Randomized Linear Algebra with Sparse Sketching Matrices

Lukas Florian Münzel

Department of Mathematics
ETH Zurich

Under the direction of

Petar Nizić-Nikolac
Department of Mathematics, ETH Zurich

Prof. Dr. Afonso S. Bandeira
Department of Mathematics, ETH Zurich

March 15, 2026

Abstract

This thesis reviews the 2025 paper *Faster Linear Algebra Algorithms with Structured Random Matrices* by Camaño et al. [1]. Concretely, we study the Oblivious Subspace Injection (OSI) condition on sketching matrices. First, we prove that such OSI matrices can be used for fast, provably reliable randomized linear algebra algorithms for SVD, inversion of PSD matrices, and least squares. We then construct the very sparse *SparseStack* matrix and, following the argument presented in Camaño et al. [1], prove that it is OSI.

1 Introduction

It is of significant practical interest to accelerate operations in linear algebra. One class of approaches that have proven particularly fruitful utilize random mappings to lower dimensional spaces which, with high probability, preserve the structure of the higher-dimensional space.

But what structure exactly should be maintained? Classic results require a sketching matrix $\Omega \in \mathbb{F}^{n\times k}$, $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$ that neither overly stretches nor compresses vectors as they get mapped from $\mathbb{F}^n$ to $\mathbb{F}^k$ [2, 1, 3]. Rigorously, we fix some $r < n, \alpha \in (0,1), \beta \in (1,\infty)$ and require the *oblivious subspace embedding* (OSE) property:

For every subspace $V \subset \mathbb{F}^n$ of dimension r, the following inequality holds with probability at least $\frac{19}{20}$:

$$\forall x \in V \,:\, \alpha \|x\|_2^2 \leq \|\Omega^* x\|_2^2 \leq \beta \|x\|_2^2$$

It is known that these bounds hold for sketching matrices with iid Gaussian entries [2]. Furthermore, we can use these sketching matrices to find approximate solutions to, for example, SVD [4] or least squares [2].

The 2025 paper *Faster Linear Algebra Algorithms with Structured Random Matrices* by Camaño et al. [1] that this thesis aims to review introduces a looser condition, the *Oblivious subspace injection* property: For *Oblivious subspace injections*, we still limit how much vectors may be compressed. However, instead of limiting how much any sketching matrix may expand space, we only require that our random sketching matrices don't distort lengths on average:

Definition 1.1. *A random matrix $\Omega \in \mathbb{F}^{n\times k}$ is called (r,α)-OSI if*

1. *Isometry: For every $x \in \mathbb{F}^n$, we have $\mathbb{E}[\|\Omega^* x\|_2^2] = \|x\|_2^2$*
2. *Injectivity: For every subspace $V \subset \mathbb{F}^n$ of dimension r, the following inequality holds with probability at least $\frac{19}{20}$:*

$$\forall x \in V \,:\, \|\Omega^* x\|_2^2 \geq \alpha \|x\|_2^2$$

We now present the following two core contributions of Camaño et al. [1]:

1. Using embedding matrices which are merely *oblivious subspace injections (OSI)*, we implement randomized, provably reliable versions of SVD (Section 2.2), inversion of PSD matrices (Section 2.3), and least squares (Section 2.4)

2. The construction of a class of sparse matrices that are provably *OSI*. These need embedding dimension $k = \mathcal{O}(r)$ and sparsity $\zeta = \mathcal{O}(\log(r))$, matching theoretical lower-bounds for OSEs [1][5, Theorem 7] and improving on similar known constructions for OSEs with embedding dimension $k = \mathcal{O}(r)$ that require $\zeta = \mathcal{O}(\log^3(r))$ [1]

1.1 Construction of an OSI matrix

We shall already introduce the class of sparse matrices which we shall later prove to be oblivious subspace injections (OSI).

These so-called *SparseStack* matrices were originally introduced by Nelson and Nguyen [6] in 2013. They can be viewed as scaled, vertically stacked independent CountSketch matrices, which were in turn introduced by Charikar, Chen, and Farach-Colton [7]

Definition 1.2. *Let b and n be positive integers. Then the CountSketch [7] matrix $\Phi \in \mathbb{R}^{n\times b}$ is defined by:*

$$\Phi = \begin{bmatrix} \rho_1 e^*_{s_1} \\ \rho_2 e^*_{s_2} \\ \vdots \\ \rho_n e^*_{s_n} \end{bmatrix}$$

for $s_1, \ldots, s_n \overset{\text{iid}}{\sim} \mathsf{Unif}(1, \ldots, \mathsf{b})$ and $\rho_1, \ldots, \rho_n \overset{\text{iid}}{\sim} \mathsf{Rademacher}$

Definition 1.3. *Let b, n, ζ and $k = \zeta b$ be positive integers. We define the SparseStack [6] matrix with sparsity level ζ and embedding dimension k as $\Omega \in \mathbb{R}^{n\times k}$ with*

$$\begin{aligned} \Omega &= \frac{1}{\sqrt{\zeta}} \begin{bmatrix} \Phi_1 & \Phi_2 & \cdots & \Phi_\zeta \end{bmatrix} \\ &= \frac{1}{\sqrt{\zeta}} \begin{bmatrix} \rho_{11} e^*_{s_{11}} & \rho_{12} e^*_{s_{12}} & \cdots & \rho_{1\zeta} e^*_{s_{1\zeta}} \\ \rho_{21} e^*_{s_{21}} & \rho_{22} e^*_{s_{22}} & \cdots & \rho_{2\zeta} e^*_{s_{2\zeta}} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} e^*_{s_{n1}} & \rho_{n2} e^*_{s_{n2}} & \cdots & \rho_{n\zeta} e^*_{s_{n\zeta}} \end{bmatrix} \end{aligned}$$

for $s_{i,j} \overset{\text{iid}}{\sim} \mathsf{Unif}(1, \ldots, \mathsf{b})$ and $\rho_{i,j} \overset{\text{iid}}{\sim} \mathsf{Rademacher}$

2 Linear Algebra algorithms with OSI matrices

We now wish to use (r, α)-OSI matrices to compute linear algebra operations much faster. Concretely, we will

1. Describe algorithms that significantly speed up the calculation of common linear algebra operations. We present such algorithms for Singular Value Decomposition (Section 2.2), Nyström approximation (Section 2.3), least squares (Section 2.4)

2. For each algorithm, we prove that there exists a constant $C > 0$ such that the error of the solution found by it divided by the error of the best rank-r solution is with high probability bounded by $\frac{C}{\alpha}$

3. Demonstrate the empirical performance of the algorithms we present

2.1 Preliminaries

Let us first introduce and prove the following lemma on which all theoretical bounds in this section rely:

Lemma 2.1. *Let $\Omega \in \mathbb{F}^{n\times k}$ be (r,α)-OSI and $Q \in \mathbb{F}^{n\times r}$, $Q_\perp \in \mathbb{F}^{n\times l}$ orthogonal (i.e. $Q^*Q = I_r$ and $Q_\perp^* Q_\perp = I_l$) with mutually orthogonal ranges (i.e. $Q^*Q_\perp = 0$)*

Then, with probability at least 0.9,

$$\left\|B(Q_\perp^*\Omega)(Q^*\Omega)^\dagger\right\|_F^2 \leq \frac{C}{\alpha}\left\|B\right\|_F^2 \tag{1}$$

with $\sigma_{min}^2(Q^*\Omega) \geq \alpha$

We first note that, in general, $\|AB\|_F^2 \leq \|A\|_F^2 \|B\|_{\text{op}}^2$. Indeed, for $AB \in \mathbb{R}^{n\times m}$ with a_p being the p-th row vector of A, we get that

$$\begin{aligned}\|AB\|_F^2 &= \sum_{p=1}^{n}\|a_pB\|_2^2 = \sum_{p=1}^{n}\|B^Ta_p^T\|_2^2\\ &\leq \|B\|_{\text{op}}^2\sum_{p=1}^{n}\|a_p\|_2^2\\ &= \|A\|_F^2\|B\|_{\text{op}}^2\end{aligned}$$

Thus,

$$\left\|B(Q_\perp^*\Omega)(Q^*\Omega)^\dagger\right\|_F^2 \leq \left\|B(Q_\perp^*\Omega)\right\|_F^2\left\|(Q^*\Omega)^\dagger\right\|_{\text{op}}^2$$

We analyze each term separately.

For $\left\|(Q^*\Omega)^\dagger\right\|_{\text{op}}^2$, note that if $\sigma_{\min}^2(Q^*\Omega) > 0$, we get

$$\left\|(Q^*\Omega)^\dagger\right\|_{\text{op}}^2 = \frac{1}{\sigma_{\min}^2(Q^*\Omega)}$$

Now, we note that Range(Q) has dimension r. Then, applying the definition of OSI yields that, with probability at least $\frac{19}{20}$,

$$\sigma_{\min}^2(Q^*\Omega) = \lambda_{\min}(Q^*\Omega\Omega^*Q) \geq \alpha$$

Thus, with probability at least $\frac{19}{20}$, $\left\|(\Omega^*Q)^\dagger\right\|_{\text{op}}^2 \leq \frac{1}{\alpha}$

For $\|B(Q_\perp^*\Omega)\|_F^2$, we note that

$$\begin{aligned}
\mathbb{E}[\|B(Q_\perp^*\Omega)\|_F^2] &= \mathbb{E}[\|\Omega^* Q_\perp B^*\|_F^2] \\
&= \|Q_\perp B^*\|_F^2 && \text{(by isometry)} \\
&= \mathrm{Tr}(BQ_\perp^* Q_\perp B^*) \\
&= \mathrm{Tr}(BB^*) && \text{(by orthogonality of } Q_\perp\text{)} \\
&= \|B\|_F^2
\end{aligned}$$

Thus, by Markov's inequality, with probability at least $\frac{19}{20}$,

$$\|B(Q_\perp^*\Omega)\|_F^2 \leq 20\,\|B\|_F^2$$

We combine both claims with union bound to yield that, with probability at least 0.9,

$$\left\|B(Q_\perp^*\Omega)(Q^*\Omega)^\dagger\right\|_F^2 \leq \frac{20}{\alpha}\,\|B\|_F^2$$

2.2 Randomized Singular Value Decomposition

We wish to compute approximations to the singular value decomposition of a matrix $A \in \mathbb{R}^{n\times d}$. Our strategy is to use an OSI matrix $\Omega \in \mathbb{R}^{d\times k}$ to project Range(A) onto the lower-dimensional space Range($A\Omega$):

Algorithm 1 Randomized SVD

Input: $A \in \mathbb{F}^{n\times d}$ and a sketching matrix $\Omega \in \mathbb{F}^{d\times k}$
Output: $\hat{A} = U\Sigma V^*$ an approximation of the SVD of A

1: $Q \leftarrow \mathrm{orth}(A\Omega)$
2: $B \leftarrow Q^*A$
3: $[\hat{U}, \Sigma, V^*] \leftarrow \mathrm{svd\_econ}(B)$
4: $\hat{U} \leftarrow QU$
5: **return** $[\hat{U}, \Sigma, V^*]$

Imported Lemma 2.1. *Let $A \in \mathbb{F}^{n\times d}, \Omega \in \mathbb{F}^{d\times k}$. Additionally, let $Q \in \mathbb{F}^{n\times k}$ be orthogonal with column range containing that of $A\Omega$. Write the SVD of A as*

$$U \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix}$$

where $\Sigma_1 \in \mathbb{R}^{r\times r}$ for some $r \leq k$. Define $\hat{A} = QQ^*A$

Then, if $V_1^*\Omega$ has full row rank,

$$\left\|A - \hat{A}\right\|_F^2 \leq \|\Sigma_2\|_F^2 + \left\|\Sigma_2(V_2^*\Omega)(V_1^*\Omega)^\dagger\right\|_F^2$$

Theorem 2.1. *Let $\hat{A}$ be the RSVD obtained through algorithm 1 with a sketching matrix $\Omega \in \mathbb{F}^{d\times k}$ that is (r,α)-OSI. Furthermore, we define $[A]_r$ to be the best rank-r approximation*

of A (w.r.t. the Frobenius norm). Then we find some constant $\tilde{C} > 0$ that depends on neither A nor α for which we have, with a probability at least $\frac{9}{10}$,

$$\left\|A - \hat{A}\right\|_F^2 \leq \frac{\tilde{C}}{\alpha} \|A - [A]_r\|_F^2$$

We apply imported Lemma 2.1 and an argument analogous to Lemma 2.1 to yield that, with probability at least 0.9,

$$\begin{aligned}\left\|A - \hat{A}\right\|_F^2 &\leq \|\Sigma_2\|_F^2 + \left\|\Sigma_2 (V_2^* \Omega)(V_1^* \Omega)^\dagger\right\|_F^2 \\ &\leq \|\Sigma_2\|_F^2 + \frac{C}{\alpha} \|\Sigma_2\|_F^2 \\ &\leq \frac{\tilde{C}}{\alpha} \|\Sigma_2\|_F^2 = \frac{\tilde{C}}{\alpha} \|A - [A]_r\|_F^2\end{aligned}$$

We compare this to the results other authors have obtained for Gaussian sketching matrices:

Imported Theorem 2.1. *Suppose we perform RSVD as described, but with a sketching matrix $\Omega \in \mathbb{R}^{d \times k}$ with iid, zero-centered Gaussian entries. Then Theorem 3.6 from* Kireeva *and* Tropp *[8] shows that for $r \leq \min(n, d), k \geq r + 2$*

$$\mathbb{E}\|A - \hat{A}\|_F^2 \leq (1 + \frac{r}{k - r - 1}) \cdot \sum_{i > r} \sigma_i^2(A)$$

Thus, by Markov's inequality, we get embeddings with error that is within a constant of the best rank r approximations with probability $\frac{19}{20}$ for embedding dimension $k = 2r$. This is in fact a better constant than the $k = 128r$ we obtain for SparseStack matrices in Section 4, though in both cases we get that k need only be linear in r.

