{R}$ such that $p(\omega) = \frac{1}{2\pi}\int_{\mathbb{R}^d} K(x)e^{-\mathrm{i}\langle\omega,x\rangle}\,\mathrm{d}x$, which is the Fourier transform of $K(\cdot)$, is a probability distribution (guaranteed by Bochner's theorem (Bochner, 1933; Rudin, 1991)). Then, the RFF mapping is defined as

$$\pi(x) := \sqrt{\frac{1}{D}}\begin{pmatrix}\sin\langle\omega_1,x\rangle\\ \cos\langle\omega_1,x\rangle\\ \vdots\\ \sin\langle\omega_D,x\rangle\\ \cos\langle\omega_D,x\rangle\end{pmatrix}$$

where $\omega_1,\omega_2,\ldots,\omega_D \in \mathbb{R}^d$ are i.i.d. samples from distribution with densitiy p.

Theorem 2.1 (Rahimi & Recht 2007)**.** $\mathbb{E}[\langle\pi(x),\pi(y)\rangle] = \frac{1}{D}\sum_{i=1}^{D}\mathbb{E}[\cos\langle\omega_i,x-y\rangle] = K(x-y)$.

Fact 2.1. *Let ω be a random variable with distribution p over $\mathbb{R}^d$. Then*

$$\forall t\in\mathbb{R},\quad \mathbb{E}[\cos(t\langle\omega,x-y\rangle)] = \Re\int_{\mathbb{R}^d} p(\omega)e^{\mathrm{i}\langle\omega,t(x-y)\rangle}\,\mathrm{d}\omega = K(t(x-y)), \tag{1}$$

and $\mathrm{Var}(\cos\langle\omega,x-y\rangle) = \frac{1+K(2(x-y))-2K(x-y)^2}{2}$.

3 UPPER BOUNDS

We present two results in this section. We start with Section 3.1 to show RFF preserves the relative error of kernel distance using $\mathrm{poly}(\varepsilon^{-1}\log n)$ target dimensions with high probability, when the kernel function is shift-invariant and analytic at origin. Then in Section 3.2, combining this bound with a generalized analysis from a recent paper (Makarychev et al., 2019), we show that RFF also preserves the clustering cost for kernel k-clustering problems with ℓ_p-objective, with target dimension only $\mathrm{poly}(\varepsilon^{-1}\log k)$ which is independent of n.

3.1 PROOF OF THEOREM 1.2: THE RELATIVE ERROR FOR PRESERVING KERNEL DISTANCE

Since K is shift-invariant, we interpret K as a function on $\mathbb{R}^d$ instead of $\mathbb{R}^d\times\mathbb{R}^d$, such that $K(x,y) = K(x-y)$. As in Section 2, let $p : \mathbb{R}^d \to \mathbb{R}$ be the Fourier transform of K, and suppose in the RFF mapping π, the random variables $\omega_1,\ldots,\omega_d \in \mathbb{R}^d$ are i.i.d. sampled from the distribution with density p. When we say K is analytic at the origin, we mean there exists some constant r s.t. K is analytic in $\{x\in\mathbb{R}^d : \|x\|_1 < r\}$. We pick r_K to be the maximum of such constant r. Also notice that in $D \geq \max\{\Theta(\varepsilon^{-1}\log^3(1/\delta)),\Theta(\varepsilon^{-2}\log(1/\delta))\}$, there are constants about K hidden inside the Θ, i.e. R_K as in Lemma 3.2.

Fact 3.1. *The following holds.*

- $\mathrm{dist}_\pi(x,y) = \sqrt{2-2/D\sum_{i=1}^{D}\cos\langle\omega_i,x-y\rangle}$*, and* $\mathrm{dist}_\varphi(x,y) = \sqrt{2-2K(x-y)}$.
- $\Pr[|\,\mathrm{dist}_\pi(x,y)-\mathrm{dist}_\varphi(x,y)| \leq \varepsilon\cdot\mathrm{dist}_\varphi(x,y)] \geq \Pr[|\,\mathrm{dist}_\pi(x,y)^2-\mathrm{dist}_\varphi(x,y)^2| \leq \varepsilon\cdot\mathrm{dist}_\varphi(x,y)^2]$.

Define $X_i(x) := \cos\langle\omega_i,x\rangle - K(x)$. As a crucial step, we next analyze the moment of random variables $X_i(x-y)$. This bound will be plugged into Bernstein's inequality to conclude the proof.

Lemma 3.1. *If for some $r>0$, K is analytic in $\{x\in\mathbb{R}^d : \|x\|_1 < r\}$, then for every $k\geq 1$ being even and every x s.t. $\|x\|_1 < r$, we have* $\mathbb{E}[|X_i(x)|^k] \leq \left(\frac{4k\|x\|_1}{r}\right)^{2k}$.

Proof. The proof can be found in Section A. □

Lemma 3.2. *For kernel K which is shift-invariant and analytic at the origin, there exist $c_K, R_K > 0$ such that for all $\|x\|_1 \le R_K$, $\frac{1-K(x)}{\|x\|_1^2} \ge \frac{c_K}{2}$.*

Proof. The proof can be found in Section B. □

Proof sketch of Theorem 1.2. We present a proof sketch for Theorem 1.2, and the full proof can be found in Section C. We focus on the case when $\|x-y\|_1 \le R_K$ (the other case can be found in the full proof). Then by Lemma 3.2, we have $2 - 2K(x-y) \ge c\|x-y\|_1^2$. Then we have:

$$\Pr\left[\left|\frac{2}{D}\sum_{i=1}^{D} X_i(x-y)\right| \le \varepsilon \cdot (2 - 2K(x-y))\right] \ge \Pr\left[\left|\frac{2}{D}\sum_{i=1}^{D} X_i(x-y)\right| \le c\varepsilon \cdot \|x-y\|_1^2\right].$$

We take $r = r_K$ for simplicity of exhibition. Assume $\delta \le \min\{\varepsilon, 2^{-16}\}$, let $k = \log(2D^2/\delta), t = 64k^2/r^2$, is even. By Markov's inequality and Lemma 3.1:

$$\Pr[|X_i(x-y)| \ge t\|x-y\|_1^2] = \Pr\left[|X_i(x-y)|^k \ge t^k\|x-y\|_1^{2k}\right] \le \frac{(4k)^{2k}}{t^k r^{2k}} = 4^{-k} \le \frac{\delta}{2D^2}.$$

For simplicity denote $X_i(x-y)$ by X_i, $\|x-y\|_1^2$ by ℓ and define $X_i' = \mathbb{1}_{[|X_i| \ge t\ell]} t\ell \cdot \operatorname{sgn}(X_i) + \mathbb{1}_{[|X_i| < t\ell]} X_i$, note that $\mathbb{E}[X_i] = 0$. By some further calculations and plugging in the parameters t, δ, D, we can eventually obtain $\mathbb{E}[|X_i'|] \le \delta\ell$. Denote σ'^2 as the variance of X_i', then again by Lemma 3.1 we immediately have $\sigma' \le 64\ell/r^2$. The theorem follows by a straightforward application of Bernstein's inequality. □

3.2 Dimension Reduction for Kernel Clustering

We present the formal statement for Theorem 1.3 in Theorem 3.1. In fact, we consider the more general k-clustering problem with ℓ_2^p-objective defined in the following Definition 3.1, which generalizes kernel k-means (by setting $p = 2$).

Definition 3.1. Given a data set $P \subset \mathbb{R}^d$ and kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, denoting the feature mapping as $\varphi : \mathbb{R}^d \to \mathcal{H}$, the kernel k-clustering problem with ℓ_p-objective asks for a k-partition $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ of P that minimizes the cost function: $\operatorname{cost}_p^{\varphi}(P, \mathcal{C}) := \sum_{i=1}^{k} \min_{c_i \in \mathcal{H}} \sum_{x \in C_i} \|\varphi(x) - c_i\|_2^p$.

Theorem 3.1 (Generalization of Makarychev et al. 2019, Theorem 3.6)**.** *For kernel k-clustering problem with ℓ_2^p-objective whose kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is shift-invariant and analytic at the origin, for every data set $P \subset \mathbb{R}^d$, the RFF mapping $\pi : \mathbb{R}^d \to \mathbb{R}^D$ with target dimension $D = \Omega(p^2 \log^3 \frac{k}{\alpha} + p^5 \log^3 \frac{1}{\varepsilon} + p^8)/\varepsilon^2$ satisfies*

$$\Pr[\forall k\text{-partition } \mathcal{C} \text{ of } P : \operatorname{cost}_p^{\pi}(P, \mathcal{C}) \in (1 \pm \varepsilon) \cdot \operatorname{cost}_p^{\varphi}(P, \mathcal{C})] \ge 1 - \delta.$$

Proof. The proof can be found in Section D. □

4 Lower Bounds

Theorem 4.1. *Consider $\Delta \ge \rho > 0$, and a shift-invariant kernel function $K : \mathbb{R}^d \to \mathbb{R}$, denoting its feature mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$. Then there exists $x, y \in \mathbb{R}^d$ that are (Δ, ρ)-bounded, such that for every $0 < \varepsilon < 1$, the RFF mapping π for K with target dimension D satisfies*

$$\Pr[|\operatorname{dist}_{\varphi}(x,y) - \operatorname{dist}_{\pi}(x,y)| \ge \varepsilon \cdot \operatorname{dist}_{\varphi}(x,y)] \ge \frac{2}{\sqrt{2\pi}} \int_{6\varepsilon\sqrt{D/s_K^{(\Delta,\rho)}}}^{\infty} e^{-s^2/2}\, \mathrm{d}s - O\left(D^{-\frac{1}{2}}\right) \quad (2)$$

where $s_K^{(\Delta,\rho)} := \sup_{(\Delta,\rho)\text{-bounded } x \in \mathbb{R}^d} s_K(x)$, and $s_K(x) := \dfrac{1 + K(2x) - 2K(x)^2}{2(1-K(x))^2}$.

Proof. The proof can be found in Section E. □

Note that the right hand side of (2) is always less than 1, since the first term $\frac{2}{\sqrt{2\pi}}\int_{6\varepsilon\sqrt{D/s_K^{(\Delta,\rho)}}}^{\infty} e^{-s^2/2}\,\mathrm{d}s$ achieves its maximum at $\frac{D\varepsilon^2}{s_K^{(\Delta,\rho)}} = 0$, and this maximum is 1. On the other hand, we need the right hand side of (2) to be > 0 in order to obtain a useful lower bound, and a typical setup to achieve this is when $D = \Theta\left(s_K^{(\Delta,\rho)}\right)$.

Intuition of s_K. Observe that $s_K(x)$ measures the ratio between the variance of RFF and the (squared) expectation evaluated at x. The intuition of considering this comes from the central limit theorem. Indeed, when the number of samples/target dimension is sufficiently large, the error/difference behaves like a Gaussian distribution where with constant probability the error $\approx \mathrm{Var}$. Hence, this s_K measures the "typical" relative error when the target dimension is sufficiently large, and an upper bound of $s_K^{(\Delta,\rho)}$ is naturally a necessary condition for the bounded relative error. The following gives a simple (sufficient) condition for kernels that do not have a bounded $s_K(x)$.

