On Sketching for Gaussian Process Regression with New Statistical Guarantees

Anonymous authors
Paper under double-blind review

Abstract

The cubic computational complexity of Gaussian Process Regression (GPR) with respect to the number of data points is a major bottleneck to its scalability. While various approaches have been proposed to address this, few come with provable guarantees. Inspired by the success of ridge leverage score based sampling in scaling kernel ridge regression El Alaoui & Mahoney (2015), we propose a sketch-based approximation for GPR using ridge leverage scores. We provide theoretical guarantees on the approximation of the predictive mean, predictive variance, and negative log-marginal likelihood in this setting. To the best of our knowledge, these are the first theoretical guarantees for approximating the predictive variance and negative log-marginal likelihood of GPR using ridge leverage score sampling. We further show that a carefully constructed sketch of the kernel matrix preserves key statistical properties of the full GPR model with high probability. Our theoretical results are supported by empirical evaluations on real-world datasets, demonstrating strong trade-offs between accuracy and efficiency.

1 Introduction

Gaussian Process Regression (GPR) is a fundamental method in probabilistic machine learning, offering a principled non-parametric approach to modeling distributions over functions Rasmussen & Williams (2005). Its strength lies in its ability to provide calibrated uncertainty estimates, which are critical in applications such as Bayesian optimization Xu et al. (2024), active learning Kapoor et al. (2007); Schreiter et al. (2015); Tebbe et al. (2024), and reinforcement learning Bıyık et al. (2020).

Gaussian Process Regression. Given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, we assume the outputs are generated from a latent function $f \sim \mathcal{GP}(m(\cdot), k_\theta(\cdot,\cdot))$, corrupted by Gaussian noise, i.e.,

$$y_i = f(x_i) + \xi_i, \quad \xi_i \sim \mathcal{N}(0, \sigma_\xi^2).$$

Here, $m(\cdot)$ is the prior mean function (often taken as constant), and $k_\theta(\cdot,\cdot)$ is a positive-definite kernel function parameterized by hyperparameters θ. A commonly used choice for the kernel function is the Radial Basis Function (RBF) kernel, defined as,

$$k_\theta(x, x') = \sigma_f^2 \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right) \tag{1}$$

where σ_f^2 controls the variance and ℓ is the lengthscale hyperparameter.

The prior over latent function values $\mathbf{f} = [f(x_1), \ldots, f(x_n)]^\top$ is multivariate Gaussian,

$$\mathbf{f} \sim \mathcal{N}(m(X), K)$$

where $K \in \mathbb{R}^{n \times n}$ is the kernel matrix with entries $K_{ij} = k_\theta(x_i, x_j)$, and $m(X)$ is the vector of prior means evaluated at training inputs.

Under this model, the noisy observations $\mathbf{y} \in \mathbb{R}^n$ are distributed as

$$\mathbf{y} \sim \mathcal{N}(m(X), K + \sigma_\xi^2 I)$$

Prediction at Test Time. For a new test point x_*, the predictive distribution of y_* conditioned on the training data is also Gaussian,

$$y_* \mid x_*, X, \mathbf{y} \sim \mathcal{N}(\mu(x_*), \mathrm{Var}(x_*))$$

where the predictive mean and variance are respectively given as,

$$\mu(x_*) = k_*^\top (K + \sigma_\xi^2 I)^{-1} \mathbf{y} \tag{2}$$

$$\mathrm{Var}(x_*) = k(x_*, x_*) - k_*^\top (K + \sigma_\xi^2 I)^{-1} k_* \tag{3}$$

with $k_* = [k_\theta(x_*, x_1), \ldots, k_\theta(x_*, x_n)]^\top$

Learning via Marginal Likelihood. Given the data, the task in GPR is to learn the kernel hyperparameters θ and noise variance σ_ξ^2 . This is typically done by maximizing the log marginal likelihood of the observed outputs,

$$\log p(\mathbf{y} \mid X, \theta) = -\frac{1}{2}\mathbf{y}^\top (K + \sigma_\xi^2 I)^{-1} \mathbf{y} - \frac{1}{2}\log\det(K + \sigma_\xi^2 I) - \frac{n}{2}\log 2\pi \tag{4}$$

This objective balances data fit (first term), model complexity (second term), and normalization.

