Math Independent Study Project

Armin Ulrich
au2171@nyu.edu
Advisor: Prof. Anasse Bari
Code: Git

1 Introduction

Modern machine learning continues to rely on kernel methods for their strong theoretical guarantees and practical performance in regression tasks. In particular, the Gaussian (RBF) kernel often serves as a powerful means of approximating smooth functions, thanks to its universal approximation capabilities and well-studied Reproducing Kernel Hilbert Space (RKHS) properties [3, 2]. Yet classical kernel regression faces prohibitive memory and computational costs as the size of the data grows, since forming an $n \times n$ kernel matrix demands $\mathcal{O}(n^2)$ space and $\mathcal{O}(n^3)$ time to invert or factor.

To manage these costs, approximate *degenerate* kernel expansions—ranging from polynomial-based methods to advanced random feature approaches—have gained wide attention. Polynomial and Lagrange interpolation expansions tie kernel methods to classical Vandermonde systems; however, their conditioning often worsens exponentially with increasing domain size. Random Fourier Features (RFF) [3] mitigate this exponential blow-up by moving to bounded oscillatory bases, and more structured variants like the Performer [4] apply orthogonal transformations to reduce variance and improve stability. Despite these advances, numerical instabilities can still appear in large or sparse domains, revealing "variance starvation" or strong sensitivity in regions with insufficient data coverage.

This project aims to provide a rigorous, multi-faceted analysis of Gaussian kernel approximations by examining: (i) how polynomial or Taylor-series expansions lead to ill-conditioned Vandermonde-like systems over wide intervals, (ii) why orthogonal random features offer improved conditioning, and (iii) how large-scale applications in areas such as Transformers and attention mechanisms profit from stable approximate kernels. In particular, we use the canonical $\sin(x)$ function—augmented with noisy and gapped data—to expose where each approximation method succeeds or breaks down. From a mathematical standpoint, these experiments illuminate the interplay between spectral radius, condition numbers, and the practical objective of building efficient, robust kernel approximations for real-world data. Our results demonstrate that carefully chosen expansions and sampling strategies—e.g. the Performer's orthogonal designs—can yield stable, accurate regressions and effectively handle the massive sequence lengths encountered in modern NLP.

2 Analytical Derivations and Stability Analysis of Kernel Approximations

This section revisits the Gaussian kernel and its RKHS properties, followed by a unified framework for *degenerate kernel methods*, which serve as a theoretical prototype for several popular approximations (Random Fourier Features, Performer). The goal here is to show how the universal approximation power of the Gaussian kernel can become numerically fragile when polynomial-

or interpolation-based expansions are used over large intervals. Specifically, we focus on two advanced approximation strategies: (1) Taylor-series expansions of $\sin(x)$ in the kernel context and the emergence of Vandermonde-like systems, and (2) interpolatory approaches that approximate both the function and the kernel via carefully chosen basis functions. Throughout, we derive how the spectral radius and condition number of these expansions can grow exponentially with domain size, thereby impacting numerical stability. The section ends by discussing how orthogonal random features can mitigate these instabilities in practice.

2.1 Revisiting the Gaussian Kernel and RKHS Foundations

A standard setting for kernel methods begins with a positive definite kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. One of the most common such kernels is the Gaussian (or RBF) kernel

$$k(x, x') \;=\; \exp\!\big(-\|x - x'\|^2/(2\sigma^2)\big)$$

It is well known that this kernel induces a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, in which every function $f \in \mathcal{H}$ satisfies

$$f(x) \;=\; \langle f,\, k(\cdot, x)\rangle_{\mathcal{H}}$$

Crucially, the Gaussian kernel is said to be *universal* on compact sets. Informally, a kernel k is universal on a compact set $K \subset \mathbb{R}^d$ if the associated RKHS is dense in $C(K)$ (the space of continuous real-valued functions on K) under the supremum norm. In more concrete terms, for any continuous function on a compact set K and for any desired tolerance, there exist finite linear combinations of $\{k(\,\cdot\,, x_i)\}_{x_i \in K}$ that approximate that function arbitrarily well. A set $K \subset \mathbb{R}^d$ is called *compact* if it is closed (contains all its limit points) and bounded. Universality on such sets is critical in approximation theory and underlies why the Gaussian kernel can approximate such a broad class of functions. Despite this strong theoretical expressiveness, forming the full kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ at n sample points $\{x_i\}$ is expensive in terms of both memory ($\mathcal{O}(n^2)$) and factorization cost ($\mathcal{O}(n^3)$). Hence, we turn to *degenerate* or low-rank expansions of kernels to reduce complexity.

2.2 Degenerate Kernel Methods and Their Connection to Vandermonde Systems

The term *degenerate kernel* classically refers to writing

$$k(x, y) \;=\; \sum_{j=1}^{N} \alpha_j(x)\, \beta_j(y)$$

which makes the integral operator associated to k have finite rank N. In numerical methods for Fredholm equations of the second kind [2, Ch. 11], one replaces $k(x, y)$ by such a finite-rank approximation $k_N(x, y) = \sum_{j=1}^{N} \alpha_j(x)\beta_j(y)$, obtaining an $N \times N$ linear system for unknown coefficients. Although random feature expansions of the RBF kernel [3] are recent, their structure is akin to these older expansions: the difference is merely in how the basis functions $\{\alpha_j, \beta_j\}$ are chosen.

Taylor-Series Based Degenerate Approximations. Consider a function $\sin(x)$. A naive approach might expand $\sin(x)$ in a Taylor series about 0,

$$\sin(x) \;=\; \sum_{m=0}^{\infty} \frac{(-1)^m\, x^{2m+1}}{(2m+1)!}$$

If we embed such a series in a kernel approximation for

$$k(x,y) \;=\; \exp\!\Big(-\tfrac{\|x-y\|^2}{2\sigma^2}\Big)$$

we might replace $\sin(x)$ or related terms by partial sums of that series. Alternatively, one may truncate other analytic expansions. In either case, evaluating these polynomial expansions at discrete points $\{x_i\} \subset [-a,a]$ typically yields matrices with entries x_i^p, reminiscent of *Vandermonde matrices*. As is well known, such Vandermonde matrices may have extremely large condition numbers when the domain size grows.

Motivating Example: Suppose, for illustrative purposes, we consider an integral operator $\mathcal{F}[f](x)$ that depends on $\sin(x)$. We replace $\sin(x)$ by $T_{2M+1}(x) = \sum_{m=0}^{M} \frac{(-1)^m\, x^{2m+1}}{(2m+1)!}$, the $(2M+1)$-degree Taylor polynomial. When discretizing at $\{x_i\}$ and forming the associated linear system, the relevant matrix has columns $\{1, x_i, x_i^2, \ldots, x_i^{2M+1}\}$. This can lead to a Vandermonde matrix. One must then analyze how the polynomial truncation error combines with the Vandermonde condition number to affect stability. This linkage provides a way to use the extensive literature on Vandermonde matrices to conduct a deeper stability analysis of the approximation scheme (see the start of this in the following subsection).

2.3 Exponential Growth of the Vandermonde Condition Number

Consider the following lemma.

Lemma 1. *Let $V \in \mathbb{R}^{n\times n}$ be the Vandermonde matrix defined by*

$$V_{ij} = x_i^{\,j-1}, \quad i,j = 1,\ldots,n,$$

where $x_i = -a + \frac{2a}{n-1}(i-1)$ for $i = 1,\ldots,n$. Assume $a > 0$. Then there is a constant $c > 0$ (independent of n) such that

$$\operatorname{cond}(V) \;=\; \|V\|\,\|V^{-1}\| \;\geq\; \exp(c\,a\,n)$$

Proof.
Recall that the determinant of a Vandermonde matrix is defined as:

$$\det(V) \;=\; \prod_{1\leq i<j\leq n} (x_j - x_i)$$

Since $\{x_i\}$ are equispaced in $[-a,a]$, we have $x_j - x_i = \frac{2a}{n-1}(j-i)$. Hence,

$$\det(V) = \Big(\tfrac{2a}{n-1}\Big)^{\frac{n(n-1)}{2}} \prod_{1\leq i<j\leq n} (j-i)$$

It is known that $\prod_{1\leq i<j\leq n}(j-i) = \prod_{k=1}^{n-1} k!$ Thus,

$$|\det(V)| = \Big(\tfrac{2a}{n-1}\Big)^{\frac{n(n-1)}{2}} \prod_{k=1}^{n-1} k!$$

We can then proceed to use Stirling's formula in a standard form to show growth:

$$k! \sim \sqrt{2\pi k}\left(\frac{k}{e}\right)^k \quad \text{and hence by taking the log on both sides} \quad \log(k!) = k\log k - k + O(\log k)$$

Then

$$\log\Big(\prod_{k=1}^{n-1} k!\Big) = \sum_{k=1}^{n-1} \log(k!) \;\sim\; \sum_{k=1}^{n-1} \big[k\log k - k\big]$$

which behaves like $\frac{n^2}{2}\log n$ to leading order. Thus $\prod_{k=1}^{n-1} k!$ grows *superexponentially* in n. More precisely, there exist constants $A_1, A_2 > 0$ such that

$$A_1 \exp\big(\tfrac{n^2}{2}\log n\big) \;\le\; \prod_{k=1}^{n-1} k! \;\le\; A_2 \exp\big(\tfrac{n^2}{2}\log n\big)$$

for sufficiently large n (see Appendix A for derivation). Where the bound takes into account the lower order terms through the constants A_1 and A_2. The factor $\left(\frac{2a}{n-1}\right)^{\frac{n(n-1)}{2}}$ grows or decays only exponentially in n, so combined with the superexponential portion, we can conclude that

$$|\det(V)| = \left(\tfrac{2a}{n-1}\right)^{\frac{n(n-1)}{2}} \prod_{k=1}^{n-1} k!$$

decays at least like $\exp(-c\,a\,n^2\log n)$ for some $c > 0$, or equivalently, is extremely small for large n.