Remark 4.1 (Simple sufficient conditions for lower bounds). Assume the input dimension is 1, so $K : \mathbb{R} \to \mathbb{R}$, and assume $\Delta = 1$, $\rho < 1$. Then the (Δ, ρ)-bounded property simply requires $\rho \le |x| \le 1$. We claim that, if K's first derivative at 0 is non-zero, i.e., $K'(0) \neq 0$, then RFF cannot preserve relative error for such K. To see this, we use Taylor's expansion for K at the origin, and simply use the approximation to degree one, i.e., $K(x) \approx 1 + ax$ (noting that $x \le 1$ so this is a good approximation), where $a = K'(0)$. Then

$$s_K(x) = \frac{1 + 1 + 2ax - 2(1 + ax)^2}{2a^2x^2} = -1 - \frac{1}{ax}.$$

So if $a = K'(0) \neq 0$, then for sufficiently small ρ and $|x| \ge \rho$, $s_K(\rho) \ge \Omega(1/\rho)$. This also implies the claim in Theorem 1.1 for Laplacian kernels (even though one needs to slightly modify this analysis since strictly speaking K' is not well defined at 0 for Laplacian kernels). As a sanity check, for shift-invariant kernels that are analytic at the origin (which include Gaussian kernels), it is necessary that $K'(0) = 0$.

5 Beyond RFF: Oblivious Embedding for Laplacian Kernel

In this section we provide a proof sketch for theorem 1.4. A more detailed proof is in section F.

Embedding To handle a Laplacian kernel function $K(x, y) = e^{-\frac{\|x-y\|_1}{c}}$ with some constant c, we cannot directly use the RFF mapping ϕ since our lower bound shows that the output dimension has to be very large when K is not analytical around the origin. To overcome this issue, we come up with the following idea. Notice that $L(x, y)$ relies on the ℓ_1-distance between x, y. If one can embed (embedding function f) the data points from the original ℓ_1 metric space to a new metric space and ensure that there is an analytical kernel function K' for the new space s.t. $K(x, y) = K'(f(x), f(y))$ for every pair of original data points x, y, then one can use the function composition $\phi \circ f$ to get a desired mapping.

Indeed, we find that ℓ_1 can be embedded to ℓ_2^2 isometrically (Kahane, 1981) in the following way. Here for simplicity of exhibition we only handle the case where input data are from $\mathbb{N}^d$. Notice that even though input data points are only consisted of integers, the mapping construction needs to handle fractions, as we will later consider some numbers generated from Gaussian distributions or numbers computed in the RFF mapping. So we first setup two numbers, $\Delta' = \mathrm{poly}(N)$ large enough and $\rho' = 1/\mathrm{poly}(N)$ small enough. All our following operations are working on numbers that are (Δ', ρ')-bounded. For each dimension we do the following transformation. Let $\pi_1 : \mathbb{N} \to \mathbb{R}^N$ be such that for every $x \in \mathbb{N}, x \le N$, $\pi_1(x)[j] = 1$, if $j \in [1, x]$ and $\pi_1(x)[j] = 0$ otherwise. Then consider all dimensions. The embedding function $\pi_1^{(d)} : \mathbb{N}^d \to \mathbb{R}^{Nd}$ be such that for every $x \in \mathbb{N}^d, x_i \le N, \forall i \in [d]$, we have $\pi_1^{(d)}(x)$ being the concatenation of d vectors $\pi_1(x_i), i \in [d]$. After embedding, consider a new kernel function $K' = e^{-\frac{\|x'-y'\|_2^2}{c}}$, where $x' = \pi_1^{(d)}(x), y' = \pi_1^{(d)}(y)$. One can see immediately that $K'(x', y') = K(x, y)$. Hence, we can apply RFF then, i.e. the mapping is $\phi \circ \pi_1^{(d)}$, which has a small output dimension. Detailed proofs can be seen in section F.1.

However, there is another issue. In our setting, if the data is (Δ, ρ) bounded, then we have to pick $N = O(\frac{\Delta}{\rho})$. The computing time has a linear factor in N, which is too large.

Polynomial Time Construction To reduce computing time, we start from the following observation about the RFF mapping $\phi(x')$. Each output dimension is actually a function of $\langle \omega, x' \rangle$, where ω is a vector of i.i.d Gaussian random variables. For simplicity of description we only consider that x has only one dimension and $x' = \pi_1(x)$. So x' is just a vector consists of x number of 1's entries starting from the left and then all the remaining entries are 0's. Notice that a summation of Gaussian random variables is still a Gaussian. So given x, one can generate $\langle \omega, x' \rangle$ according to the summation of Gaussians. But here comes another problem. For two data points x, y, we need to use the same ω. So if we generate $\langle \omega, x' \rangle$ and then $\langle \omega', y' \rangle$, then ω, ω' are independent.

To bypass this issue, first consider the following alternate way to generate $\langle \omega, x' \rangle$. Let h be the smallest integer s.t. $N \leq 2^h$. Consider a binary tree where each node has exactly 2 children. The depth is h. So it has exactly $2^h \geq N$ leaf nodes in the last layer. For each node v, we attach a random variable α_v in the following way. For the root, we attach a Gaussian variable which is the summation of 2^h independent Gaussian variable with distribution ω_0. Then we proceed layer by layer from the root to leaves. For each u, v being children of a common parent w, assume that α_w is the summation of 2^l independent ω_0 distributions. Then let α_u be the summation of the first 2^{l-1} distributions among them and α_v the summation of the second 2^{l-1} distributions. That is $\alpha_w = \alpha_u + \alpha_v$ with α_u, α_v being independent. Notice that conditioned on $\alpha_w = a$, then α_u takes the value b with probability $\Pr_{\alpha_u, \alpha_v \text{ i.i.d.}} [\alpha_u = b \mid \alpha_u + \alpha_v = a]$. α_v takes the value $a - b$ when α_u takes value b.

The randomness for generating every random variable corresponding to a node, are presented as a sequence, in the order from root to leaves, layer by layer, from left to right. We define α^x to be the summation of the random variables corresponding to the first x leaves. Notice that α^x can be sampled efficiently in the following way. Consider the path from the root to the x-th leaf. First we sample the root, which can be computed using the corresponding randomness. We use a variable z to record this sample outcome, calling z a accumulator for convenience. Then we visit each node along the path. When visiting v, assume its parent is w, where α_w has already been sampled previously with outcome a. If v is a left child of w, then we sample α_v conditioned on $\alpha_w = a$. Assume this sampling has outcome b. Then we add $-a + b$ to the current accumulator z. If v is a right child of a node w, then we keep the current accumulator z unchanged. After visiting all nodes in the path, z is the sample outcome for α^x. We can show that the joint distribution $\alpha^x, x = 0, 1, \ldots, N$ has the same distribution as $\langle \omega, \pi_1(x) \rangle, x = 0, 1, \ldots, N$. See lemma F.2.

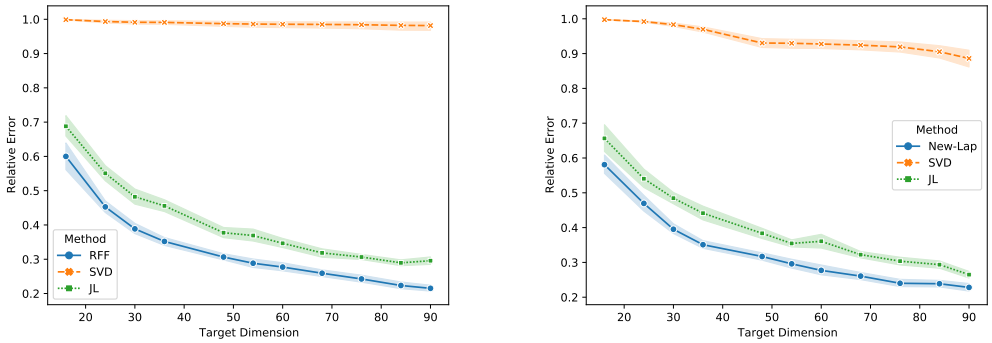
The advantage of this alternate construction is that given any x, to generate α^x, one only needs to visit the path from the root to the x-th leaf, using the above generating procedure. To finally reduce the time complexity, the last issue is that the uniform random string for generating random variables here is very long. If we sweep the random tape to locate the randomness used to generate a variable corresponding to a node, then we still need a linear time of N. Fortunately, PRGs for space bounded computation, e.g. Nisan's PRG (Nisan, 1992), can be used here to replace the uniform randomness. Because the whole procedure for deciding if $\|\phi \circ \pi_1(x) - \phi \circ \pi_1(y)\|_2$ approximates $\sqrt{2 - K(x, y)}$ within $(1 \pm \varepsilon)$ multiplicative error, is in poly-logarithmic space. Also the computation of such PRGs can be highly efficient, i.e. given any index of its output, one can compute that bit in time polynomial of the seed length, which is poly-logarithmic of N. Hence the computing time of the mapping only has a factor poly-logarithmic in N instead of a factor linear in N.

Now we have shown our construction for the case that all input data points are from $\mathbb{N}^d$. One can generalize this to the case where all numbers are (Δ, ρ) bounded, by doing some simple roundings and shiftings of numbers. More details of this part are deferred to section F.4.

6 EXPERIMENTS

We evaluate the empirical relative error of our methods on a simulated dataset. Specifically, we do two experiments, one to evaluate RFF on a Gaussian kernel, and the other one to evaluate the new algorithm in Theorem F.1, which we call "New-Lap", on a Laplacian kernel. In each experiment, we compare against two other popular methods, particularly Nyström and random-projection methods.

Baselines. Observe that there are many possible implementations of these two methods. However, since we focus on the accuracy evaluation, we choose computationally-heavy but more accurate implementations as the two baselines (hence the evaluation of the error is only in the baseline's favor). In particular, we consider 1) SVD low-rank approximation which we call "SVD", and 2) the



(a) RFF

(b) New-Lap

Figure 1: The dimension-error tradeoff curves for both experiments, i.e., the experiment that evaluates RFF and the one that evaluates New-Lap.

vanilla Johnson-Lindenstrauss algorithm performed on top of the high-dimensional representation of points in the feature space, which we call "JL". Note that SVD is the "ideal" goal/form of Nyström methods and that Johnson-Lindenstrauss applied on the feature space can obtain a theoretically-tight target-dimension bound (in the worst-case sense).

Experiment Setup. Both experiments are conducted on a synthesized dataset X which consists of $N = 100$ points with $d = 60$ dimensions generated i.i.d. from a Gaussian distribution. For the experiment that we evaluate RFF, we use a Gaussian kernel $K(x) = \exp(-0.5 \cdot \|x\|^2)$, and for that we evaluate New-Lap, we use a Laplacian kernel $K(x) = \exp(-0.5 \cdot \|x\|_1)$. In each experiment, for each method, we run it for varying target dimension D (for SVD, D is the target rank), and we report its *empirical* relative error, which is defined as

$$\max_{x \neq y \in X} \frac{|d'(x,y) - d_K(x,y)|}{d_K(x,y)},$$

where d_K is the kernel distance and d' is the approximated distance. To make the result stabilized, we conduct this entire experiment for every D for $T = 20$ times and report the average and 95% confident interval. We plot these dimension-error tradeoff curves, and we depict the results in Figure 1.