Computational Challenges. The exact computation of equation 2 to equation 4 requires $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ memory due to the inversion and determinant of the full kernel matrix Rasmussen & Williams (2005). This limits the applicability of standard GPR to small or moderate-sized datasets. In this work, we address this scalability bottleneck through a sketching-based approximation using ridge leverage scores.

To handle this problem, a rich line of work has focused on approximating the kernel matrix using techniques such as inducing points Snelson & Ghahramani (2006), Nyström methods Williams & Seeger (2001), and randomized sketching El Alaoui & Mahoney (2015); Pilanci & Wainwright (2017). Among these, sketching methods stand out for their ability to compress large kernel matrices into compact representations with statistical guarantees. However, existing analyses often fail to characterize the precise impact of sketching on uncertainty quantification and negative log marginal likelihood of the gaussian process regression. El Alaoui & Mahoney (2015) successfully applied the ridge leverage score based sampling technique to Nystrom approximation for kernel ridge regression. However it is non-trivial to extend their method to the case of GPR with provable guarantees , specifically for the predictive variance and negative log marginal likelihood approximation. Inspired by their method, we next describe the Nystrom Sketching for Kernel Approximation specifically for GPR and then describe our main contributions in this paper.

2 Nystrom Sketching for Kernel Approximation in GPR

To overcome the above computational challenges, we employ Nystrom sketching, which provides a low-rank approximation $\widehat{K}$ to K while preserving its essential spectral structure.

Nystrom approximation. Let $J \subset \{1, \ldots, n\}$ be a set of $m \ll n$ sampled indices. Define

$$C \;=\; K_{:,J} \in \mathbb{R}^{n \times m}, \qquad W \;=\; K_{J,J} \in \mathbb{R}^{m \times m} \tag{5}$$

The classical Nystrom approximation is given by

$$\widehat{K} \;=\; C\, W^\dagger C^\top, \tag{6}$$

where $W^\dagger$ denotes the Moore–Penrose pseudoinverse of W. This approximation projects the kernel matrix onto the span of the sampled columns, yielding a rank-m surrogate of K.

Generalized sketching. More generally, let $S \in \mathbb{R}^{n\times m}$ be a tall, skinny sketching matrix (e.g., sampling matrix, random projection, or structured transform). The Nystrom approximation associated with S can be written as

$$\widehat{K}_S \;=\; K\,S\,(S^\top K S)^\dagger S^\top K, \tag{7}$$

which recovers equation 6 when S corresponds to column sampling.

Woodbury expansion for efficient solves. In GPR, inference requires computing $(K+\sigma_\xi^2 I)^{-1}$, where σ_ξ^2 is the noise variance. Replacing K with $\widehat{K}$ and applying the Woodbury identity 1 with $A=\sigma_\xi^2 I$, $U=C$, $M=W^{-1}$, $V=C^\top$, we obtain

$$(\widehat{K}+\sigma_\xi^2 I)^{-1} \;=\; \sigma_\xi^{-2} I - \sigma_\xi^{-4}\,C\big(W+\sigma_\xi^{-2}C^\top C\big)^{-1}C^\top \tag{8}$$

Lemma 1 (Woodbury Nystrom Solve)**.** *Let $\widehat{K}=CW^{-1}C^\top$ be the Nyström approximation of K. Then for any $y\in\mathbb{R}^n$,*

$$(\widehat{K}+\sigma_\xi^2 I)^{-1}y = \sigma_\xi^{-2}y - \sigma_\xi^{-4}C\big(W+\sigma_\xi^{-2}C^\top C\big)^{-1}C^\top y \tag{9}$$

Proof. Apply the Woodbury matrix identity $(A+UMV)^{-1}=A^{-1}-A^{-1}U(M^{-1}+VA^{-1}U)^{-1}VA^{-1}$ with $A=\sigma_\xi^2 I$, $U=C$, $M=W^{-1}$, $V=C^\top$. □

This expression involves inverting only an $m\times m$ matrix, substantially reducing the computational burden.