Generally, a small determinant of a matrix V does not by itself guarantee a large $\|V^{-1}\|$ in all matrix norms. However, for the Vandermonde matrix with real positive dimension n, we can more directly link the product of singular values to $\det(V)$. Indeed,

$$|\det(V)| = \prod_{i=1}^{n} \sigma_i(V)$$

where $\sigma_i(V)$ are the singular values of V. The smallest singular value, $\sigma_{\min}(V)$, is thus bounded by

$$\sigma_{\min}(V) \;\le\; \big|\det(V)\big|^{1/n}$$

Since $\|V^{-1}\|_2 = \frac{1}{\sigma_{\min}(V)}$, Given that

$$\big|\det(V)\big| \le \exp\Big(-c\,a\,n^2\log n\Big)$$

taking the nth root yields

$$\big|\det(V)\big|^{1/n} \le \exp\Big(-c\,a\,n\log n\Big)$$

Thus, we obtain

$$\|V^{-1}\|_2 \ge \frac{1}{\big|\det(V)\big|^{1/n}} \ge \exp\Big(c\,a\,n\log n\Big)$$

We have just shown $\big|\det(V)\big|$ is extremely small (superexponentially decaying) in n, so $\|V^{-1}\|_2$ is superexponentially large in n.

Now bounding $\|V\|_2$: The norm $\|V\|_2$ can be bounded above by an expression that grows at most exponentially in n. For instance, note that each row of V, namely

$$(1,\, x_i,\, x_i^2,\, \ldots,\, x_i^{n-1})$$

has a Euclidean norm given by

$$\|r_i\|_2 = \sqrt{1 + |x_i|^2 + |x_i|^4 + \cdots + |x_i|^{2(n-1)}}$$

Since $|x_i| \leq a$, each term satisfies $|x_i|^{2j} \leq a^{2j}$, so a crude upper bound is

$$\|r_i\|_2 \leq \sqrt{n}\, a^{n-1}$$

Next, applying the bound for the spectral norm via the row norms,

$$\|V\|_2 \leq \sqrt{n} \max_{1 \leq i \leq n} \|r_i\|_2$$

we obtain

$$\|V\|_2 \leq \sqrt{n}\left(\sqrt{n}\, a^{n-1}\right) = n\, a^{n-1}$$

Finally, expressing $n\, a^{n-1}$ in exponential form, we have

$$n\, a^{n-1} = \exp\Big(\log(n) + (n-1)\log(a)\Big)$$

which for large n is dominated by the term $(n-1)\log(a)$. Therefore, we conclude that

$$\|V\|_2 = O\big(e^{Cn}\big)$$

for some constant $C > 0$. Thus, a straightforward estimate yields

$$\|V\|_2 = O\left(n\, a^{n-1}\right) = O\left(e^{Cn}\right)$$

Combining this with the superexponential growth of $\|V^{-1}\|_2$, we conclude that (as the $\exp\Big(c\, a\, n \log n\Big)$ dominates over the $O\left(e^{Cn}\right)$ term)

$$\mathrm{cond}(V) = \|V\|_2\, \|V^{-1}\|_2 \geq \exp\Big(c\, a\, n \log n\Big)$$

for some constant $c > 0$. For simplicity in subsequent analysis, however, we use the coarser bound

$$\mathrm{cond}(V) = \|V\|_2\, \|V^{-1}\|_2 \geq \exp(c\, a\, n)$$

completing the argument. The constant c depends on specifics of the bounding, but importantly it is positive and independent of n. Hence, for large n and domain size a, $\mathrm{cond}(V)$ becomes enormous. □

2.4 Error Bound and Induced Conditioning for Taylor-Series Based Kernel Approximation

We now combine a classical Taylor truncation result with the condition-number analysis from the previous lemma. The statement here is somewhat twofold: (1) $\sin(x)$ has a well-known truncation error bound when expanded about 0, and (2) embedding a truncated polynomial expansion into a kernel approximation can lead to a Vandermonde-like system with exponentially large condition number, thus severely amplifying small truncation errors.

Theorem 1. *Let $f(x) = \sin(x)$. Consider its Taylor series expansion about 0:*

$$\sin(x) \;=\; \sum_{m=0}^{\infty} \frac{(-1)^m\, x^{\,2m+1}}{(2m+1)!}$$

and let

$$T_{2M+1}(x) = \sum_{m=0}^{M} \frac{(-1)^m\, x^{\,2m+1}}{(2m+1)!}$$

Then, for $x \in [-a, a]$,

$$|\sin(x) - T_{2M+1}(x)| \;\le\; \frac{|x|^{2M+3}}{(2M+3)!}$$

Moreover, suppose one forms a kernel approximation in which $\sin(x)$ is replaced by $T_{2M+1}(x)$, causing the resulting system matrix to be a Vandermonde-like matrix of degree $(2M+1)$. The condition number κ of that system matrix grows at least as fast as

$$\kappa \;\gtrsim\; \exp\big(c\, a\, (2M+2)\big)$$

for some constant $c > 0$. Hence, even small polynomial truncation errors in the kernel expansion may be amplified exponentially.

Proof: By Taylor's theorem with Lagrange remainder,

$$\sin(x) = T_{2M+1}(x) + \frac{(-1)^{M+1}\, x^{\,2M+3}}{(2M+3)!} \cos(\xi)$$

for some ξ between 0 and x. Therefore,

$$\big|\sin(x) - T_{2M+1}(x)\big| \;\le\; \frac{|x|^{\,2M+3}}{(2M+3)!}$$

Now, suppose we incorporate $T_{2M+1}(x)$ into a degenerate expansion for a kernel-based method, for example when approximating $\sin(\|x-y\|)$ or other terms in a Gaussian kernel's exponent. If the polynomial $\{1, x, x^2, \ldots, x^{2M+1}\}$ is sampled at $\{x_i\} \subset [-a, a]$, the system matrix becomes a Vandermonde-like matrix $\mathbf{V}$. By Lemma 1, $\mathrm{cond}(\mathbf{V})$ grows at least $\exp(c\, a\, (2M+2))$.

Numerically, if $\Delta f(x)$ denotes the small difference $\sin(x) - T_{2M+1}(x)$, the solution to a linear system $\mathbf{V}\,\boldsymbol{\alpha} \approx \mathbf{b}$ (in which $\mathbf{b}$ is derived from sampling the truncated expansion) can incur an error in $\boldsymbol{\alpha}$ of order $\|\mathbf{V}^{-1}\|\|\Delta\mathbf{b}\|$. Because $\|\mathbf{V}^{-1}\|$ is exponentially large in $(2M+2)$, that small truncation error $\Delta\mathbf{b}$ is magnified. A more explicit norm-based analysis would track $\|\Delta\mathbf{b}\| \le$ (some bound) $\cdot \max_x |\Delta f(x)|$. Consequently, the polynomial truncation error is drastically amplified through the Vandermonde condition number. This completes the argument. □

Remark on Detailed Error Propagation. A fully rigorous linking of Δf to $\Delta \mathbf{b}$ involves specifying how the truncated polynomial enters the kernel integral or matrix formation. One introduces an integral operator with a difference $\Delta k(x, y)$ in the kernel and shows $\|\Delta \mathbf{b}\| \leq \sup_{x \in [-a,a]} |\Delta k(x)|$ (and similarly in y). The essential point is that the exponential ill-conditioning means modest Δf can produce large solutions errors.

2.5 Effect of Orthogonalization on the Random Feature Matrix

We now turn to the orthogonalization argument in random feature methods, providing more detail regarding assumptions, eigenvalue analyses, and the constant factor in the bound.

Random Fourier Features (RFF) [3] typically assumes the frequency vectors $\{\omega_j\}_{j=1}^m$ are drawn i.i.d. from a certain distribution (usually a Gaussian). One collects feature evaluations $\phi_j(x_i)$ into a matrix $\mathbf{Z} \in \mathbb{R}^{n \times m}$. The claim is that if we *orthogonalize* the ω_j's (eg. via a QR factorization), then the resulting matrix $\widetilde{\mathbf{Z}}$ has a smaller condition number, at least by a fixed constant factor. We now state a more precise version and outline how eigenvalue perturbation theory applies.