Results. We conclude that in both experiments, our methods can indeed well preserve the relative error of the kernel distance, which verifies our theorem. In particular, the dimension-error curve is comparable (and even slightly better) to the computationally heavy Johnson-Lindenstrauss algorithm (which is theoretically optimal in the worst case). On the contrary, the popular Nyström (low-rank approximation) method is largely incapable of preserving the relative error of the kernel distance. In fact, we observe that $d'_{SVD}(x,y) = 0$ or ≈ 0 often happens for some pairs of (x,y) such that $d(x,y) \neq 0$, which explains the high relative error. This indicates that our methods can indeed well preserve the kernel distance in relative error, but existing methods struggle to achieve this.

Acknowledgments

Research is partially supported by a national key R&D program of China No. 2021YFA1000900, startup funds from Peking University, and the Advanced Institute of Information Technology, Peking University.

References

Thomas D. Ahle, Michael Kapralov, Jakob Bæk Tejs Knudsen, Rasmus Pagh, Ameya Velingker, David P. Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. In *SODA*, pp. 141–160. SIAM, 2020.

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA*, pp. 1027–1035. SIAM, 2007.

Haim Avron, Huy L. Nguyen, and David P. Woodruff. Subspace embeddings for the polynomial kernel. In *NIPS*, pp. 2258–2266, 2014.

Haim Avron, Vikas Sindhwani, Jiyan Yang, and Michael W. Mahoney. Quasi-monte carlo feature maps for shift-invariant kernels. *J. Mach. Learn. Res.*, 17:120:1–120:38, 2016.

Haim Avron, Michael Kapralov, Cameron Musco, Christopher Musco, Ameya Velingker, and Amir Zandieh. Random fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 253–262. PMLR, 2017.

Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *J. Mach. Learn. Res.*, 3:1–48, 2002.

Arturs Backurs, Piotr Indyk, and Tal Wagner. Space and time efficient kernel density estimation in high dimensions. In *NeurIPS*, pp. 15773–15782, 2019.

Andrew C Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the american mathematical society*, 49(1):122–136, 1941.

Salomon Bochner. Monotone funktionen, stieltjessche integrale und harmonische analyse. *Mathematische Annalen*, 108(1):378–410, 1933.

Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, pp. 144–152. ACM, 1992.

Wei-Cheng Chang, Chun-Liang Li, Yiming Yang, and Barnabás Póczos. Data-driven random fourier features using stein effect. In *IJCAI*, pp. 1497–1503. ijcai.org, 2017.

Di Chen and Jeff M. Phillips. Relative error embeddings of the gaussian kernel distance. In *ALT*, volume 76 of *Proceedings of Machine Learning Research*, pp. 560–576. PMLR, 2017.

Li Chen, Shuisheng Zhou, Jiajun Ma, and Mingliang Xu. Fast kernel k-means clustering using incomplete cholesky factorization. *Appl. Math. Comput.*, 402:126037, 2021.

Radha Chitta, Rong Jin, and Anil K. Jain. Efficient kernel clustering using random fourier features. In *ICDM*, pp. 161–170. IEEE Computer Society, 2012.

Krzysztof Choromanski, Mark Rowland, Tamás Sarlós, Vikas Sindhwani, Richard E. Turner, and Adrian Weller. The geometry of random features. In *AISTATS*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1–9. PMLR, 2018.

Tri Dao, Christopher De Sa, and Christopher Ré. Gaussian quadrature for kernel features. In *NIPS*, pp. 6107–6117, 2017.

Tamás Erdélyi, Cameron Musco, and Christopher Musco. Fourier sparse leverage scores and approximate kernel learning. In *NeurIPS*, 2020.

Carl-Gustav Esseen. On the liapunov limit error in the theory of probability. *Ark. Mat. Astr. Fys.*, 28:1–19, 1942.

Shai Fine and Katya Scheinberg. Efficient SVM training using low-rank kernel representations. *J. Mach. Learn. Res.*, 2:243–264, 2001.

Mark A. Girolami. Mercer kernel-based clustering in feature space. *IEEE Trans. Neural Networks*, 13(3):780–784, 2002.

Alex Gittens and Michael W. Mahoney. Revisiting the nystrom method for improved large-scale machine learning. *J. Mach. Learn. Res.*, 17:117:1–117:65, 2016.

Lars Hormander. *An introduction to complex analysis in several variables*. van Nostrand, 1966.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pp. 604–613. ACM, 1998.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, pp. 8580–8589, 2018.

Hongjie Jia, Liangjun Wang, Heping Song, Qirong Mao, and Shifei Ding. An efficient nyström spectral clustering algorithm using incomplete cholesky decomposition. *Expert Syst. Appl.*, 186: 115813, 2021.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26:28, 1984.

Jean-Pierre Kahane. Hélices et quasi-hélices. In Leopoldo Nachbin (ed.), *Mathematical Analysis and Applications, Part B*, pp. 418–432. Academic Pr, first edition edition, 1981. ISBN 0125128029,9780125128025.

Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In *FOCS*, pp. 633–638. IEEE Computer Society, 2017.

Zhu Li, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards a unified analysis of random fourier features. *J. Mach. Learn. Res.*, 22:108:1–108:51, 2021.

Fanghui Liu, Xiaolin Huang, Yudong Chen, Jie Yang, and Johan A. K. Suykens. Random fourier features via fast surrogate leverage weighted sampling. In *AAAI*, pp. 4844–4851. AAAI Press, 2020.

Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan AK Suykens. Random features for kernel approximation: A survey on algorithms, theory, and beyond. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, 2021.

Konstantin Makarychev, Yury Makarychev, and Ilya P. Razenshteyn. Performance of johnson-lindenstrauss transform for k-means and k-medians clustering. In *STOC*, pp. 1027–1038. ACM, 2019.

Cameron Musco and Christopher Musco. Recursive sampling for the nystrom method. In *NIPS*, pp. 3833–3845, 2017.

Noam Nisan. Pseudorandom generators for space-bounded computation. *Comb.*, 12(4):449–461, 1992.

Dino Oglic and Thomas Gärtner. Nyström method with kernel k-means++ samples as landmarks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2652–2660. PMLR, 2017.

Sharath Raghvendra and Pankaj K. Agarwal. A near-linear time ε-approximation algorithm for geometric bipartite matching. *J. ACM*, 67(3):18:1–18:19, 2020.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, pp. 1177–1184. Curran Associates, Inc., 2007.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *NIPS*, pp. 1313–1320. Curran Associates, Inc., 2008.

Alessandro Rudi, Daniele Calandriello, Luigi Carratino, and Lorenzo Rosasco. On fast leverage score sampling and optimal learning. In *NeurIPS*, pp. 5677–5687, 2018.

W. Rudin. *Fourier Analysis on Groups*. Wiley Classics Library. Wiley, 1991. ISBN 9780471523642.

Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press, 2002.

Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.

Shahin Shahrampour, Ahmad Beirami, and Vahid Tarokh. On data-dependent random features for improved generalization in supervised learning. In *AAAI*, pp. 4026–4033. AAAI Press, 2018.

Zhao Song, David P. Woodruff, Zheng Yu, and Lichen Zhang. Fast sketching of polynomial kernels of polynomial degree. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9812–9823. PMLR, 2021.

Bharath K. Sriperumbudur and Zoltán Szabó. Optimal rates for random fourier features. In *NIPS*, pp. 1144–1152, 2015.

Yitong Sun, Anna C. Gilbert, and Ambuj Tewari. But how does it work in theory? linear SVM with random features. In *NeurIPS*, pp. 3383–3392, 2018.

Danica J. Sutherland and Jeff G. Schneider. On the error of random fourier features. In *UAI*, pp. 862–871. AUAI Press, 2015.

Shusen Wang, Alex Gittens, and Michael W. Mahoney. Scalable kernel k-means clustering with nyström approximation: Relative-error bounds. *J. Mach. Learn. Res.*, 20:12:1–12:49, 2019.

Christopher K. I. Williams and Matthias W. Seeger. Using the nyström method to speed up kernel machines. In *NIPS*, pp. 682–688. MIT Press, 2000.

David P. Woodruff and Amir Zandieh. Near input sparsity time kernel embeddings via adaptive sampling. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10324–10333. PMLR, 2020.

Jian Zhang, Avner May, Tri Dao, and Christopher Ré. Low-precision random fourier features for memory-constrained kernel approximation. In *AISTATS*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1264–1274. PMLR, 2019.

Xiantong Zhen, Haoliang Sun, Ying-Jun Du, Jun Xu, Yilong Yin, Ling Shao, and Cees Snoek. Learning to learn kernels with variational random features. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11409–11419. PMLR, 2020.

Appendices

A Proof of Lemma 3.1

Lemma 3.1. *If for some $r > 0$, K is analytic in $\{x \in \mathbb{R}^d : \|x\|_1 < r\}$, then for every $k \geq 1$ being even and every x s.t. $\|x\|_1 < r$, we have $\mathbb{E}[|X_i(x)|^k] \leq \left(\frac{4k\|x\|_1}{r}\right)^{2k}$.*

We first introduce following two lemmas to show the properties of $\mathbb{E}[X_i(x)^k]$.

Lemma A.1. *For any ω_i sampled in RFF and $k \geq 0$, we have $\mathbb{E}[\cos^k\langle\omega_i, x\rangle] = \frac{1}{2^k}\sum_{j=0}^{k}\binom{k}{j}K((2j-k)x)$.*

Proof. By eq. (1) it is sufficient to prove that

$$\cos^k\langle\omega_i, x\rangle = \frac{1}{2^k}\sum_{j=0}^{k}\binom{k}{j}\cos((2j-k)\langle\omega_i, x\rangle).$$

We prove this by induction. In the case of $k = 0$ the lemma holds obviously.

If for k the lemma holds, we have

$$\begin{aligned}
\cos^{k+1}(\langle\omega_i, x\rangle) &= \cos(\langle\omega_i, x\rangle) \cdot \frac{1}{2^k}\sum_{j=0}^{k}\binom{k}{j}\cos((2j-k)\langle\omega_i, x\rangle) \\
&= \frac{1}{2^k}\sum_{j=0}^{k}\binom{k}{j}\cos(\langle\omega_i, x\rangle)\cos(2j-k)\langle\omega_i, x\rangle) \\
&= \frac{1}{2^{k+1}}\sum_{j=0}^{k}\binom{k}{j}\Big(\cos((2j-k+1)\langle\omega_i, x\rangle) + \cos((2j-k-1)\langle\omega_i, x\rangle)\Big) \\
&= \frac{1}{2^{k+1}}\sum_{j=0}^{k}\binom{k}{j}\Big(\cos((2(j+1)-(k+1))\langle\omega_i, x\rangle) + \cos(2j-(k+1))\langle\omega_i, x\rangle)\Big) \\
&= \frac{1}{2^{k+1}}\sum_{j=0}^{k+1}\left(\binom{k}{j}+\binom{k}{j-1}\right)\cos((2j-(k+1))\langle\omega_i, x\rangle) \\
&= \frac{1}{2^{k+1}}\sum_{j=0}^{k+1}\binom{k+1}{j}\cos((2j-(k+1))\langle\omega_i, x\rangle)
\end{aligned}$$

where in the third equality we use the fact that $2\cos\alpha\cos\beta = \cos(\alpha+\beta)+\cos(\alpha-\beta)$. □

Lemma A.2. *If there exists $r > 0$ such that K is analytic in $\{x \in \mathbb{R}^d : \|x\|_1 < r\}$, then $\forall k \geq 0$, $\lim_{x\to 0}\frac{\mathbb{E}[X_i(x)^k]}{\|x\|_1^{2k}} = c$ for some constant c.*

Proof. We denote analytic function $K(x)$ as Taylor series around origin as

$$K(x) = \sum_{\beta\in\mathbb{N}^d} c_\beta x^\beta, \tag{3}$$

where $x^\beta := \prod_{i=1}^d x_i^{\beta_i}$ is a monomial and its coefficient is c_β. By definition, $c_0 = 1$ since $K(0) = 1$. We let $s : \mathbb{N}^d \to \mathbb{N}, s(\beta) := \sum_{i=1}^d \beta_i$ denote the degree of x^β. Since $K(x-y) = K(x,y) = K(y,x) = K(y-x)$ by definition, hence $K(x)$ is an even function, so $c_\beta = 0$ for $s(\beta)$ odd.