Predictive mean using Nystrom. The GPR predictive mean for a test point x_* with kernel vector $k_*\in\mathbb{R}^n$ is

$$\mu_* \;=\; k_*^\top (K+\sigma_\xi^2 I)^{-1} y \tag{10}$$

Using the Nystrom surrogate and equation 8, this becomes

$$\mu_* \;\approx\; \sigma_\xi^{-2}k_*^\top y - \sigma_\xi^{-4}k_*^\top C\big(W+\sigma_\xi^{-2}C^\top C\big)^{-1}C^\top y \tag{11}$$

Analogous derivations yield a similar reduction for the predictive variance.

Computational complexity. Constructing C requires $\mathcal{O}(nm)$ kernel evaluations and storing it uses $\mathcal{O}(nm)$ memory. Forming and inverting $A=W+\sigma_\xi^{-2}C^\top C$ costs $\mathcal{O}(nm^2+m^3)$, and each test prediction costs $\mathcal{O}(m^2)$. This is a dramatic improvement over the $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ memory required for standard GPR.

Our Contributions. The main challenge in applying the sketched Nystrom method to GPR lies in designing an efficient sketching matrix and analyzing its impact on the predictive mean, predictive variance, and negative log-marginal likelihood (NLML). While sketching methods particularly those based on ridge leverage scores (RLS) have been well studied for Kernel Ridge Regression (KRR) El Alaoui & Mahoney (2015); Rasmussen & Williams (2005), their theoretical guarantees do not fully capture the unique properties of GPR. KRR analyses typically bound the statistical risk of the point predictor El Alaoui & Mahoney (2015), whereas GPR's strengths lie in uncertainty quantification via predictive variance and model selection through NLML Hensman et al. (2013). These quantities, fundamental to GPR, have no direct analogues in KRR; the GPR posterior variance is distinct from the variance of the KRR estimator, and NLML is critical for hyperparameter learning. Our work bridges this gap by extending guarantees for the predictive mean and, more importantly, providing the first explicit approximation bounds for the predictive variance and NLML under RLS sketching, establishing a complete theoretical foundation for scalable, high-fidelity GPR.

To summarize, the main contributions of this work are as follows:

1. We propose a kernel sketching framework based on ridge leverage scores for Gaussian Process Regression (GPR).

2. We provide, to the best of our knowledge, the first theoretical guarantees for ridge leverage score–based sketching specifically for the GPR problem. Specifically, we derive non-trivial bounds on the approximation error for the predictive mean, predictive variance, and negative log-likelihood.

3. We conduct extensive empirical evaluations across multiple real-world regression benchmarks, demonstrating the effectiveness of our method compared to standard baselines.

It is important to note that while our approach is comparable to some state-of-the-art methods in terms of runtime performance, we demonstrate that it achieves provable superior predictive quality and uncertainty calibration, thereby offering an accurate and efficient alternative to existing scalable GP techniques.

3 Related Work

Various methods have been applied to scale GPR for the big data regime; see Liu et al. (2020) and references therein. There are methods based on variational inference Hensman et al. (2013), and conjugate gradient–based iterative methods Artemev et al. (2021). However, handling GPR using sampling or sketching-based methods with theoretical guarantees is relatively less explored. Hayashi et al. (2020), using a novel graphon-based analysis, derive error bounds for Gaussian process subsampling via uniform random selection. However, their bounds decay slowly; for example, predictive error scales like $O(\log^{-1/4} s)$ with the number of subsamples s. Fiedler et al. (2021) provide practical bounds on GPR in general; however, these bounds are not directly comparable to ours.