2.2.1 Numerical experiments

We wish to compare the performance of exact SVD and RSVD with iid Gaussian as well as SparseStack sketching matrices. The matrices on which we perform the (randomized) SVD come from the SuiteSparse [9] collection test matrices, mirroring what Camaño et al. [1] used for their first figure.

We report error ratios compared to those obtained with Gaussian sketching matrices of matching dimensions - Gaussian matrices are considered to be the gold-standard sketching matrices for output quality [1, Page 1][10]. We note that speed improvements from going from Gaussians to SparseStack are very limited. However, SparseStack matrices, despite their extreme sparsity (we use $\zeta = 4$!) suffer essentially no loss in accuracy compared to Gaussian sketching matrices. Note that there is in fact also variance when comparing the performance of the RSVD runs with two independent Gaussian sketching matrices, as visible in the bottom left part of figure 1.

Furthermore, we do see a very clear speedup of one to two orders of magnitude by switching from exact to random singular value decompositions.

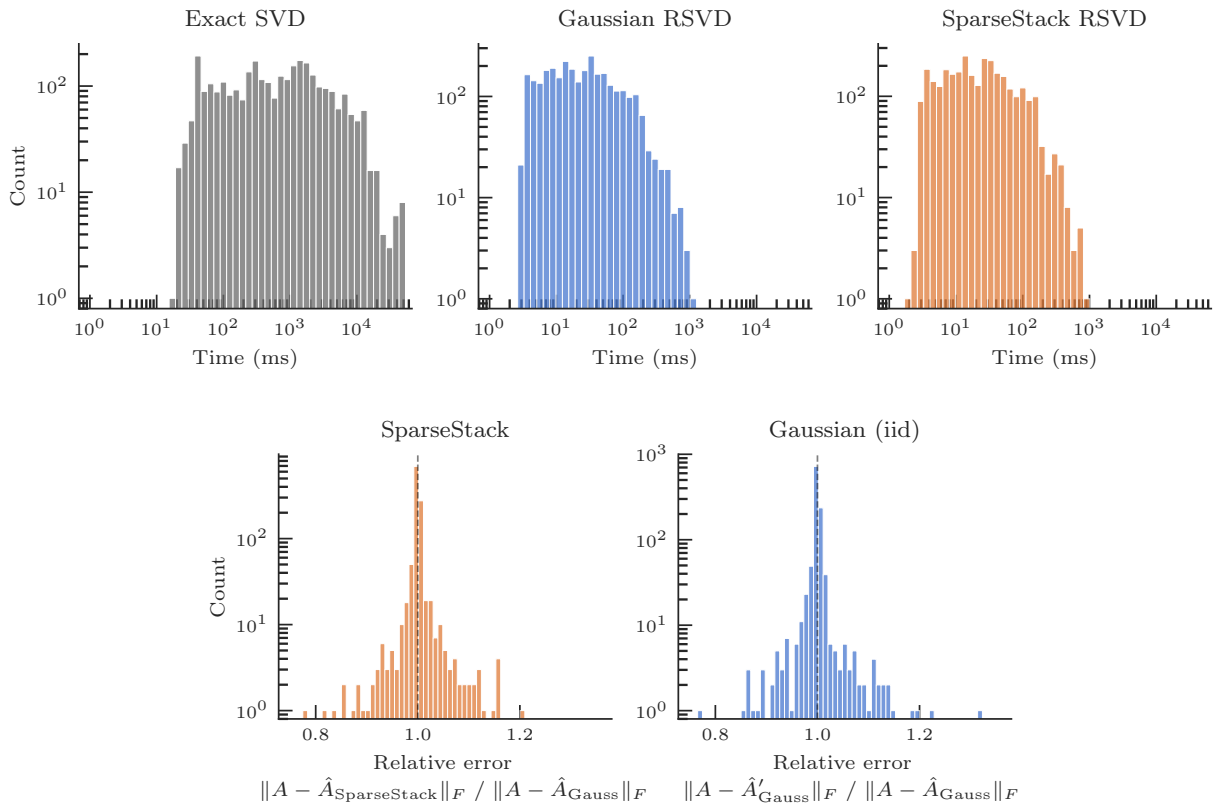

Figure 1: Top: Runtime histograms for exact SVD and RSVD with Gaussian/SparseStack sketching matrices. Bottom: Error of RSVD with SparseStack and Gaussian sketching matrices relative to RSVD with an independently sampled Gaussian sketching matrix

2.3 Nyström approximation

Let $A \in \mathbb{F}^{n\times n}$ be positive semi-definite and $\Omega \in \mathbb{F}^{n\times k}$ a sketching matrix. We wish to find a rank-r approximation of A. To that end, define the sketch $Y = A\Omega$. We approximate A with

$$\hat{A} = Y(\Omega^* Y)^\dagger Y^* \tag{2}$$

We conclude that $\hat{A}$ is PSD by noting that, in general, for B PSD and any matrix C, C^*BC and CBC^* are also PSD. Furthermore, the adjoint and the Moore-Penrose pseudoinverse commute and we have that

$$\hat{A} = Y(\Omega^* Y)^\dagger Y^* = Y(\Omega^* A\Omega)^\dagger Y^*$$

Imported Lemma 2.2. *(Gram correspondance [1, Imported Theorem 4.2][11, Lemma 1]) Let $A \in \mathbb{R}^{n\times n}$, $\Omega \in \mathbb{F}^{n\times k}$ an arbitrary test matrix, and $\hat{A}$ as defined in equation 2. Then, for $Q = \text{orth}(A^{\frac{1}{2}}\Omega)$, we get that*

$$\|A - \hat{A}\|_* \leq \|A^{\frac{1}{2}} - QQ^* A^{\frac{1}{2}}\|_F^2 \tag{3}$$

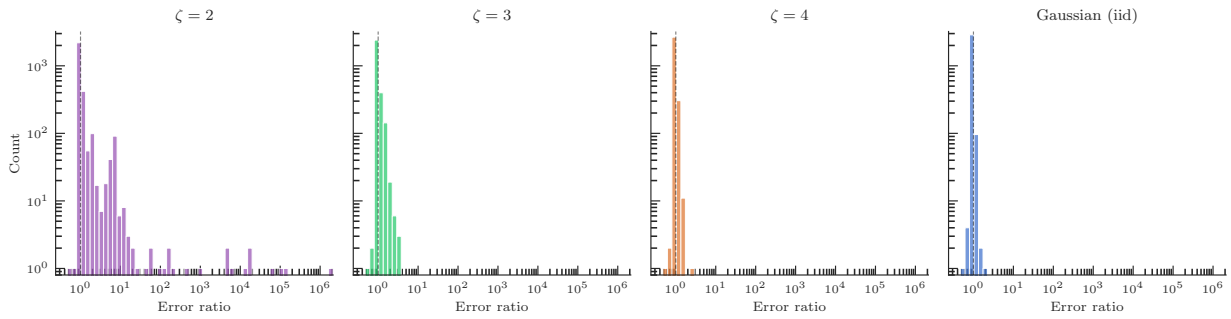

Figure 2: Distribution of the RSVD error ratio $\|A - \hat{A}_{\text{SparseStack}}\|_F / \|A - \hat{A}_{\text{Gauss}}\|_F$ for SparseStack matrices with different sparsity levels ζ, compared to the Gaussian sketching matrix baseline.

Notice that the right hand side of line 3 is exactly the Frobenius norm error obtained from performing RSVD on $A^{\frac{1}{2}} \in \mathbb{F}^{n \times n}$ with the sketching matrix $\Omega \in \mathbb{F}^{n \times k}$. Therefore, we can just directly apply the bound obtained in Theorem2.1 to obtain that

Theorem 2.2. *Let $A \in \mathbb{R}^{n \times n}$ be PSD and $\Omega \in \mathbb{F}^{n \times k}$ (r, α)-OSI. Then, for the Nyström approximation $\hat{A}$ as defined in equation 2, we get that with probability at least $\frac{9}{10}$,*

$$\|A - \hat{A}\|_* \leq \frac{C}{\alpha} \|A^{\frac{1}{2}} - [A^{\frac{1}{2}}]_r\|_F^2 = \frac{C}{\alpha} \|A - [A]_r\|_*$$

In Section 3.2, Nyström approximation shall prove useful for efficiently finding approximation of inverses of large covariance matrices.

2.4 Sketch and solve

We wish to get approximate solutions to the least squares problem

$$X \in \text{argmin}_{X \in \mathbb{F}^{d \times m}} \|AX - B\|_F^2 \tag{4}$$

where $A \in \mathbb{F}^{n \times d}$, $B \in \mathbb{F}^{n \times m}$

To that end, we take a random test matrix $\Omega \in \mathbb{F}^{n \times p}$ and multiply both sides in (4) with it. We then solve the sketched problem

$$\tilde{X} \in \text{argmin}_{\tilde{X} \in \mathbb{F}^{d \times m}} \left\| \Omega^* A \tilde{X} - \Omega^* B \right\|_F^2$$

We hope that the random mapping of the problem into a lower-dimensional space maintains enough characteristics of the original problems that a solution in the low-dimensional space approximates the true solution well with high probability.