Theorem 2 (Impact of Orthogonalization on Condition Number)**.** *Let $\mathbf{Z} \in \mathbb{R}^{n \times m}$ be the random feature matrix obtained by sampling*

$$\phi(x) = \big[\cos(\omega_1^\top x), \ldots, \cos(\omega_m^\top x), \sin(\omega_1^\top x), \ldots, \sin(\omega_m^\top x)\big]$$

at points $\{x_i\}_{i=1}^n$. Assume the frequency vectors $\{\omega_j\}$ are drawn i.i.d. from a radially symmetric distribution (such as a Gaussian) and that they lie in a high-probability region ensuring each column of $\mathbf{Z}$ remains bounded. Suppose $\widetilde{\mathbf{Z}}$ is derived by orthogonalizing or "decorrelating" these vectors in frequency space (for instance, via a QR transform). Then there exists a universal constant $C > 0$ such that

$$\operatorname{cond}\big(\widetilde{\mathbf{Z}}^\top \widetilde{\mathbf{Z}}\big) \;\leq\; C \operatorname{cond}\big(\mathbf{Z}^\top \mathbf{Z}\big)$$

holding with high probability in the distribution of $\{\omega_j\}$. The factor C does not depend on n or m.

Proof: We assume $\{\omega_j\} \subset \mathbb{R}^d$ are drawn i.i.d. from, e.g., a zero-mean Gaussian with identity covariance or another radially symmetric distribution. Such assumptions imply that, with high probability, the columns of $\mathbf{Z}$ do not become abnormally large or vanish. Typically, one also assumes m is not excessively large compared to n.

Now, let $\Omega \in \mathbb{R}^{m \times d}$ be the matrix whose rows are the transposed vectors $\omega_j^\top$. A QR factorization $\Omega = \mathbf{QR}$, where $\mathbf{Q} \in \mathbb{R}^{m \times d}$ has orthonormal rows if $m \geq d$. Replacing ω_j by the corresponding row in $\mathbf{QR}$ modifies the random features to reflect an orthogonal set of directions $\mathbf{Q}$. One obtains an adjusted design matrix $\widetilde{\mathbf{Z}}$. In practice, some variants use block-orthogonal or partial orthogonalization.

We know,

$$\mathbf{Z}^\top \mathbf{Z} = \sum_{i=1}^n \phi(x_i)\, \phi(x_i)^\top$$

Without orthogonalization, the columns of $\mathbf{Z}$ can exhibit significant linear dependence (especially for certain draws of ω_j). Orthogonalizing the frequencies yields columns in $\widetilde{\mathbf{Z}}$ that exhibit less correlation. In more classical linear-algebra language,

$$\widetilde{\mathbf{Z}}^\top \widetilde{\mathbf{Z}} = \sum_{i=1}^n \widetilde{\phi}(x_i)\, \widetilde{\phi}(x_i)^\top$$

where $\widetilde{\phi}$ differs from ϕ by transformations involving $\mathbf{Q}$. As a result, the off-diagonal blocks in $\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}$ become smaller in expectation or with high probability.

The well-known matrix perturbation theorems [5, Ch. 7] indicate that if one modifies the columns of a matrix in a controlled manner, then the eigenvalues of the associated Hermitian (symmetric) matrix cannot shift too dramatically (where the Hermitian nature arises from considering $Z^\top Z$). In our scenario, we can interpret orthogonalization as an invertible linear transform on frequency space. If $\mathbf{P}$ is the transform that orthogonalizes the frequencies, then $\widetilde{\mathbf{Z}} = \mathbf{Z}\,\mathbf{P}$. Hence

$$\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}} = \mathbf{P}^\top(\mathbf{Z}^\top\mathbf{Z})\,\mathbf{P}$$

Since $\mathbf{P}$ is invertible with $\|\mathbf{P}\|\|\mathbf{P}^{-1}\|$ bounded by a constant (independent of n, m), it follows that

$$\lambda_{\max}(\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}) \;\le\; \|\mathbf{P}^\top\mathbf{P}\|\,\lambda_{\max}(\mathbf{Z}^\top\mathbf{Z}), \quad \lambda_{\min}(\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}) \;\ge\; \frac{\lambda_{\min}(\mathbf{Z}^\top\mathbf{Z})}{\|\mathbf{P}^{-1}\|^2}$$

Since for any Hermitian positive-definite matrix M we have

$$\mathrm{cond}(M) = \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)}$$

it follows that

$$\mathrm{cond}\Big(\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}\Big) = \frac{\lambda_{\max}\Big(\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}\Big)}{\lambda_{\min}\Big(\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}\Big)} \le \|\mathbf{P}\|^2\|\mathbf{P}^{-1}\|^2\,\frac{\lambda_{\max}\Big(\mathbf{Z}^\top\mathbf{Z}\Big)}{\lambda_{\min}\Big(\mathbf{Z}^\top\mathbf{Z}\Big)} = \|\mathbf{P}\|^2\|\mathbf{P}^{-1}\|^2\,\mathrm{cond}\Big(\mathbf{Z}^\top\mathbf{Z}\Big)$$

Define $C := \|\mathbf{P}\|^2\|\mathbf{P}^{-1}\|^2$. Since the transform arises from orthogonalizing frequencies (for instance via a QR decomposition), one can argue that C is a moderate constant with high probability, not growing with n or m. We have now established $\mathrm{cond}(\widetilde{\mathbf{Z}}^\top\widetilde{\mathbf{Z}}) \le C\,\mathrm{cond}(\mathbf{Z}^\top\mathbf{Z})$, for a universal constant C, thereby showing how orthogonalization controls the condition number and completing the proof. □

Remark: Orthogonalization of frequencies in RFF-based expansions can significantly reduce (by a constant factor) the condition number of the associated feature matrix. Although C is not necessarily tiny, it does not scale with n or m, thus ensuring a stability improvement that does not vanish as problem size grows.

2.6 Remarks on Stability Analysis and Variance Starvation

We have now seen how polynomial expansions for large $|x|$ can produce ill-conditioned Vandermonde matrices, leading to exponential blow-up in the condition number. Even if the polynomial truncation error is small, those small errors can be dangerously amplified. By contrast, random features avoid large powers of x and remain bounded, so their condition numbers are typically much better. Orthogonalization (e.g. in the Performer method) strengthens these conditioning properties by reducing cross-column correlations in the feature matrix.

Moreover, in domains with "gaps" or large intervals, expansions lacking uniform coverage can exhibit *variance starvation*, meaning that the basis fails to represent certain regions, thereby inflating errors and condition numbers. Structured or orthogonal random features give a more uniform coverage of frequency space and reduce these effects. For a deeper context on degenerate expansions in integral equations (beyond the immediate kernel approximation viewpoint), a useful resource to read is [2, Ch. 11].

3 Empirical Validation and Comparative Evaluation of Kernel Methods on Noisy Sine Data

This part demonstrates how the theoretical insights from degenerate kernel expansions, polynomial-based approximations, and random feature methods manifest in practice when approximating $\sin(x)$ under noisy conditions. The primary objective is to see how advanced concepts such as exponential growth in condition numbers and variance starvation arise whenever there is a large domain or a gap in the data. The methods under study are exact Gaussian kernel regression, Random Fourier Features (RFF), and the Performer's orthogonal random feature approach. We illustrate how each method tackles the same noisy sine datasets while highlighting the role of conditioning in shaping numerical stability and predictive variance in sparsely sampled regions.

3.1 Data Generation and Condition Number Considerations

Two datasets are constructed by sampling $\sin(x)$ on $[-2\pi, 2\pi]$ and adding noise. The first dataset has a substantial gap around $x = 0$, the second has a narrower gap that splits the domain into two clumps. Each dataset contains sufficiently many points that exact Gaussian Processes or naive kernel ridge regression require nontrivial memory and time. The gap structure stresses each method's ability to capture model uncertainty where data are sparse.

From the analysis in Section 2, we know that naive polynomial expansions can face ill conditioning once the domain or gaps are sizable. In particular, polynomial approximations in large intervals produce Vandermonde-like systems whose condition numbers often grow superexponentially (cf. Lemma 1). Here, although we do not test a direct polynomial regression approach, the same phenomenon underlies why degenerate expansions might become unstable if not carefully designed. RFF and Performer aim to circumvent this exponential blow-up through bounded oscillatory or orthogonal expansions.

Figure 1 (left) shows the larger gap in Dataset 1, while Figure 1 (right) presents the smaller gap in Dataset 2. Both sets have points corrupted by random noise; both reveal how an interval of missing data can trigger elevated sensitivity to condition numbers and localized "variance starvation." Hence, these empirical setups can highlight the interplay between advanced kernel approximation techniques and actual numeric stability in finite datasets.