Nystrom approximation Williams & Seeger (2001) reduces the computational complexity by projecting the full kernel matrix onto a subspace spanned by a set of inducing points. However, uniform or heuristic-based selection of these points often fails to capture critical data-dependent structure, especially in high-dimensional or non-uniform settings.

This has led to the adoption of more principled sampling techniques based on *ridge leverage scores* El Alaoui & Mahoney (2015), which offer spectral guarantees and have been successfully applied in kernel ridge regression Rudi & Rosasco (2015); Musco & Musco (2017) and randomized matrix approximation Drineas et al. (2012). Despite their theoretical appeal, ridge leverage based Nystrom approximations have not been widely explored in the context of Gaussian Process Regression particularly with respect to predictive quantities such as the posterior mean, variance, and marginal likelihood. Existing bounds are not directly applicable to GPR settings. In contrast, we leverage similar sampling strategies but develop new, explicit guarantees on these key predictive quantities, bridging this important gap.

4 Algorithms

In this section, we outline the algorithms used to sketch the kernel matrix. While similar techniques have been studied in kernel ridge regression El Alaoui & Mahoney (2015), their application to Gaussian Process Regression (GPR) with explicit theoretical guarantees has not been previously established.

We use a generalized notion of leverage scores specifically designed for the ridge regression setting, referred to as the σ_ξ^2*-ridge leverage scores.*

Definition 1. *Given $\sigma_\xi^2 > 0$, the σ_ξ^2-ridge leverage scores corresponding to a kernel matrix K and regularization/noise parameter σ_ξ^2 are defined as*

$$\forall i \in \{1, \ldots, n\}, \quad l_i(\sigma_\xi^2) = \sum_{j=1}^{n} \frac{\sigma_j}{\sigma_j + \sigma_\xi^2} U_{ij}^2$$

Here, $l_i(\sigma_\xi^2)$ represents the i^{th} diagonal entry of the matrix product $K(K + \sigma_\xi^2 I)^{-1}$, where σ_j denotes the j^{th} eigenvalue of the kernel matrix K, and U is the orthonormal matrix of eigenvectors from its eigendecomposition. The set $(l_i(\sigma_\xi^2))_{1 \le i \le n}$ serves a similar role to classical leverage scores in statistics, as they help identify

Algorithm 1 σ_ξ^2-Ridge Leverage Score Sampling with Rescaling for Kernel Sketching

Input: Kernel matrix $K \in \mathbb{R}^{n\times n}$, noise parameter $\sigma_\xi^2 > 0$, sketch size $m \ll n$, Nystrom regularization parameter $\gamma > 0$
Output: Sketching matrix $S \in \mathbb{R}^{n\times m}$, sketched kernel $L_\gamma \in \mathbb{R}^{n\times n}$

Compute ridge leverage scores: $\ell_i^{(\sigma_\xi^2)} \leftarrow [K(K+\sigma_\xi^2 I)^{-1}]_{ii}$ for all $i \in \{1,\dots,n\}$ ▷ See Algorithm 2 for fast computation
Normalize scores: $p_i \leftarrow \ell_i^{(\sigma_\xi^2)} / \sum_{j=1}^n \ell_j^{(\sigma_\xi^2)}$
Initialize $S \in \mathbb{R}^{n\times m}$ as a zero matrix
for $j = 1$ to m **do**
    Sample index $i_j \sim$ Categorical$(p_1,\dots,p_n)$ ▷ Sample with replacement
    Set $S_{i_j,j} \leftarrow \frac{1}{\sqrt{m\cdot p_{i_j}}}$ ▷ Apply reweighting
end for
Compute sketched kernel: $L_\gamma \leftarrow KS(S^\top KS + \gamma I)^{-1} S^\top K$
return S, L_γ

Algorithm 2 Approximate σ_ξ^2-Ridge Leverage Score Computation via Nystrom Sketching El Alaoui & Mahoney (2015)

Input: Data points $\{x_1,\dots,x_n\}$, kernel function $k(\cdot,\cdot)$, sampling distribution $\{p_i\}_{i=1}^n$, sketch size m, regularization parameter $\sigma_\xi^2 > 0$
Output: Approximate ridge leverage scores $\{\tilde{\ell}_i\}_{i=1}^n$