Indeed, we shall see that the Frobenius norm error of the sketched solution differs from that of the optimum by at most a constant factor with high probability. To prove this, we shall make a change of variables and introduce two lemmata:

Let $Q\Sigma V^*$ be the reduced SVD of A, meaning that $\Sigma \in \mathbb{F}^{\text{Rank}(A) \times \text{Rank}(A)}$ is a square invertible matrix. We make a change of variables from

$$\operatorname{argmin}_X \|AX - B\|_F^2$$

by setting $\Sigma V^* X \to Z$ to obtain the minimum Frobenius norm solution

$$Z^\star = Q^* B \in \operatorname{argmin}_Z \|QZ - B\|_F^2$$

And, similarly, for the sketched problem:

$$\tilde{Z}^\star = (\Omega^* Q)^\dagger \Omega^* B \in \operatorname{argmin}_{\tilde{Z}} \left\|\Omega^* Q \tilde{Z} - \Omega^* B\right\|_F^2$$

Lemma 2.2. *Let $Q_\perp$ be a matrix such that its column vectors together with the column vectors of Q form an orthonormal basis of $\mathbb{F}^n$. Define $\Psi_1 = \Omega^* Q$ and $\Psi_2 = \Omega^* Q_\perp$*

Then $I_n = QQ^* + Q_\perp Q_\perp^*$ and, if $\Psi_1^\dagger \Psi_1 = I_r$ (i.e. $\Psi_1 = \Omega^* Q$ is injective), we get that

$$\tilde{Z}^\star - Z^\star = \Psi_1^\dagger \Psi_2 Q_\perp^* B$$

Indeed, let $e_1, \cdots e_n$ be the orthonormal basis formed by the column vectors of Q and $Q_\perp$. Then for any vector $v \in \mathbb{F}^n$,

$$v = \sum_{k=1}^{n} e_k \langle v, e_k \rangle = (QQ^* + Q_\perp Q_\perp^*) v$$

Thus, indeed, $I_n = QQ^* + Q_\perp Q_\perp^*$. Therefore,

$$\begin{aligned}
\tilde{Z} &= (\Omega^* Q)^\dagger \Omega^* B = \Psi_1^\dagger \Omega^* B \\
&= \Psi_1^\dagger \Omega^* (QQ^* + Q_\perp Q_\perp^*) B = \Psi_1^\dagger \Psi_1 Q^* B + \Psi_1^\dagger \Psi_2 Q_\perp^* B \\
&= Z^\star + \Psi_1^\dagger \Psi_2 Q_\perp^* B
\end{aligned}$$

Theorem 2.3. *Let Ω be (d, α)-OSI and $\tilde{X}^\star = (\Omega^* A)^\dagger (\Omega^* B) \in \operatorname{argmin}_{\tilde{X} \in \mathbb{F}^{d \times m}} \left\|\Omega^* A \tilde{X} - \Omega^* B\right\|_F^2$. Then, with probability at least 0.9,*

$$\left\|A\tilde{X}^\star - B\right\|_F^2 \leq \frac{\tilde{C}}{\alpha} \|AX^\star - B\|_F^2$$

We shall again use the same change of variable as before and instead prove that with probability at least 0.9, we have

$$\left\|Q\tilde{Z}^\star - B\right\|_F^2 \leq \frac{C}{\alpha} \|QZ^\star - B\|_F^2$$

First note that $Q^*(QZ^\star - B) = Z^\star - Q^* B = Z^\star - Z^\star = 0$, meaning that the column vectors of Q and $QZ^\star - B$ are pairwise orthogonal. Furthermore, the Frobenius norm is

invariant under Q because $\|QM\|_F^2 = \mathrm{Tr}(M^*Q^*QM) = \mathrm{Tr}(M^*M) = \|M\|_F^2$.Thus, with probability at least 0.9,

$$\begin{aligned}
\left\|Q\tilde{Z}^\star - B\right\|_F^2 &= \|QZ^\star - B\|_F^2 + \left\|QZ^\star - Q\tilde{Z}^\star\right\|_F^2 \\
&= \|QZ^\star - B\|_F^2 + \left\|Z^\star - \tilde{Z}^\star\right\|_F^2 \\
&= \|QZ^\star - B\|_F^2 + \left\|\Psi_1^\dagger \Psi_2 Q_\perp^* B\right\|_F^2 && \text{(by Lemma 2.2)} \\
&\leq \|QZ^\star - B\|_F^2 + \frac{C}{\alpha}\|Q_\perp^* B\|_F^2 && \text{(by an argument analogous to Lemma 2.1)} \\
&= \|QZ^\star - B\|_F^2 + \frac{C}{\alpha}\|Q_\perp Q_\perp^* B\|_F^2 \\
&= \|QZ^\star - B\|_F^2 + \frac{C}{\alpha}\|(I_n - QQ^*)B\|_F^2 \\
&= \|QZ^\star - B\|_F^2 + \frac{C}{\alpha}\|B - QZ^\star\|_F^2 \\
&\leq \frac{\tilde{C}}{\alpha}\|QZ^\star - B\|_F^2
\end{aligned}$$

Algorithm 2 Sketch and solve

Input: Matrices $A \in \mathbb{F}^{n\times d}$ and $B \in \mathbb{F}^{n\times m}$ as well as a (d,α)-OSI matrix $\Omega \in \mathbb{F}^{n\times k}$

Output: An approximate solution $\tilde{X}^\star$ to the least squares problem

$$\mathrm{argmin}_{X\in\mathbb{F}^{d\times m}} \|AX - B\|_F^2$$

1: Compute sketch $\hat{A} = \Omega^* A$

2: Compute sketch $\hat{B} = \Omega^* B$

3: Find a solution $\tilde{X}^\star$ to the least squares problem

$$\mathrm{argmin}_{\tilde{X}} \left\|\hat{A}\tilde{X} - \hat{B}\right\|_F^2$$

This can be achieved by either computing the reduced SVD of $\hat{A} = U\Sigma V^*$ and returning $V\Sigma^{-1}U^*\hat{B}$ or via QR decomposition

4: **return** $\tilde{X}$

2.4.1 Numerical experiments

We analyze performance in terms of accuracy and runtime for sketch-and-solve on three synthetic benchmarks with $A \in \mathbb{R}^{100000\times 300}$, $B \in \mathbb{R}^{100000\times 3}$, and embedding dimension $k = 600 = 2d$. We compare Gaussian sketching against SparseStack with row sparsity $\zeta \in \{2, 4\}$. Details on the three benchmarks can be found in appendix B. The observation that SparseStack matrices with a sparsity parameter of even just $\zeta = 2$ seem to yield low error was surprising to us.

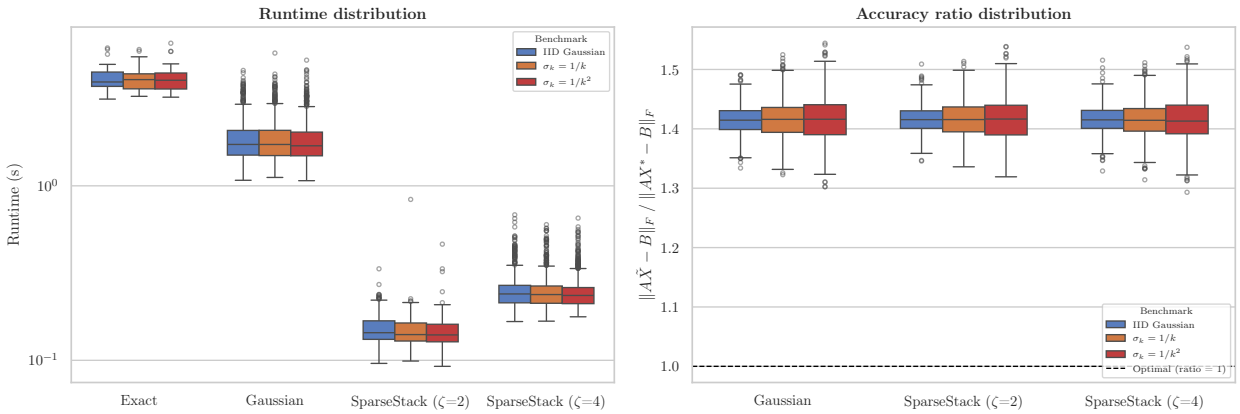

Figure 3: Runtime and accuracy distribution for sketch-and-solve with Gaussian and SparseStack ($\zeta \in \{2, 4\}$) sketching matrices across three matrix structures: iid Gaussian, $\sigma_k = 1/k$ decay, and $\sigma_k = 1/k^2$ decay. 1000 trials per sketch method, n=100,000, d=300, k=600.

2.5 Asymptotic runtime analysis

To obtain the results from table 1, we cite the standard runtime $\mathcal{O}(nd^2)$ for a classical SVD implementation, as e.g. noted by Halko, Martinsson, and Tropp [4] in Section 3.3.1. The costs for sketching (marked in blue) are also straightforward to obtain. We assume that the embedding dimension k is small relative to n, m and d.

For least squares, we can compute the reduced SVD of $A \in \mathbb{F}^{n\times d}$ and $\hat{A} = \Omega^* A \in \mathbb{F}^{k\times d}$ in $\mathcal{O}(d^2 n)$ and $\mathcal{O}(k^2 d)$ flops respectively. To then compute $V\Sigma^{-1}U^*B$ for the direct, non-sketched case takes $\mathcal{O}(nmd + d^2 m)$ operations because B, U, and V have dimensions at most $n \times m$, $n \times d$ and $d \times d$ respectively.

When sketching, we have to compute the same quantity for the sketched matrices $\hat{B} = \Omega^* B$ and $\hat{A} = \Omega^* A = \hat{U}\hat{\Sigma}\hat{V}$. Since $\hat{B}$, $\hat{U}$, and $\hat{V}$ have dimensions at most $k \times m$, $k \times k$ and $d \times k$ respectively, we get a runtime that is $\mathcal{O}(k^2 m + dkm)$

3 Gaussian matrices

3.1 Redeveloping Gaussian Processes

Definition 3.1. *A Gaussian process $(X_i)_{i\in\mathcal{I}}$ is a collection of random variables such that for any finite subset $\mathcal{J} = \{j_1, \cdots, j_n\} \subseteq \mathcal{I}$, and $a_1, \cdots, a_n \in \mathbb{R}$, $a_1 X_{j_1} + \cdots + a_n X_{j_n}$ is Gaussian.*

Theorem 3.1. *The distribution of a Gaussian process is uniquely determined by its means $\mathbb{E}[X_i]_{i\in\mathcal{I}}$ and covariance functions $\mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]_{i,j\in\mathcal{I}}$*

We follow the proof from Lemma 14.1 of Kallenberg [12]. Let $(X_i)_{i\in\mathcal{I}}$ and $(Y_i)_{i\in\mathcal{I}}$ be Gaussian processes for which the means and covariance functions agree.

Recall from undergraduate probability theory that the distribution of a random vector Z in $\mathbb{R}^n$ is uniquely determined by its characteristic function $t \in \mathbb{R}^n \mapsto \mathbb{E}[e^{itZ}]$. This implies that the joint distribution of $Z_1, \cdots, Z_n$ is uniquely determined by the distributions of the projections $a_1 Z_1 + \cdots + a_n Z_n$ for $a_1, \cdots, a_n \in \mathbb{R}$.

Table 1: Asymptotic runtime for input $A \in \mathbb{F}^{n\times d}$, $B \in \mathbb{F}^{n\times m}$, sketch dimension k, and sparsity parameter ζ for SVD, matrix inversion, and least squares. Assuming $n \geq d, m$

Task	Sketch	Total Cost
SVD	-	$\mathcal{O}(nd^2)$
Randomized SVD	Gaussian	$\mathcal{O}(ndk + k^2d + nkd)$
Randomized SVD	SparseStack	$\mathcal{O}(nd\zeta + k^2d + nkd)$
Invert positive semidefinite $A_{\text{PSD}} \in \mathbb{R}^{n\times n}$	-	$\mathcal{O}(n^3)$
Invert Nyström approximation of A_{PSD}	Gaussian	$\mathcal{O}(n^2k + nk^2)$
Invert Nyström approximation of A_{PSD}	SparseStack	$\mathcal{O}(n^2\zeta + nk^2)$
Least squares (via SVD)	-	$\mathcal{O}(d^2n + nmd)$
Sketch & Solve (via SVD)	Gaussian	$\mathcal{O}(ndk + nmk + dkm + k^2m)$
Sketch & Solve (via SVD)	SparseStack	$\mathcal{O}(nd\zeta + nm\zeta + dkm + k^2m)$

So let $a_1, \ldots, a_n \in \mathbb{R}$ and $i_1, \ldots, i_n \in \mathcal{I}$. Then by linearity, $\mathbb{E}[\sum_{k=1}^n a_k X_{i_k}] = \mathbb{E}[\sum_{k=1}^n a_k Y_{i_k}]$ and

$$\begin{aligned}
\operatorname{Var}[\sum_{k=1}^n a_k X_{i_k}] &= \mathbb{E}[(\sum_{k=1}^n a_k X_{i_k} - \mathbb{E}[a_k X_{i_k}])^2] \\
&= \sum_{k,l=1}^n E[(a_k X_{i_k} - a_k \mathbb{E}[X_{i_k}])(a_l X_{i_l} - a_l \mathbb{E}[X_{i_l}])] \\
&= \sum_{k,l=1}^n a_k a_l \operatorname{Cov}(X_{i_k}, X_{i_l})
\end{aligned}$$

and analogously for the variance of $\sum_{k=1}^n a_k Y_{i_k}$. But thus $\sum_{k=1}^n a_k Y_{i_k}$ and $\sum_{k=1}^n a_k X_{i_k}$ are Gaussians with identical mean and variance and hence have the same distribution. We now assume that the σ-algebra of $(X_i)_{i\in\mathcal{I}}$-measurable sets is the one generated by cylinder sets, i.e.

$$\sigma\left(\{(X_{i_1} \in A_1, \cdots, X_{i_n} \in A_k) \mid n \in \mathbb{N}, i_1, \cdots, i_n \in \mathcal{I} \cdots, A_1, \cdots, A_n \in \mathcal{B}(\mathbb{R})\}\right)$$

But by our argument above, the probability measures of $(X_i)_{i\in\mathcal{I}}$ and $(Y_i)_{i\in\mathcal{I}}$ agree on cylinder sets. Since the set of subsets on which any two measures agree forms a σ-algebra, we get that $(X_i)_{i\in\mathcal{I}}$ has the same distribution as $(Y_i)_{i\in\mathcal{I}}$.