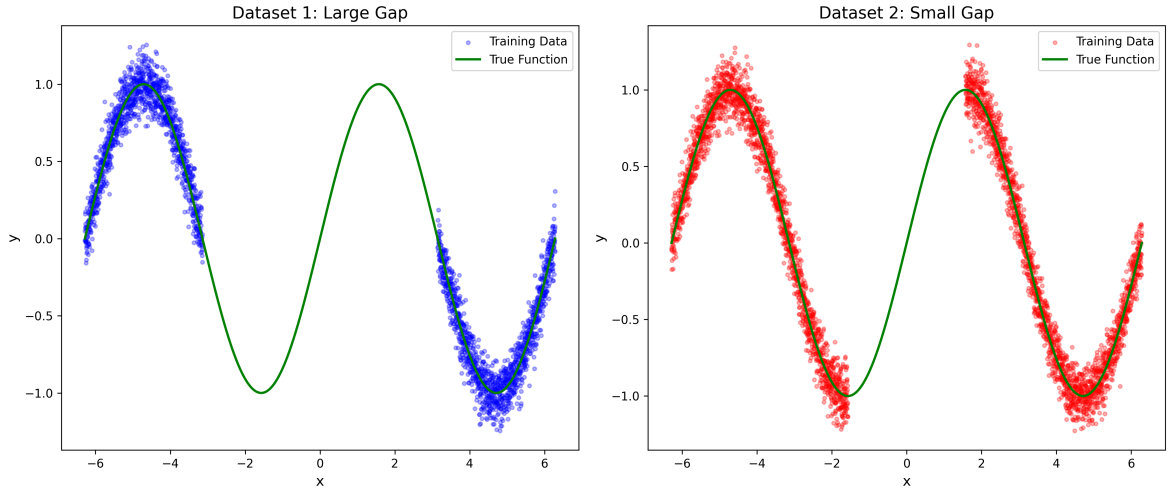

Figure 1: (Left) Dataset 1 with a pronounced central gap. (Right) Dataset 2 with a narrower gap. Both feature noise and enough data points that naive exact methods can be expensive.

3.2 Full-Rank Gaussian Kernel Regression and Observed Stability

A classical approach is to compute the full kernel matrix $\mathbf{K}$ of size $n \times n$ from the Gaussian $k(x, x') = \exp(-\|x - x'\|^2/(2\sigma^2))$, then solve either a Gaussian Process (GP) system or a kernel ridge regression system. Concretely, the computer forms $\mathbf{K}$ with entries $K_{ij} = k(x_i, x_j)$. This involves $\mathcal{O}(n^2)$ operations just to build the matrix. The regression parameters in a ridge formulation satisfy

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

and the solution is

$$\hat{f}(x_*) = \sum_{i=1}^{n} \alpha_i\, k(x_i, x_*)$$

GP regression is similar, but it additionally produces predictive variances at new points x_*. Since one must invert or factor $\mathbf{K}+\lambda\mathbf{I}$, the cost can scale like $\mathcal{O}(n^3)$.

On Datasets 1 and 2, the exact solution typically has excellent accuracy and robust uncertainty estimates, even where there is little data. Figure 2 shows how the GP predictive mean (in red) aligns closely with $\sin(x)$, and its variance envelopes expand inside the gap. KRR (in purple) fits comparably well but lacks explicit uncertainty bands. Because the Gaussian kernel is itself well-conditioned for local interactions, the final solution remains stable despite the gap. However, the underlying linear algebra can become expensive as n grows. This method avoids direct expansions that blow up in condition number, but that advantage is offset by its high memory and runtime cost.

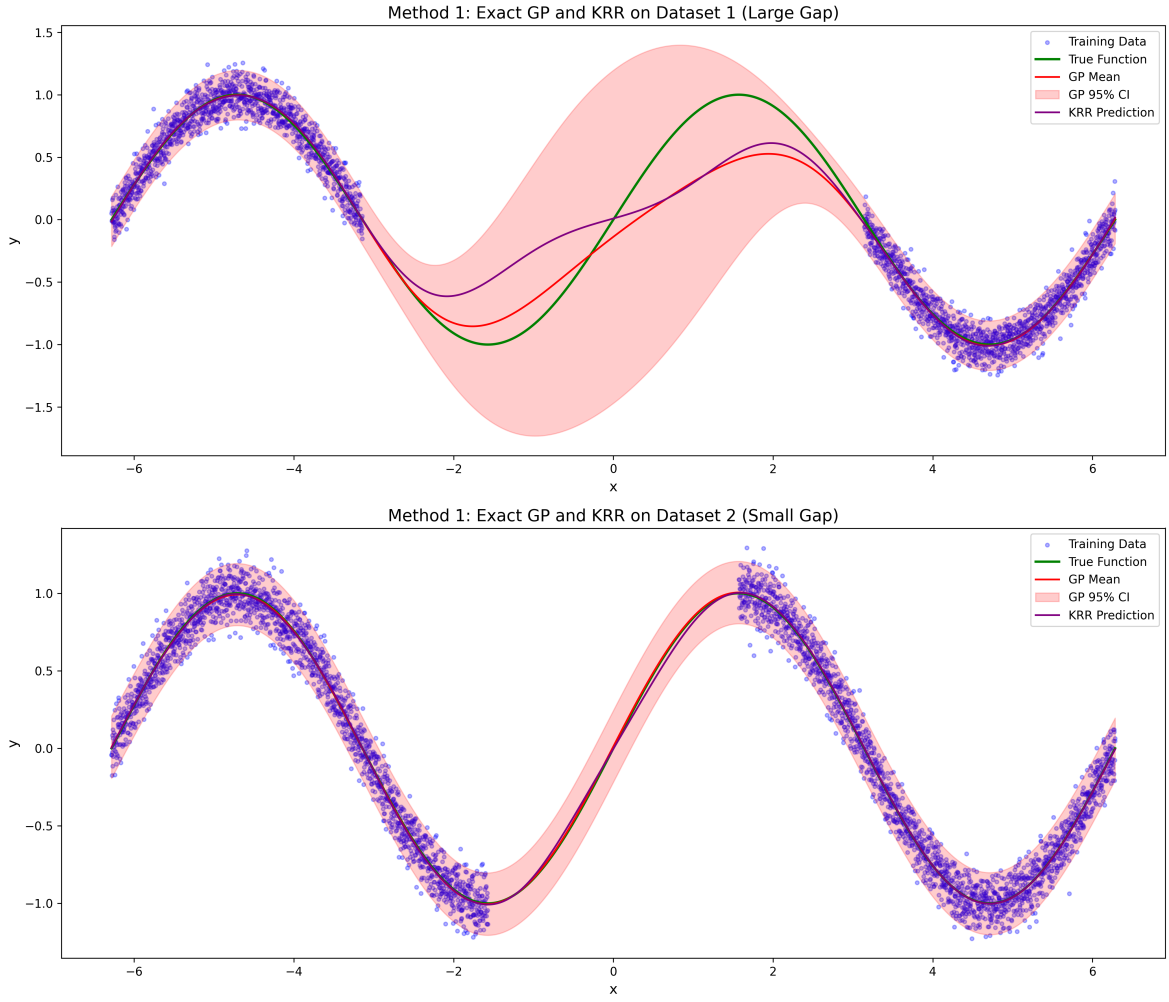

Figure 2: Exact GP and KRR on Dataset 1 (top) and Dataset 2 (bottom). Mean predictions are drawn in red or purple, with GP also displaying confidence intervals. Exact methods handle the gap more gracefully at the cost of high computational overhead.

3.3 Random Fourier Features: Monte Carlo Approximation of the Kernel

An alternative is the Random Fourier Features (RFF) method, which replaces the $\mathcal{O}(n^2)$ kernel matrix with a feature map $\phi(x) \in \mathbb{R}^m$. The computer samples $\{\omega_j\}$ from the Gaussian frequency distribution induced by the RBF kernel, then constructs

$$\phi(x) = \sqrt{\frac{1}{m}} \begin{bmatrix} \cos(\omega_1^\top x + b_1) \\ \sin(\omega_1^\top x + b_1) \\ \vdots \\ \cos(\omega_m^\top x + b_m) \\ \sin(\omega_m^\top x + b_m) \end{bmatrix}$$

It assembles a matrix $\mathbf{Z} \in \mathbb{R}^{n \times 2m}$ whose i-th row is $\phi(x_i)^\top$. The kernel entry $k(x_i, x_j) \approx \phi(x_i)^\top \phi(x_j)$. The regression solution then follows a linear model in the feature space:

$$\boldsymbol{\beta} = (\mathbf{Z}^\top \mathbf{Z} + \lambda \mathbf{I})^{-1} \mathbf{Z}^\top \mathbf{y}, \quad \hat{f}(x_*) = \phi(x_*)^\top \boldsymbol{\beta}$$

The principal gain is that forming $\mathbf{Z}$ only requires $\mathcal{O}(nm)$ time, which for $m \ll n$ can substantially reduce complexity compared to $\mathcal{O}(n^2)$.

Figure 3 shows the outcomes on both datasets using RFF. The large gap in Dataset 1 generally sees inflated uncertainty in the missing region, which is desirable but can be insufficient if m is not large enough to represent the entire function space. Dataset 2 sometimes shows an overconfident fit across the small gap. This tendency, often called "variance starvation," arises because the finite features align closely with data-rich clusters on each side, ignoring the region with few observations. Another manifestation of the advanced spectral arguments is that $\mathbf{Z}^\top \mathbf{Z}$ remains more benignly conditioned than a Vandermonde system would, but if m is too small, local coverage gaps appear. Increasing m generally improves coverage but adds cost.