Recall that $X_i(x) := \cos\langle\omega_i, x\rangle - K(x)$. In the following, we drop the subscripts in X_i, ω_i and write X, ω for simplicity. By the definition of X we have

$$\mathbb{E}[X(x)^k] = \sum_{i=0}^{k}\binom{k}{i}K(x)^i(-1)^{k-i}\mathbb{E}[\cos^{k-i}\langle\omega, x\rangle]. \tag{4}$$

Note that by Lemma A.1, $\mathbb{E}[\cos^{k-i}\langle\omega, x\rangle] = \frac{1}{2^{k-i}}\sum_{j=0}^{k-i}\binom{k-i}{j}K((2j-(k-i))x)$. Plug this in eq. (4):

$$\begin{aligned}
\mathbb{E}[X(x)^k] &= \sum_{i=0}^{k}\binom{k}{i}K(x)^i(-1)^{k-i}\frac{1}{2^{k-i}}\sum_{j=0}^{k-i}\binom{k-i}{j}K((2j-(k-i))x) \\
&= \sum_{i=0}^{k}\binom{k}{i}K(x)^i\left(\frac{-1}{2}\right)^{k-i}\sum_{j=0}^{k-i}\binom{k-i}{j}\sum_{\beta\in\mathbb{N}^d}c_\beta(2j-k+i)^{s(\beta)}x^\beta \\
&= \sum_{\beta\in\mathbb{N}^d}c_\beta x^\beta\sum_{i=0}^{k}\binom{k}{i}K(x)^i\left(\frac{-1}{2}\right)^{k-i}\sum_{j=0}^{k-i}\binom{k-i}{j}(2j-k+i)^{s(\beta)}
\end{aligned}$$

where the second equality comes from the Tyler expansion of $K((2j-k+i)x)$. Next we will show that $\mathbb{E}[X(x)^k]$ is of degree at least $2k$.

For $\beta = 0$ note that $\sum_{i=0}^{k} \binom{k}{i} K(x)^i (-1)^{k-i} = (K(x)-1)^k$, since $K(x)$ is even and $K(0) = 1$, we have $\lim_{x\to 0} \frac{(K(x)-1)^k}{x^t} = 0, \forall t < 2k$. For $\beta \neq 0$, we next show that every term of degree less than $2k$ has coefficient zero.

Fix $\beta \neq 0$ and take Tyler expansion for $K(x)^i$

$$K(x)^i = \sum_{\beta_1,\beta_2,\ldots,\beta_i} c_{\beta_1} c_{\beta_2} \ldots c_{\beta_i} x^{\beta_1+\ldots+\beta_i},$$

Without loss of generality, we assume $\beta_{l+1}, ..., \beta_i$ are all βs that equals 0, so we have $c_{\beta_{l+1}} = ... = c_{\beta_i} = 1$.

Now we consider the coefficient of term $c_\beta c_{\beta_1} c_{\beta_2} \ldots, c_{\beta_l} x^{\beta+\sum_{j=1}^{l}\beta_j}$, which would be:

$$\tilde{C} \sum_{i=0}^{k} \binom{k}{i} \binom{i}{l} \left(\frac{-1}{2}\right)^{k-i} \sum_{j=0}^{k-i} \binom{k-i}{j} (2j-k+i)^{s(\beta)}$$

where $\tilde{C}$ is the number of ordered sequence $(\beta_1, \beta_2, \ldots, \beta_l)$, here, for $\beta_1 = \beta_2$, $(\beta_1, \beta_2, \ldots, \beta_l)$ and $(\beta_2, \beta_1, \ldots, \beta_l)$ are equivalent.

Next, we show if the degree of a monomial $s(\beta) + \sum_{j=1}^{l} s(\beta_j) < 2k$, its coefficient is zero. Since all $\beta_j \neq 0$, we may assume $s(\beta_j) \geq 2$, therefore $s(\beta) < 2k - 2l$.

Suppose operator J is a mapping from a function space to itself, such that $\forall f : \mathbb{R} \to \mathbb{R}, J(f) : \mathbb{R} \to \mathbb{R}$ is defined by $J(f)(x) := f(x+1)$. Denote $J^1 = J, J^k := J \circ J^{k-1}$ as its k-time composition, define J^0 to be the identity mapping such that $J^0(f) = f$. Similarly we can define addition that $(J_1 + J_2)(f) = J_1(f) + J_2(f)$ and scalar multiplication that $(\alpha J)(f) = \alpha(J(f))$. By definition, $cJ^m \circ J^n = J^m \circ (cJ^n) = cJ^{m+n}, \forall c \in \mathbb{R}, m, n \in \mathbb{N}$.

Let $L(x) = x^{s(\beta)}$, the coefficient can be rewritten as:

$$\tilde{C} \sum_{i=0}^{k} \binom{k}{i} \binom{i}{l} \left(\frac{-1}{2}\right)^{k-i} \sum_{j=0}^{k-i} \binom{k-i}{j} \left(J^{2j+i}(L)(-k)\right)$$

Let $P = \tilde{C} \sum_{i=0}^{k} \binom{k}{i} \binom{i}{l} \left(\frac{-1}{2}\right)^{k-i} \sum_{j=0}^{k-i} \binom{k-i}{j} J^{2j+i}(L)$, the above is $P(-k)$. Now we show $P \equiv 0$

$$\begin{aligned}
P =& \tilde{C} \left(\sum_{i=0}^{k} \binom{k}{i} \binom{i}{l} \left(\frac{-1}{2}\right)^{k-i} J^i \circ \left(\sum_{j=0}^{k-i} \binom{k-i}{j} J^{2j} \right) \right) (L) \\
=& \tilde{C} \left(\sum_{i=l}^{k} \binom{k}{l} \binom{k-l}{i-l} J^i \circ \left(-\frac{J^0 + J^2}{2} \right)^{k-i} \right) (L) \\
=& \tilde{C} J^l \circ \binom{k}{l} \left(\sum_{i=l}^{k} \binom{k-l}{i-l} J^{i-l} \circ \left(-\frac{J^0 + J^2}{2} \right)^{k-i} \right) (L) \\
=& \tilde{C} J^l \circ \binom{k}{l} \left(J - \frac{J^0 + J^2}{2} \right)^{k-l} (L).
\end{aligned}$$

Note that $\left(J - \frac{J^0+J^2}{2}\right)(f)(x) = (f(x+1) - f(x))/2 - (f(x+2) - f(x+1))/2$ calculates second order difference, namely, $\forall f$ that is a polynomial of degree $k \geq 2$, $\left(J - \frac{J^0+J^2}{2}\right)(f)$ is a polynomial of degree $k-2$, and $\forall f$ that is a polynomial of degree $k < 2$, $\left(J - \frac{J^0+J^2}{2}\right)(f)$ is 0.

Since L is a polynomial of degree less than $2(k-l)$, we have

$$\left(J - \frac{J^0 + J^2}{2} \right)^{k-l} (L) \equiv 0.$$

Combining the above two cases, we have proved eq. (4) is of degree at least $2k$, which completes our proof. □

Proof of Lemma 3.1. If $2k\|x\|_1 \geq r$, since $|X_i(x)| = |\cos\langle\omega_i, x\rangle - K(x)| \leq 2$, we have $\mathbb{E}[X_i(x)^k] \leq 2^k \leq \left(\frac{4k\|x\|_1}{r}\right)^{2k}$. Otherwise $\|x\|_1 < r/2k$. Define $g_k(x) := \mathbb{E}[X_i(x)^k]$, we have:

$$g_k(x) = \sum_{i=0}^{\infty}\left(\sum_{j=1}^{d} x_j \frac{\partial}{\partial x_j}\right)^i \frac{g_k(x)}{i!} = \left(\sum_{j=1}^{d} x_j \frac{\partial}{\partial x_j}\right)^{2k} \frac{g_k(\theta x)}{(2k)!}, \quad \theta \in [0,1]$$

where the second equation comes from Lemma A.2 and Taylor expansion with Lagrange remainder.

Lemma A.3 (Cauchy's integral formula for multivariate functions Hormander 1966)**.** *For $f(z_1, ..., z_d)$ analytic in $\Delta(z, r) = \left\{\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_d) \in \mathbb{C}^d; |\zeta_\nu - z_\nu| \leq r_\nu, \nu = 1, \ldots, d\right\}$*

$$f(z_1, \ldots, z_d) = \frac{1}{(2\pi \mathrm{i})^d} \int_{\partial D_1 \times \partial D_2 \times \cdots \times \partial D_d} \frac{f(\zeta_1, \ldots, \zeta_d)}{(\zeta_1 - z_1)\cdots(\zeta_d - z_d)}\, d\zeta.$$

Furthermore,

$$\frac{\partial^{k_1+\cdots+k_d} f(z_1, z_2, \ldots, z_d)}{\partial {z_1}^{k_1} \cdots \partial {z_d}^{k_d}} = \frac{k_1! \cdots k_d!}{(2\pi \mathrm{i})^d} \int_{\partial D_1 \times \partial D_2 \cdots \times \partial D_d} \frac{f(\zeta_1, \ldots, \zeta_d)}{(\zeta_1 - z_1)^{k_1+1} \cdots (\zeta_d - z_d)^{k_d+1}}\, d\zeta.$$

If in addition $|f| < M$, we have the following evaluation:

$$\left|\frac{\partial^{k_1+\cdots+k_d} f(z_1, z_2, \ldots, z_d)}{\partial {z_1}^{k_1} \cdots \partial {z_d}^{k_d}}\right| \leq \frac{M k_1! \cdots k_d!}{{r_1}^{k_1} \cdots {r_d}^{k_d}}.$$

Recall that $g_k(x) = \mathbb{E}[X_i(x)^k] = \sum_{i=0}^{k} \binom{k}{i} K(x)^i (-1)^{k-i} \frac{1}{2^{k-i}} \sum_{j=0}^{k-i} \binom{k-i}{j} K((2j - (k-i))x)$, so $g_k(x) = \mathrm{poly}(K(x), K(-x), \ldots, K(kx), K(-kx))$ is analytic when $\|x\|_1 \leq r/k$. Applying Cauchy's integral formula Lemma A.3 (here $\|z + \theta x\|_1 \leq 2 \cdot r/2k$ is in the analytic area),

$$\begin{aligned} g_k(x) &= \sum_{t_1+\cdots+t_d=2k} \frac{x_1^{t_1} x_2^{t_2} \ldots x_d^{t_d}}{t_1! t_2! \ldots t_d!} \frac{\partial^{2k} g_k(\theta x)}{\partial x_1^{t_1} \partial x_2^{t_2} \ldots \partial x_d^{t_d}} \\ &= \sum_{t_1+\cdots+t_d=2k} \frac{x_1^{t_1} x_2^{t_2} \ldots x_d^{t_d}}{(2\pi \mathrm{i})^d} \int_{z \in \mathbb{C}^d, |z_i| = \frac{r}{2k}} \frac{g_k(z + \theta x)}{z_1^{t_1+1} \ldots z_d^{t_d+1}}\, \mathrm{d}z \end{aligned}$$

we have

$$|g_k(x)| \leq \sup_{|z_i| = r/2k} |g_k(z + \theta x)| \left(\frac{2k}{r}\right)^{2k} \left|\sum_{i=1}^{d} x_i\right|^{2k} \leq \left(\frac{4k\|x\|_1}{r}\right)^{2k}.$$