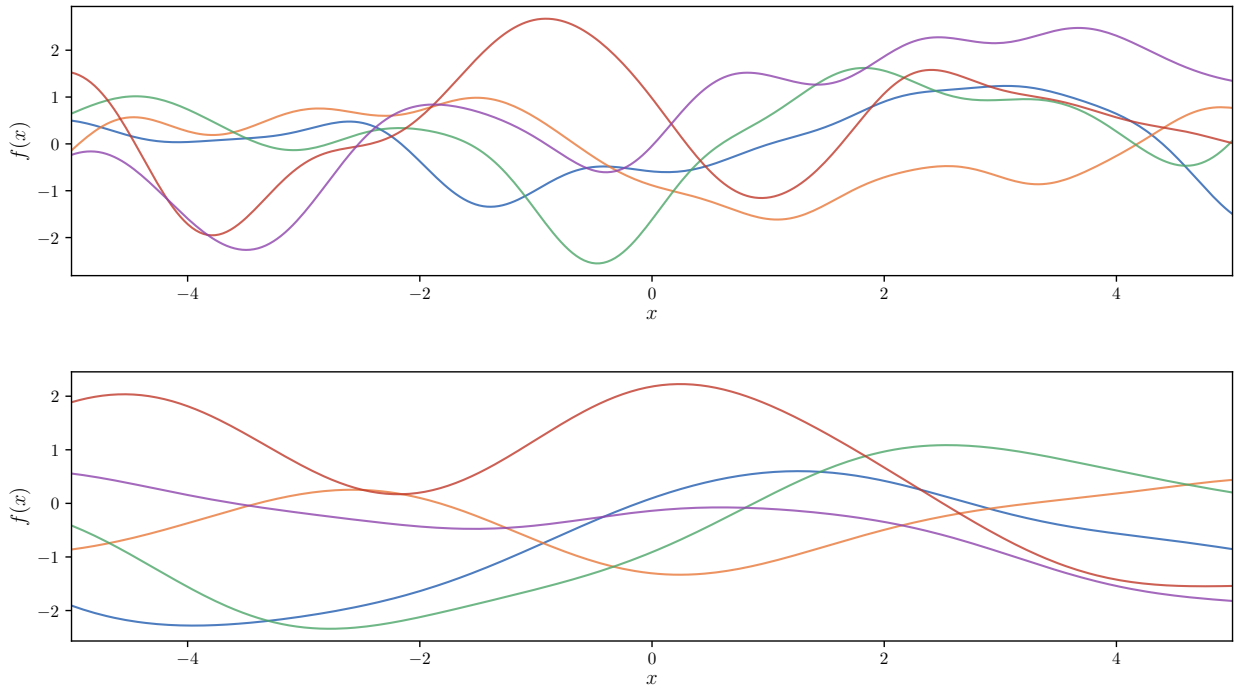

Figure 4: Samples from Gaussian process with $\text{Cov}(f(x), f(y)) = \exp(-\frac{\|x-y\|_2^2}{\sigma^2})$. The top subplot has $\sigma = 0.5$, the bottom $\sigma = 3$

3.2 Regression with Gaussian Processes

We can use Gaussian Processes (GPs) for regression. Indeed, we can let $\mathcal{I} = \mathbb{R}^n$ and get a distribution of (continuous) functions $f : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\mathbb{E}[X_x] = 0 \tag{5}$$

$$\text{Cov}(X_x, X_y) = \exp(-\frac{\|x-y\|_2^2}{2\sigma^2}) \tag{6}$$

$$f(x) = X_x \tag{7}$$

The fact that such a GP exists & that samples from it are almost-surely smooth is out of scope for this paper. What is relevant to us is that we can find relatively simple algebraic expression for its conditional distribution and with that do regression. More concretely, suppose we are given the values of some function at points $x_1, x_2, \cdots, x_n$ and we wish to predict the values of that function at $y_1, y_2, \cdots, y_m$. Ideally, we would not only predict some specific values, but an entire distribution of values based on the limited knowledge we have of the function.

In fact, our setup is perfectly suited to answer this exact question. We simply model the function f as being drawn from the distribution above, conditioned on $f(x_1), f(x_2), \cdots, f(x_n)$. All we now need is to derive the conditional distribution of a multivariate Gaussian:

Lemma 3.1. *Let* $a = (a_1 \cdots a_n) \in \mathbb{R}^n$, $b = (b_1 \cdots b_m) \in \mathbb{R}^m$, *and*

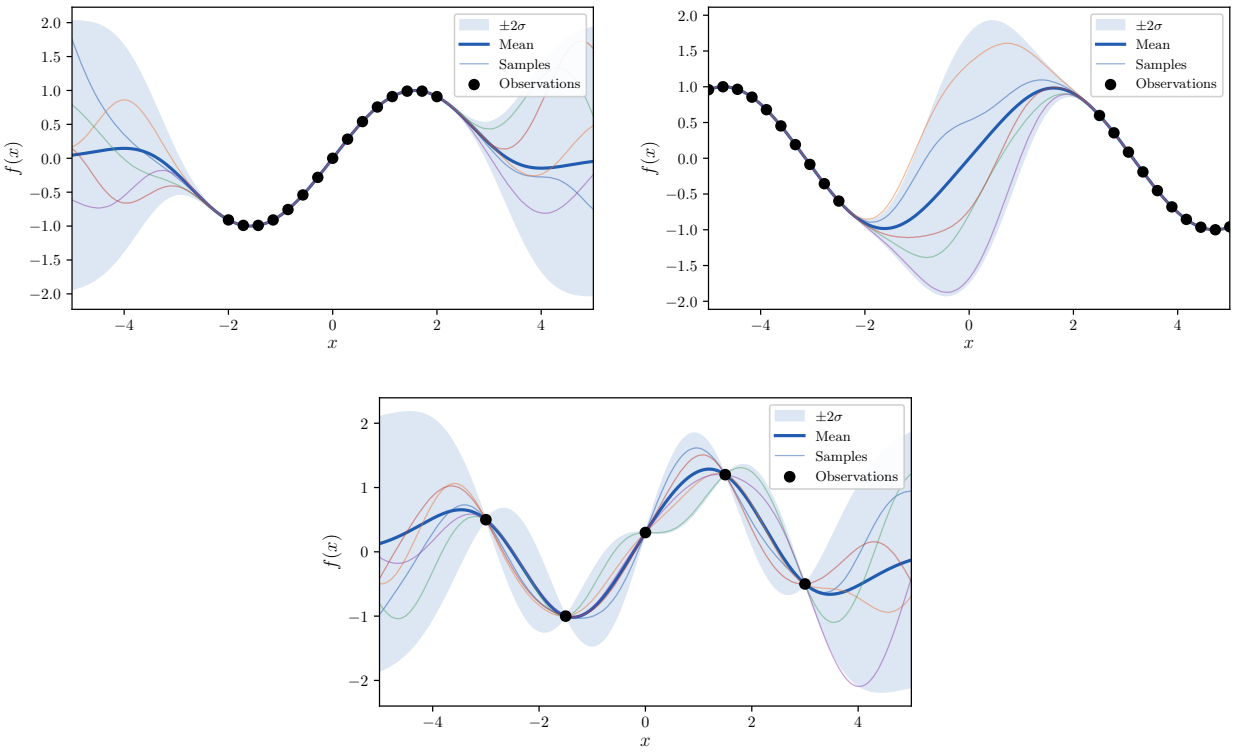

Figure 5: Conditioned Gaussian process with $\sigma = 1.0$. Observations in top subplots are drawn from $\sin(x)$

$$\begin{bmatrix} a \\ b \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix})$$

where $\Sigma_{aa}, \Sigma_{ab}, \Sigma_{ba}$, *and* Σ_{bb} *are the covariance matrices of the random vector. Then,*

$$b \,|\, a \sim \mathcal{N}(\Sigma_{ba}\Sigma_{aa}^{-1}a, \Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab}) \tag{8}$$

Define $c = b - \Sigma_{ba}\Sigma_{aa}^{-1}a$. We wish to show that c and a are independent using the following claim:

Claim 3.1. *For* X, Y *jointly Gaussian,* $\mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = 0$ *implies that* X *and* Y *are independent.*

Proof: Let $\tilde{X}$ and $\tilde{Y}$ be independently distributed like X and Y respectively. Since (X, Y) and $(\tilde{X}, \tilde{Y})$ are both jointly Gaussian with matching covariances, it follows from Theorem 3.1 that they have the same distribution.

We directly apply this to prove that a and c are independent. Indeed, note that (c, a) are jointly Gaussian and their covariance is given by:

$$\begin{aligned} \mathbb{E}[(c - \mathbb{E}[c])(a - \mathbb{E}[a])] = \mathbb{E}[ca] = \mathbb{E}[(ba - \Sigma_{ba}\Sigma_{aa}^{-1}a)a] = \mathbb{E}[ba] - \mathbb{E}[\Sigma_{ba}\Sigma_{aa}^{-1}aa] \\ = \Sigma_{ba} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{aa} = 0 \end{aligned}$$

Thus, we can independently draw $\tilde{c} \sim c$ and get that $b|a \sim \tilde{c} + \Sigma_{ba}\Sigma_{aa}^{-1}a$. Noting that, in general, $\mathrm{Cov}(Ax, By) = A\,\mathrm{Cov}(x, y)B^T$, we calculate the variance of c:

$$\begin{aligned}\mathrm{Var}[c] &= \mathrm{Var}[b - \Sigma_{ba}\Sigma_{aa}^{-1}a] = \mathrm{Var}[b] + \Sigma_{ba}\Sigma_{aa}^{-1}\,\mathrm{Cov}(a,a)\Sigma_{aa}^{-1}\Sigma_{ab} - 2\,\mathrm{Cov}[b, \Sigma_{ba}\Sigma_{aa}^{-1}a] \\ &= \Sigma_{bb} + \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{aa}\Sigma_{aa}^{-1}\Sigma_{ab} - 2\Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab} \\ &= \Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab}\end{aligned}$$

Thus, $b \,|\, a \sim \mathcal{N}(\Sigma_{ba}\Sigma_{aa}^{-1}a, \Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab})$, as desired.

We can also model measurement noise: assume that we are modeling a function f coming from a Gaussian process with kernel $\mathrm{Cov}(f(x), f(y)) = \exp(-\frac{\|x-y\|_2^2}{2\sigma^2})$ as defined in equations 5 and 6. Assume that we access samples from it through a noisy channel, i.e.

$$a_i = f(x_i) + \epsilon_i \qquad \epsilon_i \sim \mathcal{N}(0, \sigma_{\text{noise}})$$

while

$$b_j = f(y_j)$$

Then, we get covariance matrices $\Sigma_{aa}, \Sigma_{ab}, \Sigma_{ba}$ and Σ_{bb} defined by

$$\begin{aligned}\mathrm{Cov}(f(x_i) + \epsilon_i, f(x_k) + \epsilon_k) &= \mathrm{Cov}(f(x_i), f(x_k)) + \delta_{ik}\sigma_{\text{noise}}^2 \\ \mathrm{Cov}(f(x_i) + \epsilon_i, f(y_j)) &= \mathrm{Cov}(f(x_i), f(y_j)) \\ \mathrm{Cov}(f(y_j), f(y_l)) &= \mathrm{Cov}(f(y_j), f(y_l))\end{aligned}$$

with the distribution of $f(y_j)$ conditioned on our noisy observation of $f(x_i)$ being given by equation 8.