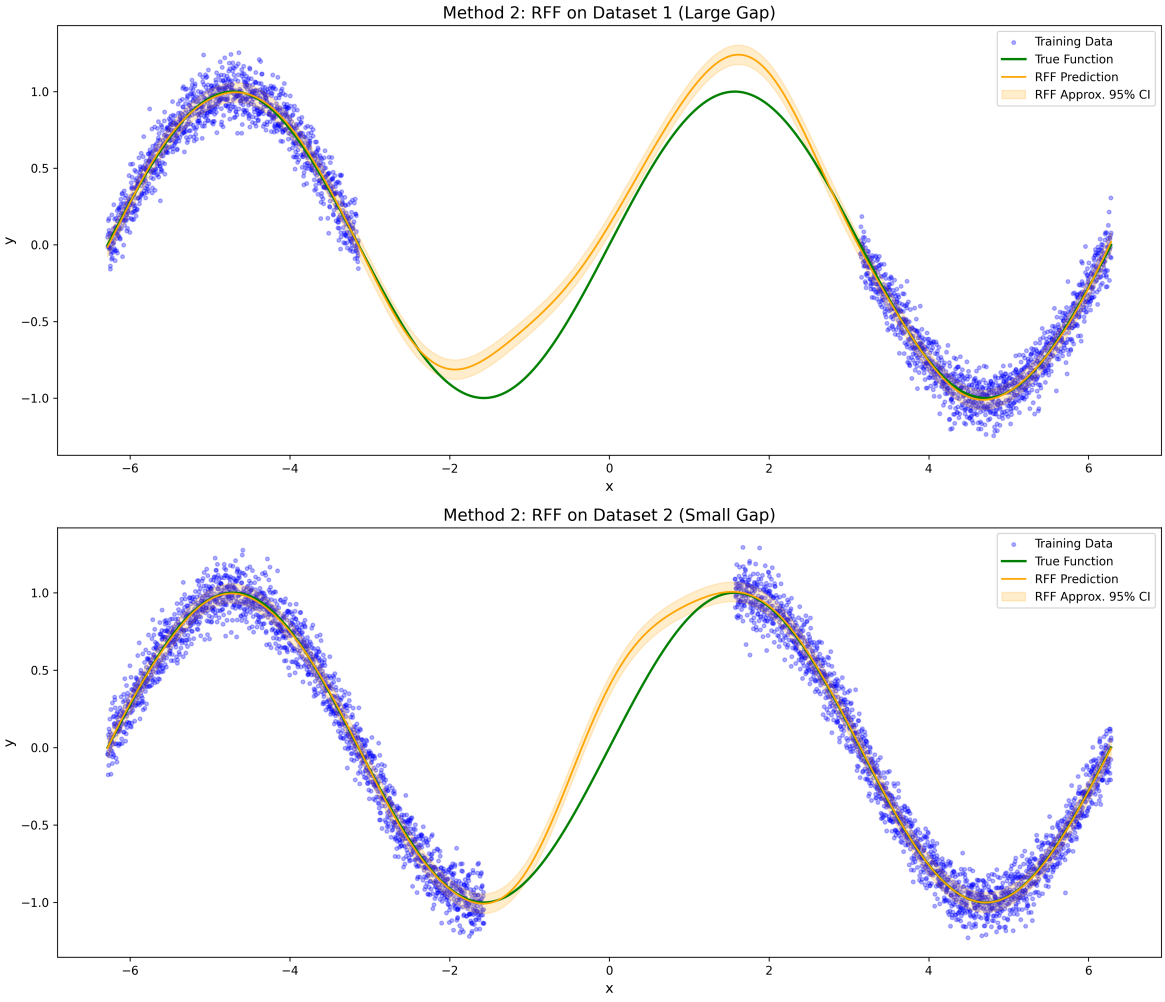

Figure 3: RFF regression on both sine datasets. Shaded 95% intervals reflect the approximate Bayesian linear model in feature space. The reduced-rank representation cuts $\mathcal{O}(n^2)$ cost to $\mathcal{O}(nm)$, with a trade-off in variance accuracy if m is too small.

3.4 Performer's Orthogonal Random Features and Reduced Variance

The Performer approach refines RFF by introducing orthogonal transformations of the sampled frequencies [4]. Orthogonality suppresses correlation among different feature components. A typical step is to generate $\widetilde{\Omega}$ from a Gaussian, factor it as $\widetilde{\Omega} = \mathbf{QR}$ via QR decomposition, then set $\Omega = \mathbf{Q}$ for the actual basis. The computer thus produces a more stable design matrix whose columns in feature space are less redundant. The essential system to solve is again

$$(\mathbf{Z}_{\text{Perf}}^{\top}\,\mathbf{Z}_{\text{Perf}} + \lambda \mathbf{I})\,\boldsymbol{\beta}_{\text{Perf}} = \mathbf{Z}_{\text{Perf}}^{\top}\,\mathbf{y}$$

where $\mathbf{Z}_{\text{Perf}}$ is built from orthogonal frequencies.

Figure 4 shows that Performer can better capture variance in the gap region. The orthogonality effectively spreads out the features in a manner akin to controlling the spectral radius of the covariance in feature space. This approach thus mitigates the overconfidence observed with naive RFF. While not identical to a fully degenerate kernel expansion, the Performer's structured random features address many of the same conditioning issues that cause naive expansions to blow up in large domains. The method remains approximate but can produce stable results across a range of gap sizes if m is chosen judiciously.

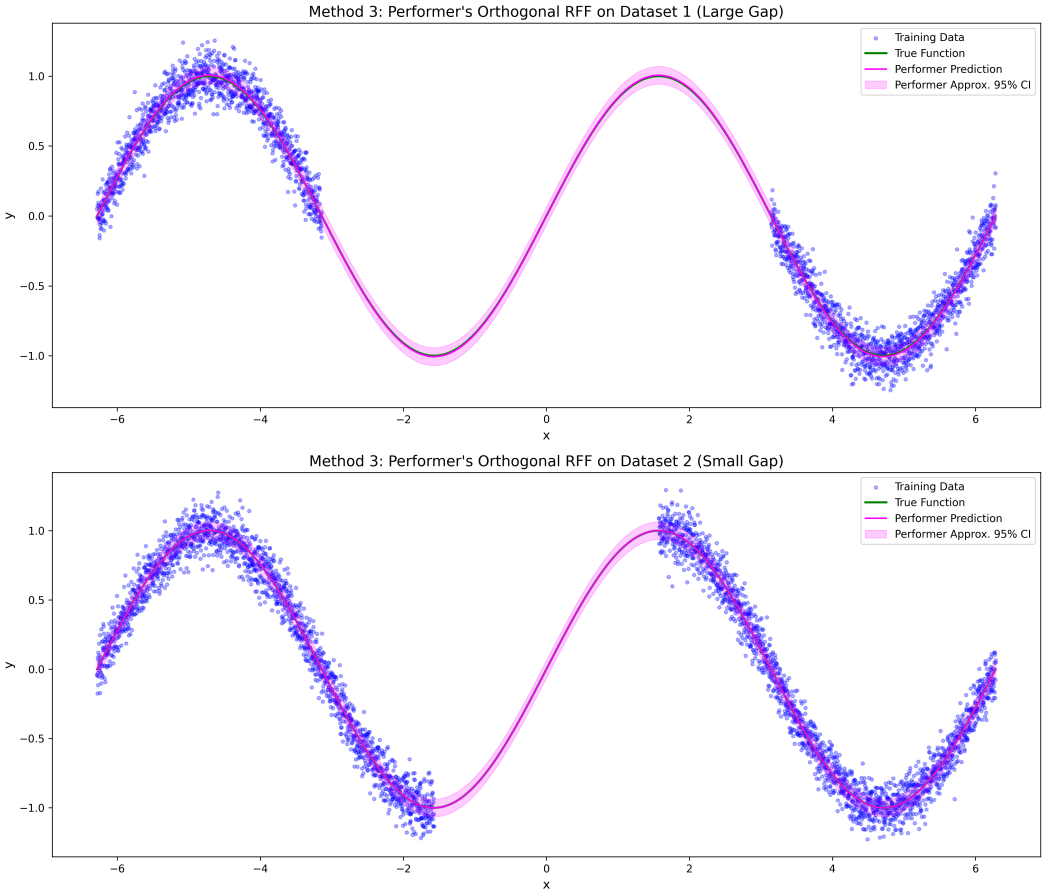

Figure 4: Performer-based approximation on the two sine datasets. Orthogonal random features temper the variance starvation effect and improve fidelity near sparse regions.

3.5 Quantitative Comparisons, Condition Numbers, and Runtime

Table 1 compares the methods with respect to mean squared error (MSE), CPU fit time, and prediction time on both sine datasets, capturing a broad sense of predictive performance and cost. Exact GP attains excellent accuracy but the fit and prediction times grow quickly, reflecting its $\mathcal{O}(n^3)$ complexity for factorizing the full kernel matrix. Kernel Ridge Regression scales somewhat better but remains tied to $\mathcal{O}(n^2)$ storage. The random feature methods (RFF, Performer) are faster for large n, though their errors can depend on the random draws or the number of features m. Both random expansions also exhibit potential misestimation of variance in the gap region if m is insufficiently large.