Sample indices $i_1,\dots,i_m \sim$ Categorical$(p_1,\dots,p_n)$ with replacement
Form matrix $C \in \mathbb{R}^{n\times m}$ such that $C_{j,\ell} = k(x_j, x_{i_\ell})$
Form $W \in \mathbb{R}^{m\times m}$ with $W_{\ell,p} = k(x_{i_\ell}, x_{i_p})$
Compute $B \in \mathbb{R}^{n\times m}$ such that $BB^\top = CW^\dagger C^\top$ ▷ Can use Cholesky or QR on W
Compute matrix $M = (B^\top B + \sigma_\xi^2 I_m)^{-1}$
for $i = 1$ to n **do**
    Set $\tilde{\ell}_i \leftarrow B_i^\top M B_i$ ▷ B_i is the i-th row of B
end for
return $\{\tilde{\ell}_i\}_{i=1}^n$

influential data points that significantly impact the model output. In traditional settings, these scores are often derived from the row norms of the left singular vectors in the matrix U.

The effective dimension, denoted by $d_{\text{eff}}(\sigma_\xi^2)$, is defined as

$$d_{\text{eff}}(\sigma_\xi^2) = \operatorname{Tr}\left(K(K+\sigma_\xi^2 I)^{-1}\right)$$

where K is the kernel matrix and $\sigma_\xi^2 > 0$ is the regularization/noise parameter.

To efficiently approximate the σ_ξ^2-ridge leverage scores without computing the full eigendecomposition of the kernel matrix, we adopt an approximation strategy inspired by Algorithm 2 El Alaoui & Mahoney (2015).

The approximation algorithm 2 accepts as input a sampling distribution over the data points, which we set to a simple yet effective diagonal proxy where each point is sampled with probability proportional to the diagonal entry of the kernel matrix, i.e., $p_i = \frac{K_{ii}}{\operatorname{Tr}(K)}$. This choice is motivated by the fact that the diagonal of K captures the self-similarity of each point and offers a computationally efficient surrogate for ridge leverage scores. The algorithm then selects a subset of size m, computes the corresponding kernel submatrices C and W, and uses a Nyström-style factorization to produce an approximate low-rank embedding B. The resulting approximate leverage scores are given by the quadratic form $\tilde{\ell}_i = B_i^\top (B^\top B + \sigma_\xi^2 I)^{-1} B_i$ for each point i. This method has runtime $\mathcal{O}(nm^2)$ and storage complexity $\mathcal{O}(nm)$, and provides provably accurate approximations

to the true ridge leverage scores with high probability, while being scalable to large datasets where exact score computation is infeasible.

5 Theoretical Guarantees

In this section, we present our main theoretical results for the predictive mean (Theorem 2), variance (Theorem 3), and negative log-likelihood (Theorem 4) under ridge leverage score–based sketching. We begin by outlining the setup, notation, and assumptions common to all three theorems. For better readability, we have deferred detailed proofs to the appendix.

Setup: Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be the dataset, where $x_i \in \mathbb{R}^d$ are the input points and $y_i \in \mathbb{R}$ are the corresponding outputs. Let $k(x, x')$ be the kernel function used in the Gaussian process regression, and let K be the $n \times n$ kernel matrix such that $K_{ij} = k(x_i, x_j)$. Let the eigenvalue decomposition of the kernel matrix be $K = U\Sigma U^T$.

Let $S \in \mathbb{R}^{n\times m}$ be a sketching matrix (obtained in Algorithm 1) so that $S_{ij} = \sqrt{\frac{1}{mp_i}}$ if $i = i_j$ else 0, where $m \ll n$ is obtained by probability distribution $(p_i)_{1\le i\le n}$ such that $\forall i \in \{1, \cdots, n\}, \quad p_i \ge \beta \cdot l_i(\sigma_\xi^2)/\sum_{i=1}^n l_i(\sigma_\xi^2)$ for some $\beta \in (0, 1]$. We define the sketch of a kernel matrix $L_\gamma = KS(S^\top KS + \gamma I)^{-1}S^\top K$ as the submatrix of K where $\gamma > 0$.