□

B PROOF OF LEMMA 3.2

Lemma 3.2. *For kernel K which is shift-invariant and analytic at the origin, there exist $c_K, R_K > 0$ such that for all $\|x\|_1 \leq R_K$, $\frac{1-K(x)}{\|x\|_1^2} \geq \frac{c_K}{2}$.*

Proof. It suffices to prove that $\liminf_{x \to 0} \frac{1-K(x)}{\|x\|_1^2} \geq c > 0$, for some c. Towards proving this, we show that K is *strongly* convex at origin. In fact, by definition, $\Re \int_{\mathbb{R}^d} p(\omega) e^{\mathrm{i}\langle\omega, tx\rangle}\, \mathrm{d}\omega = K(tx)$ for every fixed x, therefore $K''(tx) = \Re \int_{\mathbb{R}^d} \|\omega\|^2 p(\omega) e^{\mathrm{i}\langle\omega, tx\rangle}\, \mathrm{d}\omega > 0$, hence $K(tx)$ is *strongly* convex with respect to t at origin, so is $K(x)$. □

C PROOF OF THEOREM 1.2

Theorem 1.2 (Upper bound)**.** *Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a kernel function which is shift-invariant and analytic at the origin, with feature mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$ for some feature space $\mathcal{H}$. For every $0 < \delta \le \varepsilon \le 2^{-16}$, every $d, D \in \mathbb{N}$, $D \ge \max\{\Theta(\varepsilon^{-1}\log^3(1/\delta)), \Theta(\varepsilon^{-2}\log(1/\delta))\}$, if $\pi : \mathbb{R}^d \to \mathbb{R}^D$ is an RFF mapping for K with target dimension D, then for every $x, y \in \mathbb{R}^d$,*

$$\Pr[|\operatorname{dist}_\pi(x,y) - \operatorname{dist}_\varphi(x,y)| \le \varepsilon \cdot \operatorname{dist}_\varphi(x,y)] \ge 1-\delta.$$

Proof. When $\|x-y\|_1 \ge R_K$, consider the function $g(t) = K(t(x-y))$. It follows from definition that $g'(0) = 0, g''(t) = -\Re\int_{\mathbb{R}^d} \|\omega\|^2\|x-y\|^2 p(\omega)e^{\mathrm{i}\langle\omega, t(x-y)\rangle}\,\mathrm{d}\omega < 0$, so $g(t)$ strictly decreases for all $t > 0$. So $2 - 2K(x-y) \ge 2 - 2\max_{\|x-y\|_1=R_K} K(x-y) > 0$. We denote $t = 2 - 2\max_{\|x-y\|_1=R_K} K(x-y)$, so by Chernorff bound, when $D \ge \frac{1}{2t}(\ln\frac{1}{\delta} + \ln 2)$, we have:

$$\Pr\left[\left|\frac{2}{D}\sum_{i=1}^D X_i(x-y)\right| \le \varepsilon\cdot(2-2K(x-y))\right] \ge \Pr\left[\left|\frac{2}{D}\sum_{i=1}^D X_i(x-y)\right| \le t\right] \ge 1-\delta.$$

When $\|x-y\|_1 \le R_K$, by Lemma 3.2, we have $2 - 2K(x-y) \ge c\|x-y\|_1^2$. Then we have:

$$\Pr\left[\left|\frac{2}{D}\sum_{i=1}^D X_i(x-y)\right| \le \varepsilon\cdot(2-2K(x-y))\right] \ge \Pr\left[\left|\frac{2}{D}\sum_{i=1}^D X_i(x-y)\right| \le c\varepsilon\cdot\|x-y\|_1^2\right].$$

We take $r = r_K$ for simplicity of exhibition. Assume $\delta \le \min\{\varepsilon, 2^{-16}\}$, let $k = \log(2D^2/\delta), t = 64k^2/r^2$, is even. By Markov's inequality and Lemma 3.1:

$$\Pr[|X_i(x-y)| \ge t\|x-y\|_1^2] = \Pr\left[|X_i(x-y)|^k \ge t^k\|x-y\|_1^{2k}\right] \le \frac{(4k)^{2k}}{t^k r^{2k}} = 4^{-k} \le \frac{\delta}{2D^2}.$$

For simplicity denote $X_i(x-y)$ by X_i, $\|x-y\|_1^2$ by ℓ and define $X_i' = \mathbb{1}_{[|X_i|\ge t\ell]}t\ell\cdot\operatorname{sgn}(X_i) + \mathbb{1}_{[|X_i|<t\ell]}X_i$, note that $\mathbb{E}[X_i] = 0$. Then:

$$\begin{aligned}
|\mathbb{E}[X_i']| &\le |\mathbb{E}[X_i' \mid |X_i'| < t\ell]|\cdot\Pr[|X_i'| < t\ell] + t\ell\cdot|\Pr[X_i' \ge t\ell] - \Pr[X_i' \le -t\ell]| \\
&= |\mathbb{E}[X_i' \mid |X_i'| < t\ell]|\cdot\Pr[|X_i| < t\ell] + t\ell\cdot|\Pr[X_i \ge t\ell] - \Pr[X_i \le -t\ell]| \\
&= \frac{|\mathbb{E}[X_i] - \mathbb{E}\left[X_i \mid |X_i| \ge t\ell\right]\Pr[|X_i| \ge t\ell]|}{\Pr[|X_i| < t\ell]}\Pr[|X_i| < t\ell] + t\ell\cdot|\Pr[X_i \ge t\ell] - \Pr[X_i \le -t\ell]| \\
&= |\mathbb{E}[X_i] - \mathbb{E}\left[X_i \mid |X_i| \ge t\ell\right]\Pr[|X_i| \ge t\ell]| + t\ell\cdot|\Pr[X_i \ge t\ell] - \Pr[X_i \le -t\ell]| \\
&= |\mathbb{E}\left[X_i \mid |X_i| \ge t\ell\right]\Pr[|X_i| \ge t\ell]| + t\ell\cdot|\Pr[X_i \ge t\ell] - \Pr[X_i \le -t\ell]|
\end{aligned}$$

where $t\ell\cdot|\Pr[X_i > t\ell] - \Pr[X_i < -t\ell]| \le t\ell\cdot\Pr[|X_i| > t\ell] \le t\ell\delta/(2D^2)$. The first inequality is by considering the two conditions $|X_i| < t\ell$ and $|X_i| \ge t\ell$, then taking a triangle inequality. The first and second equations are by definition of X_i, X_i'. The third equation is a straightforward computation. The last equation is due to $\mathbb{E}[X_i] = 0$. By Lemma 3.1 for every integer α,

$$\Pr\left[|X_i| \ge \alpha\ell/r^2\right] = \Pr\left[|X_i|^{\sqrt{\alpha}/8} \ge (\alpha\ell/r^2)^{\sqrt{\alpha}/8}\right] \le \frac{\mathbb{E}[|X_i|^{\sqrt{\alpha}/8}]}{(\alpha\ell/r^2)^{\sqrt{\alpha}/8}} \le 4^{-\sqrt{\alpha}/8}.$$

The first equality is straightforward. The first inequality is by Markov. The second equality is by $\mathbb{E}[|X_i|^{\sqrt{\alpha}/8}] \le \left(\frac{\alpha\ell}{4r^2}\right)^{\sqrt{\alpha}/8}$ which follows from Lemma 3.1, and a rearrangement of parameters, where r is the parameter r in Lemma 3.1. Therefore,

$$\begin{aligned}
|\mathbb{E}[X_i \mid |X_i| \ge t\ell]|\Pr[|X_i| \ge t\ell] &\le \mathbb{E}[|X_i| \mid |X_i| \ge t\ell]\Pr[|X_i| \ge t\ell] \\
&\le (t + \frac{1}{r^2})\ell\cdot\Pr[|X_i| \ge t\ell] + \frac{\ell}{r^2}\sum_{\text{integer }\alpha \ge tr^2+1}\Pr[|X_i| \ge \alpha\ell/r^2] \\
&\le (t + \frac{1}{r^2})\ell\cdot\frac{\delta}{2D^2} + \ell\int_{tr^2}^\infty 4^{-\sqrt{\alpha}/8}\,\mathrm{d}\alpha \\
&\le \ell\left((t + \frac{1}{r^2})\frac{\delta}{2D^2} + \frac{16}{r^2\ln 4}4^{-tr^2/8}\right).
\end{aligned}$$

The first inequality is by the property of absolute value. The second inequality is because we can divide the event $|X_i| \geq t\ell$ into $|X_i| \in [\alpha\ell/r^2, (\alpha+1)\ell/r^2), \alpha = tr^2, tr^2+1, \ldots$ and when $|X_i| \in [\alpha\ell/r^2, (\alpha+1)\ell/r^2)$, $|X_i| < (\alpha+1)\ell/r^2$. The third inequality is by pluging in the previous bound for $\Pr[|X_i| \geq \alpha\ell/r^2]$. The last inequality is by a calculation of the integral.

By plugging in parameters t, δ, $D \geq \max\{\Theta(\varepsilon^{-1}\log^3(1/\delta)), \Theta(\varepsilon^{-2}\log(1/\delta))\}$, we have $\mathbb{E}[|X_i'|] \leq \delta\ell$. Note that the $\Theta(D)$ hides a constant r. Denote σ'^2 as the variance of X_i'. So $\sigma' \leq 64\ell/r^2$ by Lemma 3.1.

Lemma C.1 (Bernstein's Inequality)**.** *Let $X_1, .., X_D$ be independent zero-mean random variables. Suppose that $|X_i| \leq M, \forall i$, then for all positive t,*

$$\Pr\left[\sum_{i=1}^{D} X_i \geq t\right] \leq \exp\left(-\frac{t^2/2}{Mt/3 + \sum_{i=1}^{D}\mathbb{E}[X_i^2]}\right).$$

Applying Bernstein's Inequality to X_i',

$$\begin{aligned}\Pr\left[\sum_{i=1}^{D} X_i' - D\mathbb{E}[X_i'] \geq (c\varepsilon\ell/\sigma')D\sigma'\right] &\leq \exp\left(-\frac{c^2\varepsilon^2 D}{ct\varepsilon + 2\sigma'^2/\ell^2}\right) \\ &\leq \max\left\{\exp\left(-\frac{c\varepsilon^2 D}{2t\varepsilon}\right), \exp\left(-\frac{c^2\varepsilon^2 D}{4\sigma'^2/\ell^2}\right)\right\}.\end{aligned}$$

Since $D \geq \max\{\Theta\left(t\varepsilon^{-1}\log(1/\delta)\right), \Theta\left(\varepsilon^{-2}\log(1/\delta)\right)\}$, we have $\Pr\left[\sum_{i=1}^{D} X_i' \geq \varepsilon(\ell/\sigma')D\sigma'\right] \leq \delta/2$. With $1 - \frac{\delta}{2}$ probability, every $X_i \leq t\ell, X_i' = X_i$. Therefore, $\Pr\left[\sum_{i=1}^{D} X_i \geq D(\delta+\varepsilon)\ell\right] \leq \delta/2$. Combine it together, $\Pr[|\operatorname{dist}_\pi(x,y) - \operatorname{dist}_\varphi(x,y)| \leq \varepsilon \cdot \operatorname{dist}_\varphi(x,y)] \geq 1-\delta$.