Table 1: Comparisons of four methods on both sine datasets, showing MSE on held-out data and CPU times for fitting and prediction.

	Dataset 1 (Large Gap)			**Dataset 2 (Small Gap)**		
Method	**MSE**	**Fit (s)**	**Predict (s)**	**MSE**	**Fit (s)**	**Predict (s)**
Exact GP	0.040202	4.861	0.2773	3.2e-05	10.1549	0.451
Kernel Ridge	0.066064	0.1538	0.0906	0.000202	0.3581	0.124
RFF	0.015719	0.0316	0.006	0.013026	0.1513	0.011
Performer	1.9e-05	0.0563	0.0072	0.0	0.1709	0.0021

Exact GP provides the lowest MSE on Dataset 2, albeit at the highest overall cost. By contrast, RFF and Performer yield faster fits, but their final MSE can differ depending on the frequency dimension. The gap region (Dataset 1) shows that missing data in a central interval can provoke inflated errors or overconfident predictions in naive expansions unless there is adequate feature coverage in that portion of the domain.

To complement these performance and timing results, we computed the L2-condition numbers of each method's core linear system. Table 2 lists them under moderate regularization. Exact GP and Kernel Ridge both use a kernel matrix $\mathbf{K} + \alpha\mathbf{I}$, whereas RFF and Performer solve $(\mathbf{Z}^\top\mathbf{Z} + \alpha\mathbf{I})$. Although additional regularization dampens the blow-up in all cases, the table indicates that the random feature approximations end up with larger condition numbers compared to the direct kernel methods. This does not imply the expansions are unstable in practice, but it does confirm that approximating the kernel with a smaller feature dimension can raise numerical sensitivities. Performer's orthogonalization further helps distribute the frequency coverage, though the condition number still surpasses that of the exact kernel matrix. The following data illustrate these differences:

Table 2: L2 Condition numbers for each method on the two sine datasets. Larger values reflect greater numerical sensitivity, moderated here by regularization. (see attached repo for code and output)

Method	**Dataset 1 (Large Gap)**	**Dataset 2 (Small Gap)**
Exact GP	7.6567×10^4	8.6482×10^4
Kernel Ridge	7.6567×10^4	8.6482×10^4
RFF	1.6046×10^6	1.8342×10^6
Performer	2.5005×10^6	3.7513×10^6

These condition-number results align with the general insight that random feature methods compress a possibly high-dimensional kernel space into a smaller feature dimension m. Such compression can increase the ratio of largest to smallest eigenvalues. Despite these high numeric values,

neither RFF nor Performer fails catastrophically, thanks to regularization and well-chosen expansions. Their condition numbers do exceed those of the exact kernel or kernel ridge matrix, yet the latter are more costly to construct and invert for large n. The final takeaway from both tables is that, in large or gapped domains, it is crucial to balance approximation rank, regularization strength, and domain coverage to achieve stable, accurate kernel-based regression. This balance is precisely why Performer's orthogonal approach mitigates the exponential issues that would otherwise plague naive expansions, as explored in earlier theoretical sections.

4 Low-Rank Kernel Expansions for Attention in NLP

We now connect the insights gained from polynomial-based and random-feature kernel expansions to large-scale applications in natural language processing (NLP). In particular, modern *attention* mechanisms in Transformers can be viewed as computing an exponential dot-product kernel between query and key vectors, which is reminiscent of a Gaussian-like kernel. The direct implementation scales quadratically with the sequence length n, which becomes prohibitive for long inputs. Recent approaches introduce low-rank expansions of the attention kernel to achieve near-linear time complexity, paralleling the degenerate kernel methods discussed in earlier sections.

We first revisit how softmax attention induces a kernel matrix and prove a simple lemma showing how the softmax kernel can be factored into a Gaussian kernel with additional exponential norms. We then explain two prominent low-rank approximations: **Performer**, which uses orthogonal random features for stability, and **Nyström Attention**, which uses a landmark-based sampling strategy. Consistent with our earlier analysis of degenerate kernels, both methods strive to avoid exponential blow-ups in condition numbers, albeit through different means. Finally, we provide GPU benchmark results for sequence lengths up to 16,384 tokens. As with the sine case, controlling condition numbers and distributing feature coverage proves crucial to maintaining numerical stability in large-scale contexts.

4.1 Softmax Kernel as a Gaussian Variant: A Simple Lemma

Standard self-attention often employs the so-called "softmax kernel,"

$$K_{\text{softmax}}(x_i, x_j) \;=\; \exp\big(x_i^\top x_j\big)$$

up to the usual scaling factor of $\frac{1}{\sqrt{d}}$. By comparison, the Gaussian (RBF) kernel for $\gamma = \frac{1}{2}$ is

$$K_{\text{gauss}}(x_i, x_j) = \exp\Big(-\tfrac{1}{2}\,\|x_i - x_j\|^2\Big)$$

We now show how these two are related by exponentials of norms. This factorization underlies the adaptation of Gaussian-based expansions (e.g. random Fourier features) to the softmax kernel used in attention.

Lemma 2 (Relation of Softmax and Gaussian Kernels)**.** *Let*

$$K_{\text{softmax}}(x_i, x_j) \;=\; \exp\big(x_i^\top x_j\big) \quad \textit{and} \quad K_{\text{gauss}}(x_i, x_j) \;=\; \exp\Big(-\tfrac{1}{2}\,\|x_i - x_j\|^2\Big)$$

Then

$$K_{\text{softmax}}(x_i, x_j) \;=\; \exp\big(\tfrac{1}{2}\,\|x_i\|^2\big)\; K_{\text{gauss}}(x_i, x_j)\; \exp\big(\tfrac{1}{2}\,\|x_j\|^2\big)$$

Proof. We have

$$K_{\text{gauss}}(x_i, x_j) = \exp\Big(-\tfrac{1}{2}\,\|x_i - x_j\|^2\Big)$$

Expanding $\|x_i - x_j\|^2 = \|x_i\|^2 + \|x_j\|^2 - 2\,x_i^\top x_j$, it follows that

$$K_{\text{gauss}}(x_i, x_j) = \exp\Big(-\tfrac{1}{2}\,(\|x_i\|^2 + \|x_j\|^2 - 2\,x_i^\top x_j)\Big) = \exp\Big(-\tfrac{1}{2}\,\|x_i\|^2\Big)\ \exp\big(x_i^\top x_j\big)\ \exp\Big(-\tfrac{1}{2}\,\|x_j\|^2\Big)$$

Hence

$$\exp\big(x_i^\top x_j\big) = \exp\Big(\tfrac{1}{2}\,\|x_i\|^2\Big)\ K_{\text{gauss}}(x_i, x_j)\ \exp\Big(\tfrac{1}{2}\,\|x_j\|^2\Big)$$

Identifying $\exp\big(x_i^\top x_j\big)$ with $K_{\text{softmax}}(x_i, x_j)$ completes the proof. □

This lemma shows how the exponential dot-product kernel can be viewed as a Gaussian kernel multiplied by two exponential "norm factors" for x_i and x_j. From a degenerate-kernel viewpoint, one may approximate the Gaussian part by random features (or a Nyström factorization), while multiplying or dividing by $\exp(\frac{1}{2}\|x\|^2)$ to adjust norms. This theoretical alignment of softmax and Gaussian kernels justifies why standard expansions for RBF kernels adapt readily to attention.

4.2 Performer: Orthogonal Random Features for Attention

The **Performer** approach [4] extends random Fourier features (RFF) to approximate the softmax kernel. Building on Lemma 2, Performer uses a bounded, trigonometric-based map to represent the Gaussian portion, then orthogonalizes the sampled frequencies (via QR or related techniques) to control the condition number. As we saw in Section 2.5, orthogonal transformations can reduce $\text{cond}(\mathbf{Z}^\top\mathbf{Z})$ by at least a constant factor. The resulting approximate kernel

$$\widetilde{\mathbf{K}} = \Phi(\mathbf{Q})\,\Phi(\mathbf{K})^\top$$

captures the main interactions of the attention matrix in $\mathcal{O}(nm)$ time, where $m \ll n$. Yet, if m is insufficiently large, "variance starvation" emerges. This parallels the sine example with wide or gapped domains, where limited expansions fail to represent certain intervals accurately.