Moreover, let

$$D = \Phi - \Phi^{1/2}U^\top SS^\top U\Phi^{1/2}$$

with $\Phi = \Sigma(\Sigma + \gamma I)^{-1}$.

From, El Alaoui & Mahoney (2015) we have that, as long as the sketching matrix S satisfies $\lambda_{max}(D) \le t$ for $t \in (0, 1)$ and λ_{max} denoting the maximum eigenvalue, we have that

$$0 \preceq K - L_\gamma \preceq \left(\frac{\gamma}{1-t}\right) I$$

For the mean and variance inference on test data we also assume that $k_* = U\alpha$ (i.e., $\alpha_i = u_i^T k_*$) where, $\alpha \in R^n$, and $U = [u_1, \ldots, u_n]$ is the eigenvector matrix of K.

5.1 Predictive Mean Estimation using Sketching

Theorem 2 (Predictive Mean Approximation under σ_ξ^2- Ridge Leverage Score Sketching)**.** *For the notations and assumptions defined in our* ***Setup*** *let*

$$\mu(x^*) = k_*^T(K + \sigma_\xi^2 I)^{-1}y$$

be the predictive mean of Gaussian Process Regression at a new point x^* for the full kernel matrix. Here $k_* = [k(x_1, x^*), k(x_2, x^*), \ldots, k(x_n, x^*)]^T$ and σ_ξ^2 is the noise. For the sketch of a kernel matrix L_γ, the predictive mean is,

$$\mu_S(x^*) = k_*^T(L_\gamma + \sigma_\xi^2 I)^{-1}y$$

For the L_γ obtained using Algorithm 1 we have

$$|\mu(x^*) - \mu_S(x^*)| \le \left(\frac{\gamma}{1-t}\right)\sqrt{\sum_{i=1}^n \frac{\alpha_i^2}{(\Sigma_{i,i} + \sigma_\xi^2)^2}} \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D)$$

where, $\Delta_D = \left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma + \gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}$, hold with probability at least $1 - \delta$, if the sketch size m is set so that

$$m \geq 8\left(\frac{d_{\mathit{eff}}}{\beta} + \frac{1}{6}\right)\log\left(\frac{n}{\delta}\right)$$

5.2 Predictive Variance Estimation using Sketching

Theorem 3 (Predictive Variance Approximation under σ_ξ^2- Ridge Leverage Score Sketching)**.** *For the notations and assumptions defined in our* ***Setup*** *let*

$$Var(x^*) = k(x^*, x^*) - k_*^\top (K + \sigma_\xi^2 I)^{-1} k_*$$

be the predictive variance of Gaussian Process Regression at a new point x^*. Here, $k_* = [k(x_1, x^*), k(x_2, x^*), \ldots, k(x_n, x^*)]^T$ and σ_ξ^2 is the noise variance. For the sketch of a kernel matrix L_γ, the predictive variance is,

$$Var_S(x^*) = k(x^*, x^*) - k_*^\top (L_\gamma + \sigma_\xi^2 I)^{-1} k_*$$

For the L_γ obtained using Algorithm 1 we have

$$|Var(x^*) - Var_S(x^*)| \leq \left(\frac{\gamma}{1-t}\right)\left\|\alpha^\top \Delta_D\right\|_2 \cdot \sqrt{\sum_{i=1}^{n} \alpha_i^2 \left(\frac{1}{(\Sigma_{i,i} + \sigma_\xi^2)}\right)^2}$$

where, $\Delta_D = \left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma + \gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}$, hold with probability at least $1-\delta$ if the sketch size m is set so that

$$m \geq 8\left(\frac{d_{\mathit{eff}}}{\beta} + \frac{1}{6}\right)\log\left(\frac{n}{\delta}\right)$$

Remark (Interpretation of α in the Predictive Mean and Variance Bounds). In the predictive mean approximation bound, the term involving α_i^2 arises from expressing the test-to-train kernel vector $k_* \in \mathbb{R}^n$ in the eigenbasis of the kernel matrix $K = U\Sigma U^\top$, such that $k_* = U\alpha$ with $\alpha = U^\top k_*$. The coefficients α_i quantify the alignment of the test point x_* with the spectral components of the training kernel and thus determine how the eigenvalue spectrum of K influences the approximation error.