3.3 Numerical experiments

For large n, the computational cost of $\mathcal{O}(n^3)$ to compute Σ_{aa}^{-1} becomes prohibitive. Because the covariance matrix is positive definite, a natural approach is to attempt to approximate it using the Nyström approximation developed in Section 2.3.

Indeed, assume we are given points on some, in our case synthetically generated terrain. We observe heights with some fixed noise level $\sigma_{\text{noise}} = 0.08$ and wish to approximate the full terrain function using the Gaussian process as described above. We can then compute the mean $\Sigma_{yx}\Sigma_{aa}^{-1}x$ and covariance matrix $\Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab}$ of the conditional distribution by approximating Σ_{xx} with its Nyström approximation, obtained through either Gaussian or SparseStack sketches.

As we can observe in Figures 6 and 7, SparseStack yields a noticeable speedup while having quality indistinguishable from the Gaussian sketch. Furthermore, all approximations with Gaussian processes result in significantly more accurate terrain estimation than the simpler K-nearest neighbors baseline. For implementation details, see Appendix D

3.4 Gaussian random matrices

Definition 3.2. *A Gaussian random matrix [13, page 4] is a matrix $A \in \mathbb{F}^{n\times n}$ for which the real and complex entries are jointly Gaussian.*

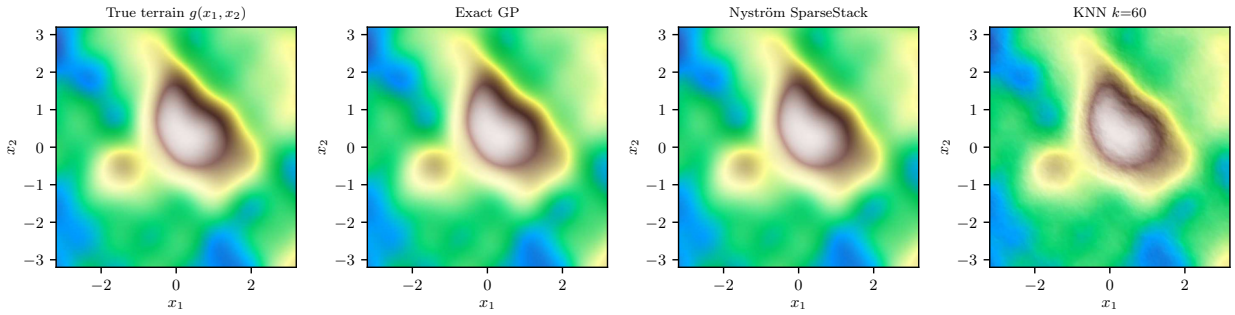

Figure 6: Estimation of terrain surface for various approximation methods, using $\zeta = 4$ for the Nyström approximation with SparseStack sketching matrix

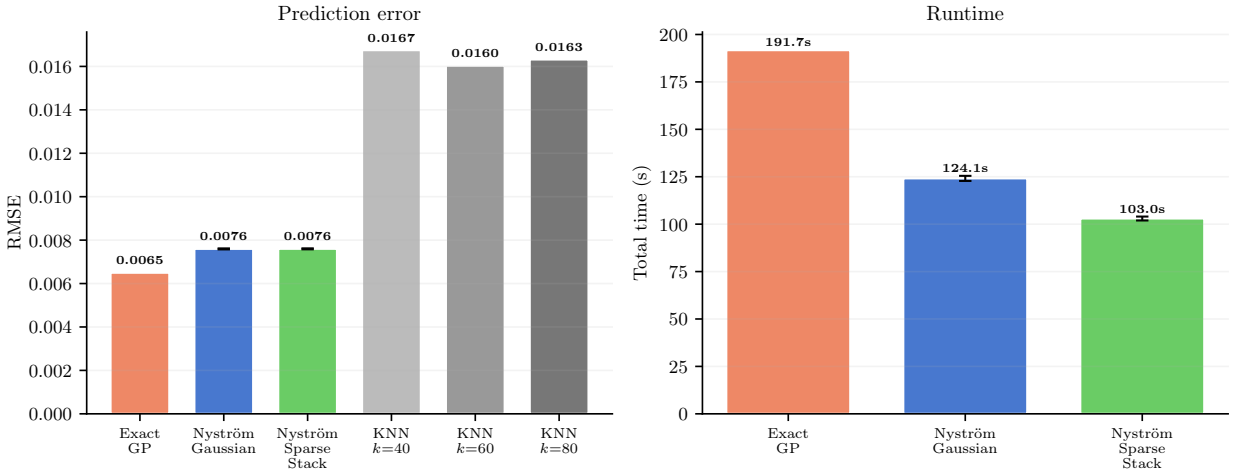

Figure 7: Root mean squared error and runtime for Gaussian Process, exact and with Nyström using various sketching methods. We also compare against a KNN baseline. Intervals shown are 95% confidence intervals.

Lemma 3.2. *For every $d \in \mathbb{N}$, let $H^{(d)} \in \mathbb{C}^{d\times d}$ be a random matrix with entries iid in $\mathcal{N}_{\mathbb{C}}(0,1)$, and $Q \in \mathbb{C}^{d\times r}$ orthogonal (i.e. $Q^*Q = I_r$), $r \leq d$. Then,*

$$Q^*H^{(d)}Q \sim H^{(r)}$$

We first observe that $M = Q^*H^{(d)}Q$ is a Gaussian process with index set

$$\mathcal{I} = \{i, j) \mid i, j \in \{1, \cdots r\}\}$$

For the expectation, we get

$$\mathbb{E}[Q^*H^{(d)}Q] = Q^*\mathbb{E}[H^{(d)}]Q = 0 = \mathbb{E}[H^{(r)}]$$

For the covariances, first note that

$$\begin{aligned}
\mathbb{E}[(Q^*H^{(d)}Q)_{a,b}\overline{(Q^*H^{(d)}Q)_{c,d}}] &= \mathbb{E}[(\sum_{i,j} Q^*_{a,i}H^{(d)}_{i,j}Q_{j,b})(\sum_{k,l}\overline{Q^*_{c,k}H^{(d)}_{k,l}Q_{l,d}})] \\
&= \sum_{i,j,k,l}\overline{Q_{i,a}}Q_{j,b}\overline{Q_{k,c}}Q_{l,d}\mathbb{E}[H_{i,j}\overline{H_{k,l}}] \\
&= \sum_{i,j,k,l}\overline{Q_{i,a}}Q_{j,b}\overline{Q_{k,c}}Q_{l,d}\delta_{(i,j),(k,l)} \\
&= \sum_{i,j}\overline{Q_{i,a}}Q_{j,b}\overline{Q_{i,c}}Q_{j,d} \\
&= \delta_{a,c}\delta_{b,d} \\
&= \mathbb{E}[H^{(r)}_{a,b}\overline{H^{(r)}_{c,d}}]
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}[(Q^*H^{(d)}Q)_{a,b}(Q^*H^{(d)}Q)_{c,d}] &= \mathbb{E}[(\sum_{i,j} Q^*_{a,i}H^{(d)}_{i,j}Q_{j,b})(\sum_{k,l}Q^*_{c,k}H^{(d)}_{k,l}Q_{l,d})] \\
&= \sum_{i,j,k,l} Q_{i,a}\overline{Q_{j,b}}Q_{k,c}\overline{Q_{l,d}}\underbrace{\mathbb{E}[H_{i,j}H_{k,l}]}_{=\delta_{(i,j),(k,l)}(\frac{1}{2}-\frac{1}{2})=0} \\
&= 0 \\
&= \mathbb{E}[H^{(r)}_{a,b}H^{(r)}_{c,d}]
\end{aligned}$$

Furthermore, note that for W, X, Y, Z real-valued random variables,

$$\begin{aligned}
A &= \mathbb{E}[(X+iY)(W+iZ)] = \mathbb{E}[XW-YZ] + i\mathbb{E}[YW+XZ] \\
B &= \mathbb{E}[(X+iY)(W-iZ)] = \mathbb{E}[XW+YZ] + i\mathbb{E}[YW-XZ]
\end{aligned}$$

Therefore, we get that

$$\begin{aligned}
\frac{A+B}{2} &= \mathbb{E}[XW] + i\mathbb{E}[YW] \\
\frac{A-B}{2} &= -\mathbb{E}[YZ] + i\mathbb{E}[XZ]
\end{aligned}$$

Hence, for complex-valued random variables X and Y, the expectations $\mathbb{E}[\mathrm{Re}(X)\,\mathrm{Re}(Y)]$, $\mathbb{E}[\mathrm{Re}(X)\mathrm{Im}(Y)]$, $\mathbb{E}[\mathrm{Im}(X)\,\mathrm{Re}(Y)]$, and $\mathbb{E}[\mathrm{Im}(X)\,\mathrm{Im}(Y)]$ are uniquely defined by $\mathbb{E}[XY]$ and $\mathbb{E}[X\bar{Y}]$.

We conclude that the covariances of the real and complex elements of $Q^*H^{(d)}Q$ and $H^{(r)}$ match. The desired lemma therefore follows directly from Theorem 3.1.

Corollary 3.1. *For $L^{(d)} \in \mathbb{C}^{d\times d}$ with complex entries iid in $\mathcal{N}_{\mathbb{C}}(0,1)$, we define the Gaussian unitary ensemble (GUE) as the random matrix $G^{(d)}_{GUE}$ with distribution*

$$G^{(d)}_{GUE} \sim \frac{L^{(d)} + (L^{(d)})^*}{\sqrt{2}}$$

Then for any matrix $Q \in \mathbb{R}^{d\times r}$ or $Q \in \mathbb{C}^{d\times r}$ that is orthogonal (i.e. $Q^*Q = I_r$), we have that

$$Q^* G_{GUE}^{(d)} Q \sim G_{GUE}^{(r)}$$

By 3.2, $Q^*L^{(d)}Q \sim L^{(r)}$. Therefore,

$$Q^* G_{\mathrm{GUE}}^{(d)} Q = \frac{Q^*L^{(d)}Q + (Q^*L^{(d)}Q)^*}{\sqrt{2}} \sim \frac{L^{(r)} + (L^{(r)})^*}{\sqrt{2}} \sim G_{\mathrm{GUE}}^{(r)}$$

3.4.1 Concrete useful calculations

Definition 3.3. *For a random matrix $\Phi \in \mathbb{C}^{n\times m}$ and a matrix $M \in \mathbb{C}^{m\times n}$, we define*

$$\mathrm{Mom}[\Phi](M) = \mathbb{E}[\mathrm{Tr}(M\Phi)^2]$$
$$\mathrm{Var}[\Phi](M) = \mathrm{Var}[\mathrm{Tr}(M\Phi)]$$

We now go on to calculate $\mathrm{Var}[X][M]$ for specific Gaussian matrices.