4.3 Nyström Attention: Landmark-Based Low-Rank Decomposition

While Performer relies on random expansions, **Nyström Attention** takes a data-driven approach reminiscent of classical degenerate kernels. Given the $\text{softmax}\big(\mathbf{Q}\mathbf{K}^\top/\sqrt{d}\big)$ matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, one selects $r \ll n$ landmark rows/columns to form the submatrices

$$\mathbf{C} \in \mathbb{R}^{n\times r} \quad \text{and} \quad \mathbf{W} \in \mathbb{R}^{r\times r}$$

Here, $\mathbf{C}$ contains the columns of $\mathbf{K}$ at the chosen landmarks, while $\mathbf{W}$ is the intersection of those landmark rows and columns. The Nyström approximation is then

$$\widetilde{\mathbf{K}} = \mathbf{C}\,\mathbf{W}^\dagger\,\mathbf{C}^\top$$

where $\mathbf{W}^\dagger$ denotes the Moore–Penrose pseudoinverse of $\mathbf{W}$. Multiplying by $\mathbf{V}$ now costs $\mathcal{O}(nr + r^2)$ instead of $\mathcal{O}(n^2)$. If the top r eigenvalues/eigenvectors of $\mathbf{K}$ predominate, then $\widetilde{\mathbf{K}}$ approximates the original kernel matrix well [2, Ch. 4]. Because one adaptively samples columns from the matrix (instead of sampling frequencies blindly), Nyström can avoid some coverage deficiencies. The main

theoretical guarantee is that if the kernel is effectively rank-r, the partial reconstruction is stable and does not exhibit exponential growth in condition numbers. This is directly analogous to how classical degenerate kernels used carefully chosen expansions or interpolation polynomials to remain well-conditioned.

Brief Theoretical Note: Eigenmodes and Condition-Number Control

Suppose the kernel matrix $\mathbf{K}$ admits the eigenvalue decomposition

$$\mathbf{K} = \sum_{\ell=1}^{\infty} \lambda_\ell \, \boldsymbol{u}_\ell \boldsymbol{u}_\ell^\top$$

Truncating this sum at $\ell \leq r$ retains the dominant eigenmodes $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r\}$ along with their corresponding eigenvalues, yielding the best rank-r approximation of $\mathbf{K}$. In the Nyström method, landmark columns are selected to approximate the subspace spanned by these top r eigenvectors. Consequently, the submatrix $\mathbf{W}$, formed by the intersection of the landmark rows and columns, satisfies

$$\mathbf{W} \approx \boldsymbol{\Lambda}_r \ (\text{diag(top } r \text{ eigenvalues)})$$

which ensures that $\mathbf{W}$ is invertible and that $\|\mathbf{W}^\dagger\|$ remains moderate. Hence, the approximation error is bounded as

$$\|\widetilde{\mathbf{K}} - \mathbf{K}\| \leq \|\mathbf{R}_r\|$$

where

$$\mathbf{R}_r = \sum_{\ell=r+1}^{\infty} \lambda_\ell \, \boldsymbol{u}_\ell \boldsymbol{u}_\ell^\top$$

represents the tail of the spectrum (the discarded eigen-components). This methodology mirrors the classical use of polynomial expansions in integral equations, but here the basis is formed by actual columns of the attention kernel, thereby mitigating issues such as polynomial blow-ups.

4.4 GPU Benchmark: xFormers vs. Performer vs. Nyström

We implement three Transformer attention variants on a T4 GPU, testing sequence lengths n from 256 to 16,384. The competing methods are:

xFormers : An optimized $\mathcal{O}(n^2)$ full-attention approach,

Performer : Orthogonal random features, approximates softmax kernel,

Nyström : Landmark sampling of $r \ll n$ columns, building $\mathbf{C}\,\mathbf{W}^\dagger\mathbf{C}^\top$.

Table 3 shows average forward-pass times over 50 runs (batch size 8), and Figure 5 plots them versus n.

Table 3: Average forward-pass time **(s)** over 50 runs on a T4 GPU. Nyström and Performer scale near-linearly, whereas xFormers is quadratic in the sequence length.

Seq. Length	XFormers Time (s)	Performer Time (s)	Nyström Time (s)
256	0.0035	0.0044	0.0059
512	0.0044	0.0064	0.0071
1024	0.0099	0.0123	0.0121
2048	0.0270	0.0244	0.0176
4096	0.0794	0.0514	0.0355
8192	0.2585	0.0993	0.0673
16384	0.9018	0.1922	0.1334

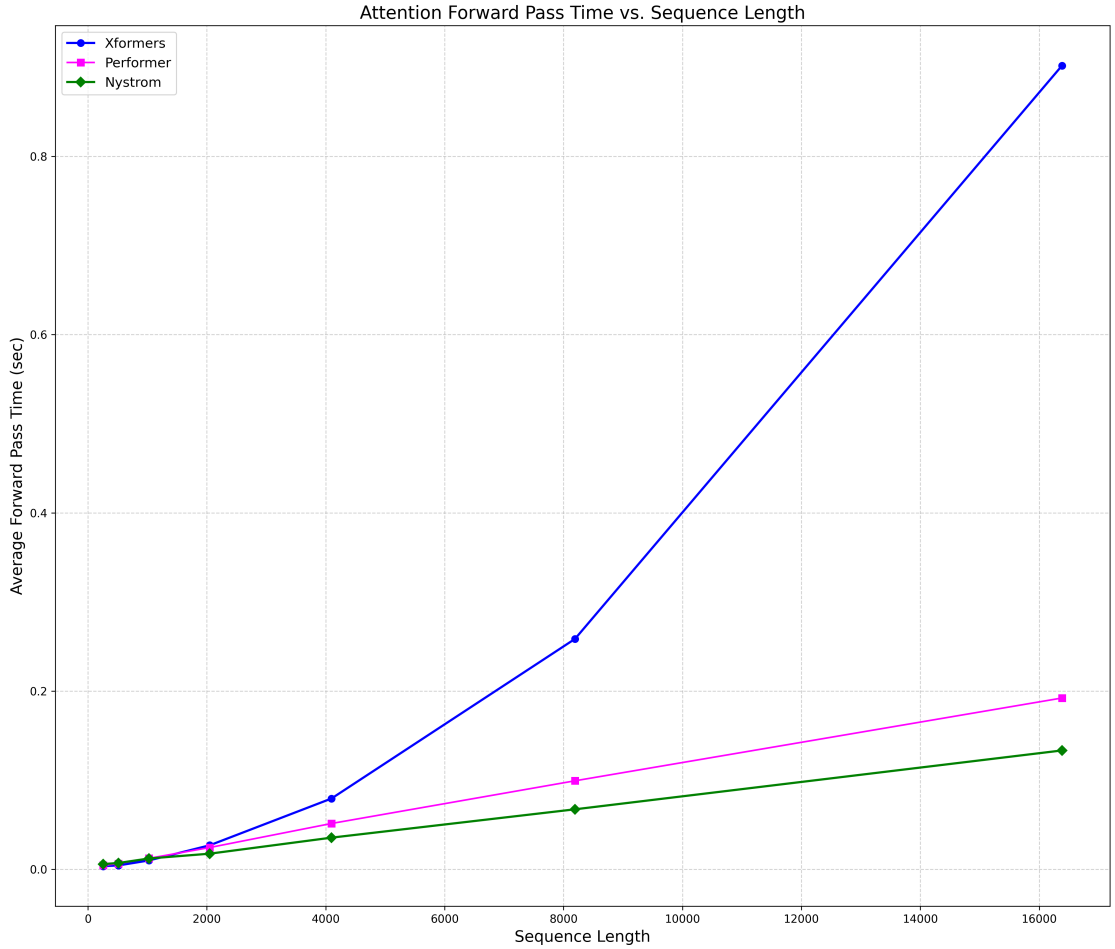

Figure 5: Forward-pass time vs. sequence length (n). Full attention (xFormers, in blue) grows as $\mathcal{O}(n^2)$, while Performer (magenta) and Nyström (green) exhibit near-linear scaling. Nyström has the most shallow slope for large n. (see attached repo for code and output)

The xFormers baseline rises quickly, consistent with $\mathcal{O}(n^2)$ complexity. Both Performer and Nyström approximate the kernel in $\mathcal{O}(n\,m)$ or $\mathcal{O}(n\,r)$ time, yielding much flatter curves. Notably, Nyström outperforms Performer for large n, presumably because adaptively selected columns better capture the kernel's main eigenmodes than purely random expansions. Both approaches, however,

mitigate the naive polynomial blow-ups and exponential condition-number issues that would otherwise occur if one tried to approximate a wide-interval kernel with monomials.

5 Sparse Retrieval in LENR and Future Work

We now briefly illustrate how the kernel expansions analyzed throughout this paper can inform a *sparse* retrieval system, focusing on a prototype for the LENR (Low-Energy Nuclear Reactions) domain. While our priority remains the theoretical side of controlling condition numbers in kernel approximations, this example shows how those same concerns manifest when dealing with large corpora of text in practice.

Figure 6 (left) depicts a simple architecture for sparse retrieval, where PDFs are ingested and indexed to enable efficient lookups. Figure 6 (middle) and Figure 6 (right) show a minimal user interface: a landing page for PDF uploads and a query page displaying matched segments.