Discussion. The bound depends on the energy of the test kernel vector in the eigenbasis of K, captured by $\sum_i \alpha_i^2/(\Sigma_{ii} + \sigma_\xi^2)^2$. This term directly links the approximation quality to both the spectral decay of the kernel and the geometric relation between the test and training points. When the kernel spectrum decays rapidly—such as for smooth kernels like RBF or high-order Matern the contributions from low-eigenvalue directions are strongly attenuated, leading to tighter bounds. Similarly, a larger noise variance σ_ξ^2 regularizes the influence of small eigenvalues, further stabilizing the approximation. Hence, the derived error bounds characterize how spectral compressibility of the kernel governs the fidelity of the sketched approximation without imposing additional assumptions on the distribution of the test inputs. An analogous interpretation applies to the predictive variance bound (Theorem 7), where the vector α again captures the projection of the test point onto the eigenspace of the kernel matrix.

5.3 Negative Log Marginal Likelihood Approximation

Theorem 4 (Negative Log Marginal Likelihood Approximation under σ_ξ^2- Ridge Leverage Score Sketching)**.** *Let $K \in \mathbb{R}^{n\times n}$ be a symmetric positive semi-definite kernel matrix and $y \in \mathbb{R}^n$ the response vector. For $\sigma_\xi^2 > 0$, the negative log marginal likelihood (NLML) be,*

$$\mathcal{L}(K) = \frac{1}{2} y^\top (K + \sigma_\xi^2 I)^{-1} y + \frac{1}{2}\log\det(K + \sigma_\xi^2 I) + \frac{n}{2}\log(2\pi)$$

The corresponding approximate NLML for the sketch of the kernel matrix L_γ obtained using Algorithm 1 is given as,

$$\mathcal{L}(L_\gamma) = \frac{1}{2} y^\top (L_\gamma + \sigma_\xi^2 I)^{-1} y + \frac{1}{2} \log\det(L_\gamma + \sigma_\xi^2 I) + \frac{n}{2}\log(2\pi)$$

Then, for any $0 \leq \delta \leq 1$*, if*

$$m \geq 8\left(\frac{d_{\mathit{eff}}}{\beta} + \frac{1}{6}\right)\log\left(\frac{n}{\delta}\right)$$

then, with probability at least $1-\delta$ *following inequality holds,*

$$|\mathcal{L}(K) - \mathcal{L}(L_\gamma)| \leq \frac{\gamma}{2(1-t)}\left(\frac{1}{\lambda_{min}(K+\sigma_\xi^2 I)}\right)\cdot \|y\|_2^2 \cdot \lambda_{max}(\Delta_D) + \frac{\gamma}{2(1-t)}\mathrm{Tr}(\Delta_D)$$

where, $\Delta_D = \left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma+\gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}$.

5.4 Interpretation of Δ_D

All bounds in our analysis share a central spectral term involving the matrix

$$\Delta_D = \left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma+\gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}$$

where Σ denotes the diagonal matrix of kernel eigenvalues, $\gamma > 0$ is the regularization parameter, $t \in (0,1)$, and σ_ξ^2 is the noise.

Spectral Dependence of the Bounds. The tightness of the predictive mean, variance, and NLML bounds is depends on the terms $\lambda_{\max}(\Delta_D)$ and $\mathrm{Tr}(\Delta_D)$, both of which are minimized when the spectrum of Σ exhibits fast decay. In such cases, Δ_D becomes better conditioned, as low-eigenvalue directions are strongly regularized or suppressed by the additive noise. This yields tighter theoretical guarantees for the sketched approximation. Smooth kernels such as the RBF and high-ν Matérn families naturally induce this spectral decay, particularly when applied to well-distributed, low-dimensional inputs typical of geostatistical data or physical simulations.