□

D Proof of Theorem 3.1

Theorem 3.1 (Generalization of Makarychev et al. 2019, Theorem 3.6)**.** *For kernel k-clustering problem with ℓ_2^p-objective whose kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is shift-invariant and analytic at the origin, for every data set $P \subset \mathbb{R}^d$, the RFF mapping $\pi : \mathbb{R}^d \to \mathbb{R}^D$ with target dimension $D = \Omega(p^2\log^3\frac{k}{\alpha} + p^5\log^3\frac{1}{\varepsilon} + p^8)/\varepsilon^2$ satisfies*

$$\Pr[\forall k\text{-partition } \mathcal{C} \text{ of } P : \operatorname{cost}_p^\pi(P,\mathcal{C}) \in (1\pm\varepsilon)\cdot\operatorname{cost}_p^\varphi(P,\mathcal{C})] \geq 1-\delta.$$

The proof relies on a key notion of $(\varepsilon,\delta,\rho)$-dimension reduction from (Makarychev et al., 2019), and we adopt it with respect to our setting/language of kernel distance as follows.

Definition D.1 (Makarychev et al. 2019, Definition 2.1)**.** For $\varepsilon, \delta, \rho > 0$, a feature mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$ for some Hilbert space $\mathcal{H}$, a random mapping $\pi_{d,D} : \mathbb{R}^d \to \mathbb{R}^D$ is an $(\varepsilon,\delta,\rho)$-dimension reduction, if

- for every $x, y \in \mathbb{R}^d$, $\frac{1}{1+\varepsilon}\operatorname{dist}_\varphi(x,y) \leq \operatorname{dist}_\pi(x,y) \leq (1+\varepsilon)\operatorname{dist}_\varphi(x,y)$ with probability at least $1-\delta$, and
- for every fixed $p \in [1,\infty)$, $\mathbb{E}\left[\mathbb{1}_{\{\operatorname{dist}_\pi(x,y) > (1+\varepsilon)\operatorname{dist}_\varphi(x,y)\}}\left(\frac{\operatorname{dist}_\pi(x,y)^p}{\operatorname{dist}_\varphi(x,y)^p} - (1+\varepsilon)^p\right)\right] \leq \rho$.

In Makarychev et al. (2019), most results are stated for a particular parameter setup of Definition D.1 resulted from Johnson-Lindenstrauss transform (Johnson & Lindenstrauss, 1984), but their analysis actually works for other similar parameter setups. The following is a generalized statement of (Makarychev et al., 2019, Theorem 3.5) which also reveals how alternative parameter setups affect the distortion. We note that this is simply a more precise and detailed statement of (Makarychev et al., 2019, Theorem 3.5), and it follows from exactly the same proof in Makarychev et al. (2019).

Lemma D.1 (Makarychev et al. 2019, Theorem 3.5)**.** *Let $0 < \varepsilon, \alpha < 1$ and $\theta := \min\{\varepsilon^{p+1}3^{-(p+1)(p+2)}, \alpha\varepsilon^p/(10k(1+\varepsilon)^{4p-1}), 1/10^{p+1}\}$. If some $(\varepsilon,\delta,\rho)$-dimension reduction*

π for feature mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$ of some kernel function satisfies $\delta \le \min(\theta^7/600, \theta/k)$, $\binom{k}{2}\delta \le \frac{\alpha}{2}, \rho \le \theta$, then with probability at least $1-\alpha$, for every partition $\mathcal{C}$ of P,

$$\begin{aligned}\operatorname{cost}_p^{\pi}(P,\mathcal{C}) &\le (1+\varepsilon)^{3p}\operatorname{cost}_p^{\varphi}(P,\mathcal{C}),\\ (1-\varepsilon)\operatorname{cost}_p^{\varphi}(P,\mathcal{C}) &\le (1+\varepsilon)^{3p-1}\operatorname{cost}_p^{\pi}(P,\mathcal{C}).\end{aligned}$$

Proof of Theorem 3.1. We verify that setting $D = \Theta(\log^3 \frac{k}{\alpha} + p^3 \log^3 \frac{1}{\varepsilon} + p^6)/\varepsilon^2$, the RFF mapping π with target dimension D satisfies the conditions in Lemma D.1, namely, it is a $(\varepsilon, \delta, \rho)$-dimension reduction . In fact, Theorem 1.2 already implies such π satisfies that for every $x, y \in \mathbb{R}^d$, $\frac{1}{1+\varepsilon}\operatorname{dist}_{\varphi}(x,y) \le \operatorname{dist}_{\pi}(x,y) \le (1+\varepsilon)\operatorname{dist}_{\varphi}(x,y)$ with probability at least $1-\delta$, where $\delta = e^{-cf(\varepsilon,D)}$ for some constant c, and $f(\varepsilon, D) := \max\{\varepsilon^2 D, \varepsilon^{1/3}D^{1/3}\}$. For the other part,

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{1}_{\{\operatorname{dist}_{\pi}(x,y)>(1+\varepsilon)\operatorname{dist}_{\varphi}(x,y)\}}\left(\frac{\operatorname{dist}_{\pi}(x,y)^p}{\operatorname{dist}_{\varphi}(x,y)^p} - (1+\varepsilon)^p\right)\right]\\
&= \int_{\varepsilon}^{\infty}\left((1+t)^p - (1+\varepsilon)^p\right)\mathrm{d}\left(-\Pr\left(\frac{\operatorname{dist}_{\pi}(x,y)}{\operatorname{dist}_{\varphi}(x,y)} > t+1\right)\right)\\
&= \left[-(1+m)^p + (1+\varepsilon)^p\right]\Pr\left(\frac{\operatorname{dist}_{\pi}(x,y)}{\operatorname{dist}_{\varphi}(x,y)} > m+1\right)\Bigg|_{m=\varepsilon}^{m=+\infty}\\
&\qquad + \int_{\varepsilon}^{\infty} p(1+t)^{p-1}\Pr\left(\frac{\operatorname{dist}_{\pi}(x,y)}{\operatorname{dist}_{\varphi}(x,y)} > t+1\right)\mathrm{d}t \qquad \text{(integration by part)}\\
&= \int_{\varepsilon}^{\infty} p(1+t)^{p-1}\Pr\left(\frac{\operatorname{dist}_{\pi}(x,y)}{\operatorname{dist}_{\varphi}(x,y)} > t+1\right)\mathrm{d}t\\
&\le \int_{\varepsilon}^{\infty} p(1+t)^{p-1}e^{-cf(t,D)}\,\mathrm{d}t.
\end{aligned}$$

Where the third equality follows by $\Pr\left(\frac{\operatorname{dist}_{\pi}(x,y)}{\operatorname{dist}_{\varphi}(x,y)} > m\right)$ decays exponentially fast with respect to m. Observe that for $p \ge 1, D \ge \frac{(p-1)^3}{8c^3}, p(1+t)^{p-1}e^{-cD^{\frac{1}{3}}t^{\frac{1}{3}}/2}$ decrease when $t \ge \varepsilon$, and for $D \ge \frac{c(p-1)}{\varepsilon^2}, p(1+t)^{p-1}e^{-ct^2D/2}$ decrease when $t \ge \varepsilon$. Hence for $D \ge \max\{\frac{(p-1)^3}{8c^3}, \frac{c(p-1)}{\varepsilon^2}\}$, we have

$$\int_{\varepsilon}^{\infty} p(1+t)^{p-1}e^{-cf(t,D)}dt \le c'\int_{\varepsilon}^{\infty} e^{-cf(t,D)/2}dt < c''e^{-cf(\varepsilon,D)/2}.$$

In conclusion, by setting $D = \Theta(\log^3 \frac{k}{\alpha} + p^3 \log^3 \frac{1}{\varepsilon} + p^6)/\varepsilon^2$, for $\delta = e^{-cf(\varepsilon,D)}, \rho = c''e^{-cf(\varepsilon,D)}$ and $f(\varepsilon, D) = \max\{\varepsilon^2 D, \varepsilon^{1/3}D^{1/3}\}$, it satisfies $\delta \le \min(\theta^7/600, \theta/k)$, $\binom{k}{2}\delta \le \frac{\alpha}{2}, \rho \le \theta$. This verifies the condition of Lemma D.1.

Finally, we conclude the proof of Theorem 3.1 by plugging $\varepsilon' = \varepsilon/3p$ and the above mentioned RFF mapping π with target dimension D into Lemma D.1. □

E Proof of Theorem 4.1

Theorem 4.1. *Consider $\Delta \ge \rho > 0$, and a shift-invariant kernel function $K : \mathbb{R}^d \to \mathbb{R}$, denoting its feature mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$. Then there exists $x, y \in \mathbb{R}^d$ that are (Δ, ρ)-bounded, such that for every $0 < \varepsilon < 1$, the RFF mapping π for K with target dimension D satisfies*

$$\Pr[|\operatorname{dist}_{\varphi}(x,y) - \operatorname{dist}_{\pi}(x,y)| \ge \varepsilon\cdot\operatorname{dist}_{\varphi}(x,y)] \ge \frac{2}{\sqrt{2\pi}}\int_{6\varepsilon\sqrt{D/s_K^{(\Delta,\rho)}}}^{\infty} e^{-s^2/2}\,\mathrm{d}s - O\left(D^{-\frac{1}{2}}\right) \quad (2)$$

where $s_K^{(\Delta,\rho)} := \sup_{(\Delta,\rho)\text{-bounded } x\in\mathbb{R}^d} s_K(x)$, and $s_K(x) := \dfrac{1+K(2x)-2K(x)^2}{2(1-K(x))^2}$.

Proof. Our proof requires the following anti-concentration inequality.