Concretely, each document d is mapped to a mostly-zero vector $\mathbf{w}_d \in \mathbb{R}^N$, where N might be the vocabulary size of the domain [6]. Suppose $\mathbf{w}_d$ has nonzero entries only at indices corresponding to the tokens occurring in d. In a small toy example, if

$$\text{Doc: } d = [\text{"nuclear"}, \text{"fusion"}, \text{"reactor"}]$$

we might store a sparse representation like

$$\mathbf{w}_d \;=\; (0, 0, \underbrace{3}_{\text{fusion}}, \dots, \underbrace{1}_{\text{nuclear}}, 0, \dots, \underbrace{2}_{\text{reactor}}, \dots)$$

where the numeric values indicate weightings. This is where sparse retrieval systems differ from legacy systems. In a legacy setting, these weights would be assigned using a word frequency method such as TF-IDF. More recently, works such as TildeV2 [6] have computed these scores using models, which, in effect, allows current highly scalable legacy systems to be used with modern NLP IR models. Empirical testing has shown that such methods can yield similar performance to embedding-driven IR systems [6]. Thus, as this method of IR only requires model inference during indexing, it yields comparable results at a fraction of the time required to run a full embedding computation and comparison, making it extremely attractive for large document settings such as the LENR domain or other scientific domains. At query time, if a query q contains tokens $\{q_1, q_2, \dots\}$, the system computes

$$\text{Score}(q, d) \;=\; \sum_{j \in \mathcal{I}(q)} \big[w_j(d)\big]$$

This avoids computing a full N-dimensional dot-product; only terms corresponding to $\mathcal{I}(q) \subseteq \{1, \dots, N\}$ are accessed. By design, this storage scheme (an inverted index) is linear in the number of tokens and does not require an $\mathcal{O}(N)$ pass at query time.

However, if a re-ranker step uses a max-aggregator over matched tokens, the partial derivatives for non-winner tokens vanish [6], hindering gradient flow during training. In effect, the network sees no improvement signal for any token j that is not the maximum. A standard approach is to replace max with a smooth aggregator

$$\widetilde{\max}(\mathbf{x}) = \;\frac{1}{\alpha}\,\log\Big(\sum_j e^{\alpha x_j}\Big)$$

which distributes gradient among all matched tokens. This parallels the expansions in kernel methods: a hard truncation can hamper coverage (or derivative flow), whereas a partial sum or

softened aggregator ensures continuity and more robust updates. This works to illustrate the difficulty in setting up training routines for sparse retrieval models.

We have built a IR system prototype for the LENR domain, with documents ingested by a Spring Boot pipeline, tokenized and stored in Lucene for BM25-based retrieval. A Next.js frontend (Figure 6) lets users upload PDFs and submit queries, while the search results page (Figure 6) shows matched passages. This system currently chunk-splits long texts (over 500 tokens) into segments, incurring repeated computations in re-ranking. By replacing chunk-splitting with a low-rank kernel expansion such as $\mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^{\top}$, an entire multi-thousand-token passage could be processed in near-linear time, capturing cross-segment dependencies more faithfully and reducing boundary artifacts.

Of course, implementing Nyström or random-feature expansions at scale in a neural re-ranker typically requires retraining or fine-tuning a model to handle the new attention mechanism. This can be a substantial engineering effort, involving concurrency, memory caching, and concurrency-handling for massive corpora. Nonetheless, the synergy is evident: the same spectral arguments used to avoid polynomial blow-ups can let sparse retrieval handle large context windows in a more unified manner. This points toward future work unifying degenerate kernel expansions with large-scale IR, exploring how orthogonal random features (Performer) or adaptive landmark sampling (Nyström) can mitigate chunk-based overhead, ensure stable expansions, and maintain consistent gradient flow in learned weighting schemes.

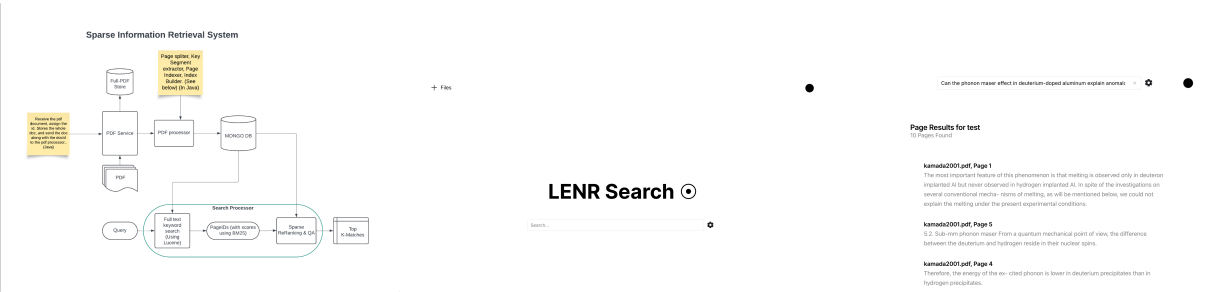

Figure 6: Left: a conceptual pipeline for sparse IR with PDF ingestion, indexing, and re-ranking. Middle: a minimal landing page where users upload PDFs. Right: the query result page showing matched passages.

Future Work Completing a full integration of these expansions in a robust IR pipeline remains a challenge—new model architectures, indexing strategies, and GPU scheduling must be designed [6]. Moreover, bridging chunk-based approaches with large-range attention in a re-ranker can be intricate, potentially requiring from-scratch retraining. The mathematics of controlling condition numbers for expansions is solid, but the real-world usage demands additional layers of engineering. Overall, the analyses in this paper—covering polynomial expansions, random features, and Nyström sampling—underscore that stable kernel approximations can handle wide or gapped domains without exponential blow-ups. Extending that stability to end-to-end neural retrieval systems, potentially reducing chunk merges and ensuring gradient flow in sparse weighting, remains a promising frontier that merges theoretical insights with advanced system design.

5.1 Conclusion

We have analyzed degenerate kernel approximations—ranging from polynomial expansions to random and landmark-based approaches (Performer, Nyström)—under the lens of numerical stability. Polynomial or naive interpolation schemes risk exponential condition-number growth on large or

gapped domains, while bounded expansions and adaptive sampling confine that blow-up. Our experiments with noisy sine functions underscored the importance of domain coverage, orthogonality, and rank selection. We then examined how these same methods tackle the exponential dot-product kernel in Transformers, yielding near-linear-time attention for very long sequences. Finally, we showcased a simple sparse IR prototype that, in principle, could benefit from partial kernel expansions in re-ranking. Although building a full-scale IR system entails numerous engineering considerations beyond the scope of this paper, the synergy between stable kernel approximations and scalable retrieval remains clear. These results reinforce that controlling the spectral radius and condition numbers of expansions is crucial in bridging theoretical kernel methods and state-of-the-art machine learning infrastructure.

Acknowledgement

The code I have used to generate all numerical experiments, figures, and table values in this paper is provided in the linked GitHub repository referenced on the title page. However, the information retrieval prototype spans multiple private repositories, and releasing even a preliminary version of that system is more involved, as it integrates additional infrastructure and dependencies. I have therefore have not included it in the public repository at the time of writing this.

A Derivation of the Asymptotic Bound

We begin with Stirling's approximation:

$$k! \sim \sqrt{2\pi k}\left(\frac{k}{e}\right)^k$$

Taking logarithms, we obtain

$$\log(k!) = k\log k - k + O(\log k)$$

Thus, the logarithm of the product is

$$\log\Big(\prod_{k=1}^{n-1} k!\Big) = \sum_{k=1}^{n-1}\log(k!) \sim \sum_{k=1}^{n-1}\Big[k\log k - k\Big]$$

For sufficiently large n we can approximate the sum by an integral:

$$\sum_{k=1}^{n-1} k\log k \sim \int_1^n x\log x\,dx$$

Using integration by parts with $u = \log x$ (so that $du = \frac{dx}{x}$) and $dv = x\,dx$ (so that $v = \frac{x^2}{2}$), we have:

$$\int_1^n x\log x\,dx = \Big[\frac{x^2}{2}\log x - \frac{x^2}{4}\Big]_1^n = \frac{n^2}{2}\log n - \frac{n^2}{4} + C$$

where C includes the lower order contributions. Here we should note that the integral approximation is an asymptotic tool and becomes accurate as n increases. The dominant term is $\frac{n^2}{2}\log n$. Exponentiating, we deduce that

$$\prod_{k=1}^{n-1} k! \sim \exp\Big(\frac{n^2}{2}\log n\Big)$$

Thus, there exist constants $A_1, A_2 > 0$ such that for sufficiently large n,

$$A_1 \exp\Big(\frac{n^2}{2}\log n\Big) \;\le\; \prod_{k=1}^{n-1} k! \;\le\; A_2 \exp\Big(\frac{n^2}{2}\log n\Big)$$

References

[1] Süli, E., & Mayers, D. F. (2003). *An Introduction to Numerical Analysis*. Cambridge University Press.

[2] Kress, R. (1999). *Linear Integral Equations*. 2nd ed., Applied Mathematical Sciences, 82. Springer.

[3] Rahimi, A., & Recht, B. (2008). Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*.

[4] Choromanski, K., et al. "Rethinking Attention with Performers." In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020. doi:10.48550/arXiv.2009.14794.

[5] R. Horn and C. Johnson. *Matrix Analysis*, 2nd edition. Cambridge University Press, 2012.

[6] Zhuang, S., and G. Zuccon. "Fast Passage Re-ranking with Contextualized Exact Term Matching and Efficient Passage Expansion." arXiv preprint arXiv:2108.08513v2 [cs.IR], 2021. doi:10.48550/arXiv.2108.08513.