For $X = gI_d$ with g being a unit gaussian, we get

$$\mathrm{Var}[X][M] = \mathrm{Var}[\mathrm{Tr}(Mg)] = \mathrm{Var}(g)\,\mathrm{Tr}(M)^2 = \mathrm{Tr}(M)^2$$

Let $L \in \mathbb{C}^{d\times d}$ have entries iid in $\mathcal{N}_{\mathbb{C}}(0,1)$, $X = G_{\mathrm{GUE}}^{(d)} \in \mathbb{C}^{d\times d}$ the *Gaussian Unitary Ensemble* as defined in Corollary 3.1, and $a, b \sim \mathcal{N}_{\mathbb{R}}(0,1)$. We get that

$$\begin{aligned}
\mathrm{Var}[X][M] &= \mathrm{Var}[\mathrm{Tr}(M\frac{L+L^*}{\sqrt{2}})] = \frac{1}{2}\mathrm{Var}[\mathrm{Tr}(M(L+L^*)] \\
&= \frac{1}{2}\mathbb{E}[(\sum_{i,j} M_{ij}(L_{ji} + \overline{L}_{ij}))\overline{(\sum_{k,l} M_{kl}(L_{lk} + \overline{L}_{kl}))}] \\
&= \frac{1}{2}\sum_{i,j,k,l} M_{ij}\overline{M}_{kl}\mathbb{E}[(L_{ji} + \overline{L}_{ij})(L_{kl} + \overline{L}_{lk})] \\
&= \frac{1}{2}\sum_{i,j,k,l} M_{ij}\overline{M}_{kl}\mathbb{E}[L_{ji}\overline{L}_{lk} + \overline{L}_{ij}L_{kl}] + \mathbb{E}[L_{ji}L_{kl} + \overline{L}_{ij}\overline{L}_{lk}] \\
&= \sum_{i,j,k,l} M_{ij}\overline{M}_{kl}(\mathbb{E}[L_{ji}\overline{L}_{lk}] + \mathbb{E}[L_{ji}L_{kl}]) \qquad \text{(by symmetry)} \\
&= \sum_{i,j,k,l} M_{ij}\overline{M}_{kl}(\delta_{ji,kl} + \delta_{ji,lk}\mathbb{E}[(a + \sqrt{-1}b)^2]) \\
&= \sum_{i,j,k,l} M_{ij}\overline{M}_{kl}(\delta_{ji,kl} + \delta_{ji,lk}\mathbb{E}[a^2 - b^2]) \\
&= \sum_{i,j,k,l} M_{ij}\overline{M}_{kl}\delta_{ji,kl} \\
&= \sum_{i,j} M_{ij}\overline{M}_{ji} \\
&= \mathrm{Tr}(M^*M) = \|M\|_F^2
\end{aligned}$$

Furthermore, $\sigma_*^2(gI_d) = 1$ and $\sigma_*^2(G_{\mathrm{GUE}}^{(d)}) = 1$ - the latter claim is stated as a fact from Camaño et al. [1]

Lemma 3.3. *Let $X, W \in \mathbb{F}^{d\times d}$ be random self-adjoint matrices and $Q \in \mathbb{F}^{d\times r}$ Then we have that*

1. $\mathbb{E}[X] = \mathbb{E}[W] \implies \mathbb{E}[Q^*XQ] = \mathbb{E}[Q^*WQ]$
2. $\mathrm{Var}[X] \geq \mathrm{Mom}[W] \implies \mathrm{Var}[Q^*XQ] \geq \mathrm{Mom}[Q^*WQ]$

We leave this proof be and instead just cite existing work - see [1, Section 7.3.1] and [13]

4 The SparseStack matrix is OSI

We are now almost ready to prove that the SparseStack matrix is OSI, very closely following the proof for imported Theorem 1.8 in Appendix B of Camaño et al. [1].

Theorem 4.1. *The SparseStack matrix $\Omega \in \mathbb{R}^{d\times k}$ is an $(r, \frac{1}{2})$-OSI for embedding dimension $k = 128r$ and sparsity level $\zeta = 96\log(40r)$*

One very interesting note: In all our empirical studies (see e.g. Figures 1, 2, 3, and 6), $\zeta = 4$ actually turns out to be sufficient to match the performance of the gold-standard Gaussian sketching matrices. This is a very curious effect that the theory presented here can not fully explain. The more general statement that SparseStack matrices are $(r, \frac{1}{2})$-OSI for some embedding dimension $\mathcal{O}(r)$ with ζ being $\mathcal{O}(1)$ was actually explicitly conjectured by Camaño et al. [1].

But now for the actual proof - we begin with isotropy:

Lemma 4.1. *The SparseStack matrix $\Omega \in \mathbb{R}^{d\times k}$ is isotropic, meaning that for every $x \in \mathbb{F}^d$,*

$$\mathbb{E}\,\|\Omega^* x\|^2 = \|x\|^2$$

Recall that we have ζ groups of column vectors in Ω from which we construct the matrix as follows:

$$\Omega = \frac{1}{\sqrt{\zeta}} \begin{bmatrix} \rho_{11} e^*_{s_{11}} & \rho_{12} e^*_{s_{12}} & \cdots & \rho_{1\zeta} e^*_{s_{1\zeta}} \\ \rho_{21} e^*_{s_{21}} & \rho_{22} e^*_{s_{22}} & \cdots & \rho_{2\zeta} e^*_{s_{2\zeta}} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{d1} e^*_{s_{d1}} & \rho_{d2} e^*_{s_{d2}} & \cdots & \rho_{d\zeta} e^*_{s_{d\zeta}} \end{bmatrix}$$

So now let $\Omega_{(l,m)}$ be the m-th column vector of the l-th group and $b = \frac{k}{\zeta}$. Then

$$\begin{aligned} \mathbb{E}[((\Omega^* x)_{(l-1)b+m})^2] = \mathbb{E}[(\Omega^*_{(l,m)} x)^2] &= \frac{1}{\zeta}\mathbb{E}[|\sum_{k=1}^{d} \rho_{kl}(e^*_{s_{kl}})_m x_k|^2] \\ &= \frac{1}{\zeta}\sum_{k=1}^{d} \mathbb{E}[|\rho_{kl}(e^*_{s_{kl}})_m x_k|^2] = \frac{1}{\zeta}\sum_{k=1}^{d} |x_k|^2 \mathbb{E}[|(e^*_{s_{kl}})_m|^2] \\ &= \sum_{k=1}^{d} \frac{|x_k|^2}{b\zeta} \end{aligned}$$

Since Ω has k identically distributed column vectors, we therefore get that

$$\mathbb{E}[\|\Omega^* x\|^2] = k\mathbb{E}[(\Omega^*_{(l,m)}x)^2] = \frac{k}{b\zeta}\sum_{k=1}^{d} x_k^2 = \|x\|_2^2$$

Before we can proceed with the proof of injectivity, we shall need to introduce the following two lemmas:

Lemma 4.2. *For a CountSketch matrix $\Phi \in \mathbb{C}^{d\times b}$ as defined in Definition 1.2, we have $\mathbb{E}[\Phi\Phi^*] = I_d$. Furthermore, for any self-adjoint matrix $M \in \mathbb{C}^{d\times d}$, the following inequality holds:*

$$\mathrm{Mom}[\Phi\Phi^*](M) \leq \mathrm{Tr}(M)^2 + \frac{2}{b}\|M\|_F^2$$

Proof, following Lemma B.1 of Camaño et al. [1]:
For the first part, simply note that

$$\mathbb{E}[\Phi\Phi^*]_{i,j} = \mathbb{E}[\rho_i\rho_j e^*_{s_i}e_{s_j}] = \mathbb{E}[\rho_i\rho_j]\,\mathbb{E}[e^*_{s_i}e_{s_j}] = \delta_{i,j}\,\mathbb{E}[e^*_{s_i}e_{s_j}] = \delta_{i,j}$$

For the second part, note that

$$\begin{aligned}
\mathrm{Mom}[\Phi\Phi^*](M) &= \mathbb{E}[\mathrm{Tr}(M\Phi\Phi^*)^2] = \mathbb{E}[(\sum_{i,j}\sum_{\alpha} M_{i,j}\Phi_{j,\alpha}\Phi^*_{\alpha,i})^2] \\
&= \mathbb{E}[\sum_{i,j,k,l} M_{i,j}M_{k,l}\sum_{\alpha,\beta}\Phi_{i,\alpha}\Phi_{j,\alpha}\Phi_{k,\beta}\Phi_{l,\beta}] \\
&= \sum_{i,j,k,l} M_{i,j}M_{k,l}\theta_{i,j,k,l}
\end{aligned}$$

where we defined

$$\theta_{i,j,k,l} = \sum_{\alpha,\beta}^{b}\mathbb{E}[\Phi_{i,\alpha}\Phi_{j,\alpha}\Phi_{k,\beta}\Phi_{l,\beta}]$$

Now, massaging the expression for $\theta_{i,j,k,l}$, we get

$$\begin{aligned}
\theta_{i,j,k,l} &= \sum_{\alpha,\beta}^{b}\mathbb{E}[\rho_i\rho_j\rho_k\rho_l\mathbb{1}_{s_i=\alpha}\mathbb{1}_{s_j=\alpha}\mathbb{1}_{s_k=\beta}\mathbb{1}_{s_l=\beta}] \\
&= \sum_{\alpha,\beta}^{b}\mathbb{E}[\rho_i\rho_j\rho_k\rho_l]\,\mathbb{E}[\mathbb{1}_{s_i=\alpha}\mathbb{1}_{s_j=\alpha}\mathbb{1}_{s_k=\beta}\mathbb{1}_{s_l=\beta}]
\end{aligned}$$

But $\mathbb{E}[\rho_i\rho_j\rho_k\rho_l] = 1$ iff one of the following cases holds (and zero otherwise):

1. $i = j = k = l$
2. $i = j \neq k = l$
3. $i = k \neq j = l$

4. $i = l \neq j = k$

We calculate $\theta_{i,j,k,l}$ for each case:

1. $$\mathbb{E}[\mathbb{1}_{s_i=\alpha}\mathbb{1}_{s_j=\alpha}\mathbb{1}_{s_k=\beta}\mathbb{1}_{s_l=\beta}] = \mathbb{1}_{\alpha=\beta}\,\mathbb{E}[\mathbb{1}_{s_i=\alpha}] = \frac{\mathbb{1}_{\alpha=\beta}}{b}$$
   $\implies \theta_{i,j,k,l} = 1$

2. $$\mathbb{E}[\mathbb{1}_{s_i=\alpha}\mathbb{1}_{s_j=\alpha}\mathbb{1}_{s_k=\beta}\mathbb{1}_{s_l=\beta}] = \mathbb{E}[\mathbb{1}_{s_i=\alpha}]\mathbb{E}[\mathbb{1}_{s_k=\beta}] = \frac{1}{b^2}$$
   $\implies \theta_{i,j,k,l} = 1$

3. $$\mathbb{E}[\mathbb{1}_{s_i=\alpha}\mathbb{1}_{s_j=\alpha}\mathbb{1}_{s_k=\beta}\mathbb{1}_{s_l=\beta}] = \mathbb{1}_{\alpha=\beta}\mathbb{E}[\mathbb{1}_{s_i=\alpha}\mathbb{1}_{s_j=\alpha}] = \frac{\mathbb{1}_{\alpha=\beta}}{b^2}$$
   $\implies \theta_{i,j,k,l} = \frac{1}{b}$

4. $$\mathbb{E}[\mathbb{1}_{s_i=\alpha}\mathbb{1}_{s_j=\alpha}\mathbb{1}_{s_k=\beta}\mathbb{1}_{s_l=\beta}] = \mathbb{1}_{\alpha=\beta}\mathbb{E}[\mathbb{1}_{s_i=\alpha}\mathbb{1}_{s_j=\alpha}] = \frac{\mathbb{1}_{\alpha=\beta}}{b^2}$$
   $\implies \theta_{i,j,k,l} = \frac{1}{b}$

   Therefore, we get

$$\begin{aligned}\mathrm{Mom}[\Phi\Phi^*](M) &= \sum_{i,j,k,l} M_{i,j}M_{k,l}\theta_{i,j,k,l} \\ &= \sum_{i,k} M_{i,i}M_{k,k} + \sum_{i\neq j}\frac{M_{i,j}M_{i,j}}{b} + \sum_{i\neq j}\frac{M_{i,j}M_{j,i}}{b} \\ &\leq \mathrm{Tr}(M)^2 + \frac{2}{b}\|M\|_F^2\end{aligned}$$

   because $M_{i,j} = \overline{M_{j,i}}$ for all i, j and thus we also get

$$\sum_{i\neq j} M_{i,j}^2 = Re(\sum_{i\neq j} M_{i,j}^2) \leq \sum_{i,j} |M_{i,j}|^2$$

Imported Lemma 4.1. *Let $W \in \mathbb{F}^{d\times d}$ be a PSD random matrix with $\mathbb{E}\,\|W\|_F^2 < \infty$ Assume that there exists a Gaussian matrix $X \in \mathbb{F}^{d\times d}$ for which*