Lemma E.1 (Berry 1941; Esseen 1942)**.** *For i.i.d. random variables $\xi_i \in \mathbb{R}$ with mean 0 and variance 1, let $X := \frac{1}{\sqrt{D}} \sum_{i=1}^{D} \xi_i$, then for any t,*

$$\Pr[X \geq t] \geq \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-s^2/2} \, \mathrm{d}s - O(D^{-\frac{1}{2}})$$

Let $X_i(x) := \cos\langle \omega_i, x\rangle - K(x), \sigma(x) := \sqrt{\mathrm{Var}(X_i(x))} = \sqrt{\frac{1+K(2(x))-2K(x)^2}{2}}$, choose x, y such that $s_K(x-y) = s_K^{(\Delta,\rho)}$. Clearly, such pair of x, y satisfies that $(x-y)$ is (Δ, ρ)-bounded. In fact, it is without loss of generality to assume that both x and y are (Δ, ρ)-bounded, since one may pick $y' = 0, x' = x - y$ and still have $x' - y' = x - y$. We next verify that such x, y satisfy our claimed properties. Indeed,

$$\begin{aligned}
&\Pr[|\operatorname{dist}_\varphi(x,y) - \operatorname{dist}_\pi(x,y)| \geq \varepsilon \cdot \operatorname{dist}_\varphi(x,y)] \\
\geq &\Pr[|\operatorname{dist}_\varphi(x,y)^2 - \operatorname{dist}_\pi(x,y)^2| \geq 6\varepsilon \cdot \operatorname{dist}_\varphi(x,y)^2] \\
= &\Pr\left[\left|\frac{2}{D}\sum_{i=1}^{D} X_i(x-y)\right| \geq 6\varepsilon(2 - 2K(x-y))\right] \\
= &\Pr\left[\left|\frac{1}{\sqrt{D}\cdot\sigma(x-y)}\sum_{i=1}^{D} X_i(x-y)\right| \geq 6\varepsilon(1 - K(x-y)) \cdot \frac{\sqrt{D}}{\sigma(x-y)}\right] \\
\geq &- O(D^{-1/2}) + \frac{2}{\sqrt{2\pi}} \int_{6\varepsilon(1-K(x-y))\sqrt{D}/\sigma(x-y)}^{\infty} e^{-s^2/2} \, \mathrm{d}s \\
= &- O(D^{-1/2}) + \frac{2}{\sqrt{2\pi}} \int_{6\varepsilon\sqrt{D/s_K^{(\Delta,\rho)}}}^{\infty} e^{-s^2/2} \, \mathrm{d}s,
\end{aligned}$$

where the second inequality is by Lemma E.1, and the the second-last equality follows from the definition of $s_K(\cdot)$, and that of x, y such that $s_K(x-y) = s_K^{(\Delta,\rho)}$. □

F Beyond RFF: Oblivious Embedding for Laplacian Kernel with Small Computing Time

In this section we show an oblivious feature mapping for Laplacian kernel dimension reduction with small computing time. The following is the main theorem.

Theorem F.1. *Let K be a Laplacian kernel with feature mapping $\varphi : \mathbb{R}^d \to \mathcal{H}$. For every $0 < \delta \leq \varepsilon \leq 2^{-16}$, every $\varepsilon_g > 0$, every $d, D \in \mathbb{N}$, $D \geq \max\{\Theta(\varepsilon^{-1}\log^3(1/\delta)), \Theta(\varepsilon^{-2}\log(1/\delta))\}$, every $\rho_0 \in (0,1)$, there is a mapping $\pi : \mathbb{R}^d \to \mathbb{R}^D$, such that for every $x, y \in \mathbb{R}^d$ (Δ, ρ)-bounded,*

$$\Pr[|\operatorname{dist}_\pi(x,y) - \operatorname{dist}_\varphi(x,y)| \leq \varepsilon \cdot \operatorname{dist}_\varphi(x,y)] \geq 1 - \delta - \varepsilon_g.$$

The time for evaluating π is $dD \operatorname{poly}(\log \frac{\Delta}{\rho} \log \varepsilon_g^{-1})$.

For simplicity of exhibition, we first handle the case when the input data are from $\mathbb{N}^d$. At the end we will describe how to handle the case when the input data are from $\mathbb{R}^d$ by a simple transformation. Let $N \in \mathbb{N}$ be s.t. every entry of an input data point is at most N. Even though input data are only consisted of integers, the mapping construction needs to handle fractions, as we will later consider some numbers generated from Gaussian distributions or numbers computed in the RFF mapping. So we first setup two numbers, $\Delta' = \operatorname{poly}(N)$ large enough and $\rho' = 1/\operatorname{poly}(N)$ small enough. All our following operations are working on numbers that are (Δ', ρ')-bounded. Denote ρ'/Δ' as ρ_0 for convenience.

F.1 Embedding from ℓ_1 to ℓ_2^2

Now we describe an isometric embedding from ℓ_1 norm to ℓ_2^2. This construction is based on Kahane (1981), in which the first such construction of finite dimension was given, to the best of our knowledge. Let $\pi_1 : \mathbb{N} \to \mathbb{R}^N$ be such that for every $x \in \mathbb{N}, x \leq N$, $\pi_1(x)[j] = 1$, if $j \in [1, x]$ and

$\pi_1(x)[j] = 0$ otherwise. Let $\pi_1^{(d)} : \mathbb{N}^d \to \mathbb{R}^{Nd}$ be such that for every $x \in \mathbb{N}^d, x_i \leq N, \forall i \in [d]$, we have $\pi_1^{(d)}(x)$ being the concatenation of d vectors $\pi_1(x_i), i \in [d]$.

Lemma F.1. *For every $x, y \in \mathbb{N}^d$ with $x_i, y_i \leq N, i \in [d]$, it holds that*

$$\|x - y\|_1 = \|\pi_1^{(d)}(x) - \pi_1^{(d)}(y)\|_2^2.$$

Proof. Notice that for every $i \in [d]$, $\pi_1(x_i)$ has its first x_i entries being 1 while $\pi_1(y_i)$ has its first y_i entries being 1. Thus $\|\pi_1(x_i) - \pi_1(y_i)\|_2^2$ is exactly $\|x_i - y_i\|_1$. If we consider all the d dimensions, then by the construction of $\pi_1^{(d)}$, the lemma holds. □

F.2 Feature Mapping for Laplacian Kernel

Notice that we can apply the mapping $\pi_1^{(d)}$ and the RFF mapping ϕ from Theorem 1.2 for the kernel function $\exp(-\|x-y\|_2^2)$ which is actually a Gaussian kernel. This gives a mapping which preserves kernel distance for Laplacian kernel. To be more precise, we setup the mapping to be $\pi = \phi \circ \pi_1^{(d)}$. The only drawback is that the running time is high, as in the above mapping we map d dimension to dN dimension. We formalize this as the following theorem.

Theorem F.2. *Let K be a Laplacian kernel with feature map $\varphi : \mathbb{R}^d \to \mathcal{H}$. For every $0 < \delta \leq \varepsilon \leq 2^{-16}$, every $d, D, N \in \mathbb{N}$, $D \geq \max\{\Theta(\varepsilon^{-1}\log^3(1/\delta)), \Theta(\varepsilon^{-2}\log(1/\delta))\}$, there exists a mapping $\pi : \mathbb{R}^d \to \mathbb{R}^D$ s.t. for every $x, y \in \mathbb{N}^d, x, y \leq N$,*

$$\Pr[|\operatorname{dist}_\pi(x, y) - \operatorname{dist}_\varphi(x, y)| \leq \varepsilon \cdot \operatorname{dist}_\varphi(x, y)] \geq 1 - \delta.$$

The time of evaluating π is $\tilde{O}(dDN)$.

Proof. Consider the map π defined above. It follows by Lemma F.1 and Theorem 1.2. The running time is as stated, since we need to compute a vector of length $O(N)$ and then apply the RFF on this vector. □

The time complexity is a bit large, since we need to compute $\pi_1^{(d)}$ which has a rather large dimension dN. Next we show how to reduce the time complexity.

F.3 An Alternate Construction

We give the following map π' which has the same output distribution as that of $\pi = \phi \circ \pi_1^{(d)}$. Then in the next subsection we will use pseudorandom generators to replace the randomness in π' while using its highly efficiency in computation to reduce the time complexity. Notice that in computing $\phi \circ \pi_1^{(d)}$, for each output dimension, the crucial step is computing $\langle \omega, \pi_1^{(d)}(x) \rangle$ for some Gaussian distribution $\omega \in \mathbb{R}^{dN}$ which has each dimension being an independent gaussian distribution ω_0. The final output is a function of $\langle \omega, \pi_1^{(d)}(x) \rangle$. So we only need to present the construction for the first part, i.e. the inner product of an N dimension Gaussian distribution and $\pi_1(x_1)$. For the other parts the computations are the same and finally we only need to sum them up. Hence to make the description simpler, we denote this inner product as $\langle \omega, \pi_1(x) \rangle$, where now we let $x \in \mathbb{N}, x \leq N$ and ω has N dimensions each being an independent ω_0.

Let h be the smallest integer s.t. $N \leq 2^h$. Consider a binary tree where each node has exactly 2 children. The depth is h. So it has exactly $2^h \geq N$ leaf nodes in the last layer. For each node v, we attach a random variable α_v in the following way. For the root, we attach a Gaussian variable which is the summation of 2^h independent Gaussian variable with distribution ω_0. Then we proceed layer by layer from the root to leaves. For each u, v being children of a common parent w, assume that α_w is the summation of 2^l independent ω_0 distributions. Then let α_u be the summation of the first 2^{l-1} distributions among them and α_v the summation of the second 2^{l-1} distributions. That is $\alpha_w = \alpha_u + \alpha_v$ with α_u, α_v being independent. Notice that conditioned on $\alpha_w = a$, then α_u takes the value b with probability $\Pr_{\alpha_u, \alpha_v \text{ i.i.d.}}[\alpha_u = b \mid \alpha_u + \alpha_v = a]$. α_v takes the value $a - b$ when α_u takes value b.

The randomness for generating every random variable corresponding to a node, are presented as a sequence, in the order from root to leaves, layer by layer, from left to right. We define α^x to be

the summation of the random variables corresponding to the first x leaves. Notice that α^x can be sampled efficiently in the following way. Consider the path from the root to the x-th leaf. First we sample the root, which can be computed using the corresponding randomness. We use a variable z to record this sample outcome, calling z a accumulator for convenience. Then we visit each node along the path. When visiting v, assume its parent is w, where α_w has already been sampled previously with outcome a. If v is a left child of w, then we sample α_v conditioned on $\alpha_w = a$. Assume this sampling has outcome b. Then we add $-a + b$ to the current accumulator z. If v is a right child of a node w, then we keep the current accumulator z unchanged. After visiting all nodes in the path, z is the sample outcome for α^x.

Lemma F.2. *The joint distribution $\alpha^x, x = 0, 1, \ldots, N$ has the same distribution as $\langle \omega, \pi_1(x) \rangle, x = 0, 1, \ldots, N$.*

Proof. According to our construction, each leaf is an independent distribution ω_0. Hence if we take all the leaves and form a vector, then it has the same distribution as w.

Notice that for each parent w with two children u, v, by the construction, $\alpha_w = \alpha_u + \alpha_v$. Here α_u, α_v are independent, each being a summation of l independent ω_0, with l being the number of leaves derived from u. Thus for each layer, for every node u in the layer, α_u's are independent and the summation of them is their parent. So for the last layer all the variables are independent and follow the distribution ω_0. And for each node w in the tree, α_w is the summation of the random variables attached to the leaves of the subtree whose root is w. So α^x is the summation of the first x leaf variables. □

We do the same operation for other dimensions of the output of $\pi_1^{(d)}$ and then sum them up to get an alternate construction π' for $\pi = \phi \circ \pi_1^{(d)}$.