- $\mathbb{E}[W] = \mathbb{E}[X]$
- $\mathrm{Var}[X][M] \geq \mathrm{Mom}[W][M]$ *for every self-adjoint matrix $M \in \mathbb{F}^{d\times d}$*

We go on define

$$Y = \sum_{i=1}^{k} W_i \qquad \textit{for } W_1, \ldots, W_k \overset{\text{iid}}{\sim} W$$

$$Z = \sum_{i=1}^{k} X_i \qquad \textit{for } X_1, \ldots, X_k \overset{\text{iid}}{\sim} X$$

Then for every $\delta > 0$, we get that with probability at least $1 - \delta$,

$$\lambda_{min}(Y) \geq \mathbb{E}[\lambda_{min}(Z)] - \sqrt{2\sigma_*^2(Z)\log(\frac{2d}{\delta})}$$

where weak variance $\sigma_*^2(Z)$ is defined as $\sigma_*^2(Z) = \max_{\|u\|=1} \mathrm{Var}[u^* Z u]$

Proof omitted, see Tropp [13], Theorem 2.3 and Camaño et al. [1], Imported Theorem 7.5

At long last, we are now ready to prove injectivity - the only missing puzzle piece to prove that SparseStack matrices are OSI as claimed in Theorem 4.1:

Lemma 4.3. *(Injectivity) Let $\Omega \in \mathbb{R}^{d \times k}$ be a SparseStack matrix with $r > 0$ and $k \geq 128r, \zeta \geq 96\log(40r)$. Then for every r-dimensional subspace $V \subset \mathbb{F}^d$, we have that with probability at least $\frac{19}{20}$,*

$$\forall x \in V \,:\, \|\Omega^* x\|_2^2 \geq \alpha \|x\|_2^2$$

Recall that for $\Phi_1 \cdots \Phi_\zeta$ being distributed according to the CountSketch matrix Φ as defined in definition 1.2:

$$\Omega = \frac{1}{\sqrt{\zeta}} \begin{bmatrix} \Phi_1 & \Phi_2 & \cdots & \Phi_\zeta \end{bmatrix}$$

We therefore get

$$\Omega\Omega^* = \frac{1}{\zeta} \begin{bmatrix} \Phi_1 & \Phi_2 & \cdots & \Phi_\zeta \end{bmatrix} \begin{bmatrix} \Phi_1^* \\ \Phi_2^* \\ \vdots \\ \Phi_\zeta^* \end{bmatrix} = \frac{1}{\zeta} \sum_{i=1}^{\zeta} \Phi_i \Phi_i^*$$

We now construct a gaussian comparison matrix

$$X = \frac{I_d}{\zeta} + \frac{g}{\zeta} I_d + \frac{1}{\zeta}\sqrt{\frac{2}{b}} G_{\mathrm{GUE}}^{(d)}$$

Note that

$$\frac{1}{\zeta}\mathbb{E}[\Phi\Phi^*] = \frac{1}{\zeta} I_d = \mathbb{E}[X]$$

Furthermore, by Lemma 4.2, we have that for $M \in \mathbb{C}^{d\times d}$ PSD, the following inequality holds:

$$\begin{aligned}\operatorname{Var}[X][M] &= \frac{1}{\zeta^2}\operatorname{Var}[I_d][M] + \frac{1}{\zeta^2}\operatorname{Var}[gI_d][M] + \frac{2}{\zeta^2 b}\operatorname{Var}[G^{(d)}_{\mathrm{GUE}}][M] \\ &= \frac{\operatorname{Tr}(M)^2}{\zeta^2} + \frac{2}{\zeta^2 b}\|M\|_F^2 \\ &\geq \operatorname{Mom}[\frac{\Phi\Phi^*}{\zeta}][M]\end{aligned}$$

Now let $Q \in \mathbb{C}^{d\times r}$ be orthogonal. Then, by Lemma 3.3, we get that $W = \frac{1}{\zeta}Q^*\Phi\Phi^*Q$ and $\tilde{X} = Q^*XQ$ fulfill $\mathbb{E}[\tilde{X}] = \mathbb{E}[W]$ and $\operatorname{Var}[\tilde{X}][M] \geq \operatorname{Mom}[W][M]$. Hence, W and $\tilde{X}$ fulfill the criteria for Imported Lemma 4.1.

So let Φ_i and $\tilde{X}_i$ be independently distributed according to $\Phi_i \sim \Phi$ and $\tilde{X}_i \sim \tilde{X}$. Then for $Y = \frac{1}{\zeta}\sum_{i=1}^{\zeta} Q^*\Phi_i\Phi_i^*Q$, $Z = \sum_{i=1}^{\zeta}\tilde{X}_i$ and every $\delta > 0$, with probability at least $1-\delta$,

$$\lambda_{\min}(Q^*\Omega\Omega^*Q) = \lambda_{\min}(Y) \geq \mathbb{E}[\lambda_{\min}(Z)] - \sqrt{2\sigma_*^2(Z)\log(\frac{2r}{\delta})}$$

We now simply bound both terms. First, by Corollary 3.1,

$$\begin{aligned}Z &\sim \sum_{i=1}^{\zeta}\left[Q^*\frac{I_d}{\zeta}Q + \frac{g_i}{\zeta}Q^*I_dQ + \frac{1}{\zeta}\sqrt{\frac{2}{b}}Q^*G^{(d)}_{\mathrm{GUE},i}Q\right] \\ &\sim \sum_{i=1}^{\zeta}\left[\frac{I_r}{\zeta} + \frac{1}{\zeta}g_iI_r + \frac{1}{\zeta}\sqrt{\frac{2}{b}}G^{(r)}_{\mathrm{GUE},i}\right] \\ &\sim I_r + \frac{1}{\sqrt{\zeta}}gI_r + \sqrt{\frac{2}{k}}G^{(r)}_{\mathrm{GUE}}\end{aligned}$$

where we used that $k = b\zeta$. For $\mathbb{E}[\lambda_{\min}(Z)]$, we get

$$\begin{aligned}\mathbb{E}[\lambda_{\min}(Z)] &= \mathbb{E}[\lambda_{\min}(I_r + \frac{1}{\sqrt{\zeta}}gI_r + \sqrt{\frac{2}{k}}G^{(r)}_{\mathrm{GUE}})] \\ &= 1 + \mathbb{E}[\lambda_{\min}(\frac{1}{\sqrt{\zeta}}gI_r + \sqrt{\frac{2}{k}}G^{(r)}_{\mathrm{GUE}})] \\ &= 1 - \mathbb{E}[\lambda_{\max}(\frac{1}{\sqrt{\zeta}}gI_r + \sqrt{\frac{2}{k}}G^{(r)}_{\mathrm{GUE}})] \\ &\geq 1 - \frac{1}{\sqrt{\zeta}}\mathbb{E}[\lambda_{\max}(gI_r)] - \sqrt{\frac{2}{k}}\mathbb{E}[\lambda_{\max}(G^{(r)}_{\mathrm{GUE}})] \\ &= 1 - \sqrt{\frac{8r}{k}}\end{aligned}$$

where we used without proof that $\mathbb{E}[\lambda_{\max}(G^{(r)}_{\mathrm{GUE}})] = 2\sqrt{r}$

Now, using $\sigma_*^2(G_{\mathrm{GUE}}^{(r)}) = \sigma_*^2(gI_r) = 1$ and σ_*^2-subadditivity (see Section 3.6 of Tropp [13]), we get that

$$\sigma_*^2(Z) \leq \frac{1}{\zeta}\sigma_*^2(gI_r) + \frac{2}{k}\sigma_*^2(G_{\mathrm{GUE}}^{(r)}) = \frac{1}{\zeta} + \frac{2}{k} \leq \frac{3}{\zeta}$$

because $\zeta \leq k$

Putting this together yields that with probability at least $1 - \delta$,

$$\begin{aligned}\lambda_{\min}(Q^*\Omega\Omega^*Q) &\geq \mathbb{E}[\lambda_{\min}(Z)] - \sqrt{2\sigma_*^2(Z)\log(\frac{2r}{\delta})} \\ &\geq 1 - \sqrt{\frac{8r}{k}} - \sqrt{\frac{6}{\zeta}\log(\frac{2r}{\delta})}\end{aligned}$$

Setting $\delta = \frac{1}{20}$, embedding dimension $k \geq 128r$ and row sparsity $\zeta \geq 96\log(40r)$ yields that, with probability at least $\frac{19}{20}$,

$$\lambda_{\min}(Q^*\Omega\Omega^*Q) \geq \frac{1}{2}$$

In other words, for every vector x in the (arbitrary!) r-dimensional space $\mathrm{Range}(Q)$, we have that

$$\|\Omega^* x\|_2^2 \geq \frac{1}{2}\|x\|_2^2$$

This is exactly injectivity. We can therefore conclude that for an embedding dimension $k \geq 128r$ and row sparsity $\zeta \geq 96\log(40r)$, the SparseStack matrix $\Omega \in \mathbb{C}^{d\times k}$ is indeed $(r, \frac{1}{2})$-OSI.

"Quod erat demonstrandum."

— Euclid in *Elements*, Book III, Proposition 4 [14]

Appendices

A On the use of AI tools

Given that this thesis is a review paper rather than an exploration of novel results, I would expect that state-of-the-art LLMs could as of March 2025 one-shot entire sections if not the vast majority of this paper. Thus viewing this thesis as an exercise in understanding and communicating mathematics, every single line of the thesis presented above was written by a human (me). I initially used AI-enabled tab suggestions by Cursor for latex, though I disabled these within the first two or three days of writing the thesis given that they were of low-quality and did not seem to be "in the spirit" of the thesis.

That being said, SOTA models were used for:

1. Coding numerical experiments: State-of-the-art models as of March 2025 can one-shot graphs such as Figure 1 and Figure 3. Accordingly, all code was written by these models & detailed descriptions of these implementations written by the model (!) can be found in the following sections of this appendix

2. Literature review: I here either fully understood and replicated the claimed proof myself or alternatively read the relevant section of a cited paper myself or had the model cite them to me ad verbum before committing anything to writing in this thesis.

3. Fixing latex syntax errors and asking questions regarding latex formatting

4. Providing feedback on a draft after I had already done one thorough round of corrections myself & sent said draft to Prof. Bandeira

The authors would like to note that, some results are original (which is not to say that these statements can not be found somewhere within the literature). For example, the proofs of Lemma 3.2 and Corollary 3.1 are original in the following sense: These results are only stated with references to proofs in other papers in Camaño et al. [1], so I came up with the proof strategy using the fact that means and covariances characterize a Gaussian Process (Theorem 3.1), which I had previously seen in Probability Theory class at Berkeley. I first developed a proof completely on my own for Q real-valued. Then, after asking claude how complex-valued Gaussian Processes are characterized, I proved the fact that the characterization indicated by it is characterizing myself and used it to finish the proof for the general case of complex-valued Q.

B Sketch-and-solve benchmark construction

All benchmarks use n=100,000, d=300, m=3, embedding dimension k=600=$2d$, and row sparsity $\zeta \in \{2, 4\}$. We report accuracy as the ratio $\|A\tilde{X} - B\|_F / \|AX^*_{\text{LS}} - B\|_F$, where X^*_{LS} is the exact least squares solution computed via the divide-and-conquer bidiagonal SVD (`bdcSvd` in Eigen 5.0, backed by LAPACK's `dgesdd` via Apple Accelerate) and $\tilde{X}$ is the sketched solution computed by the same routine applied to the reduced system. Each

sketch method is evaluated over 1000 independent trials with distinct random seeds. The Gaussian sketch $\Psi \in \mathbb{R}^{n\times k}$ has entries iid $\mathcal{N}(0, 1/k)$; the SparseStack sketch is constructed as a vertical concatenation of ζ independent CountSketch matrices, each mapping n rows into k/ζ buckets, with entries scaled by $1/\sqrt{\zeta}$. All experiments were run using Eigen 5.0.1 with 24GB of RAM, compiled with `clang++ -O3 -march=native` and linked against Apple Accelerate (BLAS/LAPACK). The PRNG is `std::mt19937_64` seeded with `0x12345678` for matrix generation and `42 + t*997 + z` for trial t with sparsity z.