We note that to generate an α_v, we only need to simulate the conditional distributions. The distribution function F of the random variable is easy to derive, since its density function is a product of three Gaussian density functions, i.e.

$$\begin{aligned}\Pr_{\alpha_u, \alpha_v \text{ i.i.d.}}[\alpha_u = b \mid \alpha_u + \alpha_v = a] &= \Pr_{\alpha_u, \alpha_v \text{ i.i.d.}}[\alpha_u = b, \alpha_v = a - b] / \Pr_{\alpha_u, \alpha_v \text{ i.i.d.}}[\alpha_u + \alpha_v = a] \\ &= \Pr_{\alpha_u}[\alpha_u = b] \cdot \Pr_{\alpha_v}[\alpha_v = a - b] / \Pr_{\alpha_u, \alpha_v \text{ i.i.d.}}[\alpha_u + \alpha_v = a],\end{aligned}$$

where α_u, α_v are Gaussians. To compute F we can use the taylor expansion of its density function to get an analytical form of F, and the evaluation then can be computed in time $t_\tau = \mathrm{poly}(\rho_0^{-1})$. Recall that ρ_0 is defined to be ρ'/Δ'. To sample α_u, we use $\tau = O(\log \rho_0^{-1})$ uniform random bits to generate a number p uniformly with precision $\mathrm{poly}(\rho_0^{-1})$ small enough. Then we use binary search to figure out an b such that $F(b) \in [p - \varepsilon_0, p + \varepsilon_0]$, for some small enough $\varepsilon_0 = \mathrm{poly}(\rho_0)$. and the space used is $s_\tau = \mathrm{poly}\log(\rho_0^{-1})$.

We remark that simulating a distribution in this way always has some simulating bias. But we can see that the statistical distance between the simulated distribution and the Gaussian is at most $\mathrm{poly}(\rho_0)$, which is small enough when ρ is small enough, since our input are all natural numbers at most N.

F.4 Reducing the Time Complexity Using PRGs

Next we use a pseudorandom generator to replace the randomness used in the above construction. A function $G : \{0,1\}^r \to \{0,1\}^n$ a pseudorandom generator for space s computations with error parameter ε, if for every probabilistic TM M with space s using n bits randomness in the read-once manner

$$|\Pr[M(G(U_r)) = 1] - \Pr[M(U_n) = 1]| \le \varepsilon.$$

Here r is called the seed length of G.

Theorem F.3 (Nisan 1992)**.** *For every $n \in \mathbb{N}$ and $s \in \mathbb{N}$, there exists an pseudorandom generator $G : \{0,1\}^r \to \{0,1\}^n$ for space s computations with parameter ε, where $r = O(\log n(\log n + s + \log \frac{1}{\varepsilon}))$. G can be computed in polynomial time (in n and s) and $O(s \log n)$ space. Moreover, given an index $i \in [n]$, the i-th bit of the output of G can be computed in time $\mathrm{poly}(r)$.*

Let $G : \{0,1\}^r \to \{0,1\}^\ell, \ell = 2dDN\tau$ be a pseudorandom generator for space $s = c_1(\log N + s_\tau)$, with error ε_g, $\tau = c_2 \log N$ for some large enough constants c_1, c_2. Again we only need to consider the construction corresponding to the first output dimension of $\phi \circ \pi_1$. We replace the randomness U_ℓ used in the construction by output of G. That is, when we need τ uniform random bits to construct a distribution α_v in the tree, we first compute positions of these bits in U_ℓ and then compute the corresponding bits in the output of G. Then use them to do the construction in the same way. We denote this mapping using pseudorandomness as our final mapping π^*.

Now we provide a test algorithm to show that the feature mapping provided by the pseudorandom distribution has roughly the same quality as that of the mapping provided by the true randomness. We denote the test algorithm as $T = T_{K,x,y,\varepsilon}$ where $x, y \in \mathbb{R}^d$ and K is a Laplacian kernel with feature mapping φ. T works as the following. Its input is the randomness either being U_ℓ or $G(U_r)$. T first computes $\operatorname{dist}_\varphi(x,y)$. Notice that T actually does not have to compute φ since the distance can be directly computed as $\sqrt{2 - 2K(x,y)}$. Then $T(G(U_r))$ computes $\operatorname{dist}_{\pi^*}(x,y)$ and test

$$|\operatorname{dist}_{\pi^*}(x,y) - \operatorname{dist}_\varphi(x,y)| \le \varepsilon \operatorname{dist}_\varphi(x,y).$$

Notice that when the input is U_ℓ, then this algorithm T is instead testing

$$|\operatorname{dist}_{\pi'}(x,y) - \operatorname{dist}_\varphi(x,y)| \le \varepsilon \operatorname{dist}_\varphi(x,y).$$

Recall that π' is defined in the previous section as our mapping using true randomness.

Next we consider using T on true randomness.

Lemma F.3. $\Pr[T(U_\ell) = 1] \ge 1 - \delta$.

Proof. By Lemma F.2, $\operatorname{dist}_{\pi'}(x,y) = \operatorname{dist}_\pi(x,y)$. By Theorem 1.2 and our setting of parameters,

$$\Pr[|\operatorname{dist}_\pi(x,y) - \operatorname{dist}_\varphi(x,y)| \le \varepsilon \operatorname{dist}_\varphi(x,y)] \ge 1 - \delta.$$

Notice that the event $T(U_\ell) = 1$ is indeed $|\operatorname{dist}_{\pi'}(x,y) - \operatorname{dist}_\varphi(x,y)| \le \varepsilon \operatorname{dist}_\varphi(x,y)$. Hence the lemma holds. □

Now we show that T is actually a small space computation.

Lemma F.4. *T runs in space $c(\log N + s_\tau)$ for some constant c and the input is read-once.*

Proof. The computing of $\operatorname{dist}_\varphi(x,y)$ is in space $O(\log N)$, since $x, y \in \mathbb{N}, x, y \le N$ and the kernel function K can be computed in that space. Now we focus on the computation of π'. We claim that by the construction of α^x in section F.3, π' can be computed using space $O(s_\tau + \log N)$. The procedure proceeds as the following. First it finds the path to the x-th leaf. This takes space $O(\log N)$. Then along this path, for each node we need to compute a distribution α_v. This takes space $O(s_\tau)$. Also notice that since the randomness is presented layer by layer, the procedure only needs to do a read-once sweep of the randomness. T needs to compute π' for both x and y, but this only blow up the space by 2. So the overall space needed is as stated. □

Finally we prove our theorem by using the property of the PRG.

Proof of Theorem F.1. We first show our result assuming $x, y \in \mathbb{N}, x, y \le N$ for an integer N. We claim that π^* is the mapping we want. Notice that by Theorem F.2 and Lemma F.2, we have

$$\Pr[|\operatorname{dist}_{\pi'}(x,y) - \operatorname{dist}_\varphi(x,y)| \le \varepsilon \cdot \operatorname{dist}_\varphi(x,y)] \ge 1 - \delta.$$

Notice that the event is equivalent to $T(U_\ell) = 1$. Hence $\Pr[T(U_\ell) = 1] \ge 1 - \delta$. By Lemma F.4, T runs in space $O(\log N + s_\tau)$ and is read-once. As G is for space $c(\log N + s_\tau)$ for some large enough constant c,

$$|\Pr[T(G(U_r)) = 1] - \Pr[T(U_\ell) = 1]| \le \varepsilon_g,$$

where $r = O(\log(dDNs_\tau/\varepsilon_g)\log(dDN\tau))$. Notice that $T(G(U_r)) = 1$ is equivalent to $|\operatorname{dist}_{\pi^*}(x,y) - \operatorname{dist}_\varphi(x,y)| \le \varepsilon \cdot \operatorname{dist}_\varphi(x,y)$. Thus

$$\Pr[|\operatorname{dist}_{\pi^*}(x,y) - \operatorname{dist}_\varphi(x,y)| \le \varepsilon \cdot \operatorname{dist}_\varphi(x,y)] \ge 1 - \delta - \varepsilon_g.$$

The running time is computed as the following. We only need to consider one dimension of the input data and one output dimension of the mapping, since others can be computed using the same time. So actually we consider the time for sampling α^x. For α^x, recall that we visit the path from the root to the x-th leaf. We don't have to compute the whole output of G, but instead only need to use some parts of the output. For sampling each variable α_v along the path, we use τ bits in the output of G. By Theorem F.3, the computing of each random bit in G's output, given the index of this bit, needs time $\mathrm{poly}(r)$. Locating the τ bits of randomness for generating α_v needs time $O(\log N)$. Generating each of the Gaussian random variable using these random bits needs time t_τ. Summing up these variables takes less time than sampling all of them. After sampling, the cosine and sine function of the RFF can be computed in time $\mathrm{poly}(1/\rho_0) = \mathrm{poly}(\log N)$. There are d input dimensions and D output dimensions. So the total time complexity is $dD\,\mathrm{poly}(\log N, \log \varepsilon_g^{-1})$.

For the case that $x, y \in \mathbb{R}^d$, we only need to modify the embedding $\pi_1^{(d)}$ in the following way. We first round every entry so that their decimal part is now finite. The rounded parts are small enough (e.g. dropping all digits after the $10 \log \rho^{-1}$)-th position to the right of the decimal point) such that this only introduce some small additive errors. Then we shift all the entries to be non-negative numbers by adding a common shift s. Then we multiply every entry of x by a common factor t s.t. every entry now only has an integer part. Notice that t and s can both be chosen according to $\frac{\Delta}{\rho}$, for example $t = s = O(\frac{\Delta}{\rho})$. And we can take N to be $\mathrm{poly}(\frac{\Delta}{\rho})$. Then we apply π_1, and multiply a factor $\sqrt{1/t}$. Denote this map as $\tilde{\pi}_1$. Notice that this ensures that $\|x-y\|_1 = \|\tilde{\pi}_1^{(d)}(x) - \tilde{\pi}_1^{(d)}(y)\|_2^2$. Then we apply the same construction as we did for the above natural number case. The theorem follows by a a similar argument. □

G Remarks and Comparisons to Chen & Phillips (2017)

Our upper bound in Theorem 1.2 is not directly comparable to that of Chen & Phillips (2017) which gave dimension reduction results for Gaussian kernels. Chen & Phillips (2017) showed in their Theorem 7 a slightly improved target dimension bound than ours, but it only works for the case of $\|x - y\| \geq \sigma$, where σ is the parameter in the Gaussian kernel[1]. For the other case of $\|x - y\| < \sigma$, their Theorem 14 gave a related bound, but their guarantee is quite different from ours. Specifically, their target dimension depends linearly on the input dimension d. Hence, when d is large (e.g., $d = \log^2 n$), this Theorem 14 is worse than ours (for the case of $\|x - y\| < \sigma$.

Finally, we remark that there might be subtle technical issues in the proof of [CP17]. Their Theorem 7 crucially uses a bound for moment generating functions that is established in their Lemma 5. However, we find various technical issues in the proof of Lemma 5 (found in their appendix). Specifically, the term $\mathbb{E}[e^{-s\frac{1}{2}\omega^2\|\Delta\|^2}]$ in the last line above "But" (in page 17), should actually be $\mathbb{E}[e^{s\frac{1}{2}\omega^2\|\Delta\|^2}]$. Even if one fixes this mistake (by negating the exponent), then eventually we can only obtain a weaker bound of $\ln M(s) \leq \frac{s^2}{4}\|\Delta\|^4 + s\|\Delta\|^2$ in the very last step, since the term $-s\|\Delta\|^2$ is negated accordingly. Hence, it is not clear if the claimed bound can still be obtained in Theorem 7.

[1] This condition is not clearly mentioned in the theorem statement, but it is indeed used, and is mentioned in one line above the statement in Chen & Phillips (2017).