IID Gaussian. $A \in \mathbb{R}^{n\times d}$ and $B \in \mathbb{R}^{n\times m}$ with entries iid $\mathcal{N}(0, 1)$. This yields a well-conditioned design matrix with isotropic, uncorrelated features — the most favorable setting for sketching.

SVD with $\sigma_k = 1/k$. $A = U\Sigma V^\top$ where $U \in \mathbb{R}^{n\times d}$ and $V \in \mathbb{R}^{d\times d}$ have orthonormal columns obtained via Householder QR factorization (`HouseholderQR` in Eigen) of matrices with iid $\mathcal{N}(0, 1)$ entries, and $\Sigma = \mathrm{diag}(1, 1/2, \ldots, 1/d)$. B is constructed analogously with its own independent random orthogonal factors and $\Sigma' = \mathrm{diag}(1, 1/2, 1/3)$. This produces moderate spectral decay with condition number $\kappa(A) = d = 300$.

SVD with $\sigma_k = 1/k^2$. Same construction as above but with $\Sigma = \mathrm{diag}(1, 1/4, \ldots, 1/d^2)$ and $\Sigma' = \mathrm{diag}(1, 1/4, 1/9)$. This yields rapid spectral decay with condition number $\kappa(A) = d^2 =$ 90,000, testing whether severe ill-conditioning affects sketch quality differently for Gaussian vs. SparseStack.

C RSVD benchmark construction

Test matrices. We benchmark on all matrices from the SuiteSparse Matrix Collection [9] satisfying $\min(n, d) \geq 200$ and $\max(n, d) \leq 5000$, yielding 1 011 matrices (4 additional matrices are excluded due to Lanczos convergence failure). Each matrix is stored in sparse Matrix Market format and densified prior to benchmarking.

Methods compared. Three methods are timed and evaluated on every matrix over 3 independent trials:

1. **Exact top-k SVD.** We compute the k=200 largest singular values via an implicitly restarted Lanczos iteration on $A^\top A$ (Spectra 1.0.1, `SymEigsSolver`). This avoids computing the full SVD ($\mathcal{O}(n^3)$) and instead costs $\mathcal{O}(nnz \cdot k \cdot \text{iters})$ per matrix. The tail error is $\|A - A_k\|_F = \sqrt{\|A\|_F^2 - \sum_{i=1}^{k} \sigma_i^2}$.

2. **Gaussian RSVD.** The test matrix $\Omega \in \mathbb{R}^{d\times k}$ has entries iid $\mathcal{N}(0, 1)$. The sketch $Y = A\Omega$ is computed via dense matrix multiplication (dispatched to Apple Accelerate `dgemm`). Error is computed as $\|A - QQ^\top A\|_F = \sqrt{\|A\|_F^2 - \|Q^\top A\|_F^2}$, where Q is the orthogonal factor from a Householder QR factorization of Y.

3. **SparseStack RSVD.** The test matrix $\Omega \in \mathbb{R}^{d\times k}$ is a SparseStack matrix with row sparsity ζ=4, i.e., $k/\zeta = 50$ blocks. Each row of Ω has exactly ζ nonzeros, one per block, at a uniformly random column within the block, with value $\pm 1/\sqrt{\zeta}$ (Rademacher). The sketch $Y = A\Omega$ is computed via a scatter-add pass over the columns of A, costing $\mathcal{O}(nd\zeta)$ FLOPs. QR and error computation are identical to the Gaussian case.

Error ratios. For the error-ratio histograms (bottom row of Figure 1), we report $\|A - \hat{A}_{\text{method}}\|_F / \|A - \hat{A}_{\text{Gauss}}\|_F$, where $\hat{A}_{\text{Gauss}}$ is the rank-k RSVD approximation from an independent Gaussian sketch. A ratio near 1 indicates that the method matches Gaussian quality.

Timing. For the two RSVD methods, reported wall-clock time includes the full pipeline: sketch computation, QR factorization, and projection error evaluation. For exact top-k SVD, it includes the full Lanczos iteration until convergence.

Implementation. The benchmark is implemented in C++17 using Eigen 3.4 for linear algebra, Spectra 1.0.1 for iterative eigenvalue computation, and nlohmann/json for output serialization. Matrix processing is parallelized across cores via OpenMP with dynamic scheduling. Compilation uses `-O3 -march=native`; on macOS, Apple Accelerate is linked for BLAS.

Sparsity comparison (Figure 2). To study the effect of the sparsity parameter ζ, we compute RSVD error ratios for SparseStack matrices with $\zeta \in \{2, 3, 4\}$ on the same set of benchmark matrices, using sparse matrix operations (no densification). For each matrix and trial, a single Gaussian RSVD serves as the reference; the error ratio for each ζ value and an independent second Gaussian draw are recorded. Trials where the reference error falls below $10^{-10}\|A\|_F$ (indicating effective rank $\leq k$) are excluded.

D GP regression benchmark construction

Terrain function. The ground-truth function $g\colon \mathbb{R}^2 \to \mathbb{R}$ is a hand-crafted mixture designed to exhibit multi-scale structure:

$$\begin{aligned} g(x_1, x_2) = {} & 2\, e^{-\frac{(x_1-0.5)^2+(x_2-0.3)^2}{1.8}} + 1.5\, e^{-\frac{(x_1+1.6)^2+(x_2+0.6)^2}{0.9}} + e^{-\frac{(x_1+0.2)^2}{0.4}-\frac{(x_2-1.5)^2}{3}} \\ & - 0.7\, e^{-\frac{(x_1-1.8)^2+(x_2+1.5)^2}{0.5}} - 0.6\, e^{-\frac{(x_1+2)^2+(x_2-2)^2}{0.4}} \\ & + 0.5 \sin(1.3x_1{+}0.2)\cos(1.1x_2{+}0.5) + 0.3\cos(2x_1{-}0.3)\sin(1.5x_2) \\ & + 0.15 \sin(3.5x_1{+}0.7)\cos(3x_2{+}1) + 0.12x_1 + 0.06x_2 \end{aligned}$$

Data generation. We sample n=50,000 training points uniformly on $[-3.2, 3.2]^2$ using `std::mt19937` with seed 42 and a `std::uniform_real_distribution`. Observations are $y_i = g(\mathbf{x}_i) + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma_n^2)$, σ_n=0.08. The test set is a regular 200×200 grid over $[-3.2, 3.2]^2$ (for terrain visualization) or 150×150 (for RMSE evaluation).

Kernel. We use the squared exponential (RBF) kernel $k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\big(-\|\mathbf{x}-\mathbf{x}'\|^2/(2\ell^2)\big)$ with length scale ℓ=0.6 and signal variance σ_f=1.5. The full kernel matrix $K \in \mathbb{R}^{n\times n}$ requires ≈20 GB and is never stored in full; instead, all kernel–matrix and kernel–vector products are computed in row chunks of size 2000.

Exact GP. The weight vector $\boldsymbol{\alpha} = (K + \sigma_n^2 I)^{-1}\mathbf{y}$ is obtained by assembling $K + \sigma_n^2 I$ in chunks and solving via Cholesky factorization (`Eigen::LLT`). Predictions are $\hat{\mathbf{z}} = K_*\boldsymbol{\alpha}$, where K_* is the test–train kernel matrix, again computed in chunks.

Nyström approximation. We implement Algorithm 2 from Tropp et al. [1] with sketch dimension k=1,000. The sketch $Y = K\Omega$ is computed in chunks of 2000 rows. A shift $\nu = \sqrt{\varepsilon_{\text{mach}}}\|Y\|_F$ is added for numerical stability: $\tilde{Y} = Y + \nu\Omega$. The Cholesky factor C of $\Omega^\top\tilde{Y}$ is computed, and the matrix $B = \tilde{Y}C^{-\top}$ is formed via triangular solve. A truncated SVD of B yields eigenvectors U and eigenvalues $\Lambda_i = \sigma_i^2 - \nu$ (thresholded at zero); components with $\Lambda_i < 10^{-12}\Lambda_1$ are discarded. The GP solve uses the low-rank structure: $\boldsymbol{\alpha} = (\mathbf{y} - UWU^\top\mathbf{y})/\sigma_{\text{eff}}^2$, where $W = \text{diag}(\Lambda_i/(\Lambda_i + \sigma_{\text{eff}}^2))$ and $\sigma_{\text{eff}}^2 = \sigma_n^2 + \max(\text{tr}(K) - \sum\Lambda_i,\, 0)/n$ accounts for truncated eigenvalue mass.

Sketching matrices. Two sketch types are compared: (1) Gaussian $\Omega \in \mathbb{R}^{n\times k}$ with entries iid $\mathcal{N}(0, 1/k)$; (2) SparseStack $\Omega \in \mathbb{R}^{n\times k}$ with row sparsity ζ=4 and k/ζ=250 buckets per block, entries $\pm 1/\sqrt{\zeta}$.

KNN baseline. KNN regression with $k \in \{40, 60, 80\}$ is computed using `scikit-learn` (`KNeighborsRegressor`) on the same training data and test grid.

Statistical evaluation. RMSE confidence intervals are 95% two-sided intervals using the t-distribution: $\bar{x} \pm t_{0.975,\,n-1} \cdot s/\sqrt{n}$, computed over just more than 200 independent trials per sketch method. Timing is reported from sequential runs only to avoid contention artifacts.

Implementation. The benchmark is implemented in C++17 using Eigen 3.4 for linear algebra, compiled with `-O3 -march=native` and linked against Apple Accelerate (local) or OpenBLAS (remote). The remote machine is a 128-core CPU server; parallel trial batches use independent processes with disjoint seeds to avoid RNG correlation. Plotting uses Python 3 with matplotlib (PGF backend), seaborn, and scipy.

References

[1] C. Camaño et al., "Faster linear algebra algorithms with structured random matrices", 2025, URL: https://arxiv.org/abs/2508.21189.

[2] T. Sarlós. "Improved Approximation Algorithms for Large Matrices via Random Projections". In: *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2006, pp. 143–152. DOI: 10.1109/FOCS.2006.37.

[3] P.-G. Martinsson and J. A. Tropp, "Randomized numerical linear algebra: Foundations and algorithms", 2020, URL: https://doi.org/10.1017/S0962492920000021.

[4] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions", 2011, URL: https://arxiv.org/abs/0909.4061.

[5] J. Nelson and H. L. Nguyen. "Lower Bounds for Oblivious Subspace Embeddings". In: *Automata, Languages, and Programming (ICALP)*. Vol. 8572. Lecture Notes in Computer Science. Springer, 2014, pp. 883–894. DOI: 10.1007/978-3-662-43948-7_73. URL: https://doi.org/10.1007/978-3-662-43948-7_73.

[6] J. Nelson and H. L. Nguyen, "OSNAP: Faster Numerical Linear Algebra Algorithms via Sparser Subspace Embeddings", 2014, URL: https://dl.acm.org/doi/10.1145/2559902.

[7] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams", 2004, URL: https://doi.org/10.1016/S0304-3975(03)00400-6.

[8] A. Kireeva and J. A. Tropp. Randomized matrix computations: Themes and variations. 2024. arXiv: 2402.17873 [math.NA]. URL: https://arxiv.org/abs/2402.17873.

[9] T. A. Davis and Y. Hu, "The University of Florida Sparse Matrix Collection", 2011, URL: https://doi.org/10.1145/2049662.2049663.

[10] M. Kapralov, V. Potluru, and D. Woodruff. "How to Fake Multiply by a Gaussian Matrix". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. PMLR, 2016, pp. 2101–2110. URL: https://proceedings.mlr.press/v48/kapralov16.html.

[11] A. Gittens and M. W. Mahoney, "Revisiting the Nyström method for improved large-scale machine learning", 2016, URL: https://dl.acm.org/doi/abs/10.5555/2946645.3007070.

[12] O. Kallenberg. Foundations of Modern Probability. 3rd. Vol. 99. Probability Theory and Stochastic Modelling. Springer, 2021. ISBN: 978-3-030-61870-4. DOI: 10.1007/978-3-030-61871-1.

[13] J. A. Tropp, "Comparison theorems for the minimum eigenvalue of a random positive-semidefinite matrix", Jan. 2025, URL: https://arxiv.org/abs/2501.16578.

[14] Euclid. Elements. Alexandria. c. 300 BCE.