6 Experiments

All experiments were conducted on a machine equipped with an NVIDIA A100 PCIe GPU with 32 GB of memory. Our implementation is written in Python and leverages PyTorch and GPyTorch Gardner et al. (2018) for efficient GPU-accelerated Gaussian Process modeling.

6.1 Datasets

We evaluate our methods on four real-world regression datasets: `California Housing` Pace & Barry (1997), `Elevators` Team (1996), `Airfoil Self-Noise` H. et al. (1999), and `Protein` Cai et al. (2003). All datasets are standardized using z-score normalization for both inputs and targets. For `California Housing` and `Protein`, we use a 70%/30% train/test split; for the others, we follow an 80%/20% split.

`California Housing` contains 20,640 samples with 8 real-valued features describing demographic and geographic attributes from the 1990 U.S. Census. The target variable is the median house value in each district, making it a widely used benchmark for medium-scale regression tasks with heterogeneous feature distributions.

`Elevators` is a large-scale regression benchmark from the DELVE framework, hosted on the UCI repository. It consists of 16,599 samples with 18 continuous features capturing the dynamics of a control system, and a real-valued target representing elevator response time. The dataset exhibits moderately complex and nonlinear patterns, making it well-suited for testing scalable GP models.

`Airfoil Self-Noise` comprises 1,503 samples with 5 continuous features representing physical properties and operating conditions of airfoils in a wind tunnel. The target is the scaled sound pressure level. Due to its small size and nonlinear behavior, it serves as a testbed for evaluating predictive uncertainty in low-data regimes.

`Protein` (also known as `Protein Structure`) is a large-scale regression dataset from the UCI repository with 45,730 samples and 9 physicochemical features describing the secondary structure of proteins. The target variable is the root mean square deviation (RMSD) of atomic positions, which measures structural variability. This dataset is widely used to benchmark scalable kernel methods due to its size, moderate dimensionality, and nonlinear structure.

6.2 Experimental Setup

We use the Radial Basis Function (RBF) kernel and Matern kernel for all three datasets. The kernel hyperparameters, including the lengthscales and variance, are initialized to 1.0 in case of RBF kernel and in Matern kernel $\nu = 1.5$ is initialized. The prior mean function is initialized as a constant set to 0 and is treated as a learnable hyperparameter during training. Based on preliminary experiments, we fix the learning rate to 0.01 and train for 300 iterations across all methods. In contrast, SVGP was trained for 1000 iterations. This is because SVGP, as a variational method, optimizes an objective (the ELBO) that iteratively approximates the true posterior, a process that generally requires more iterations to stabilize than the methods that optimize the exact marginal log-likelihood. We trained SVGP using Adam (lr=0.01) and mini-batch size 1024.

6.3 Baselines

We compare our Nystrom Ridge Leverage method against a comprehensive set of baselines spanning both the coreset selection and scalable GPR literature which are described below. All methods are evaluated over progressively increasing subset sizes, covering approximately 2% to 12% of the full training set. To ensure statistical robustness and account for variability in subset construction, each experiment is repeated across 5 independent random seeds, where each seed corresponds to a different data split and independently selected subset. We report the mean metric values across these trials, along with the corresponding standard deviations to reflect variability and robustness.

Uniform Subsampling. Uniform subsampling Hayashi et al. (2020); Malaviya et al. (2024) selects training points uniformly at random, independent of the data distribution or kernel structure. Although simple and computationally efficient, it often fails to capture important geometric or uncertainty-related aspects of the data.

Leverage Score Sampling. Leverage score sampling prioritizes points with higher statistical influence, emphasizing those contributing most to the low-rank structure of the kernel matrix Drineas et al. (2012); Zheng & Phillips (2017); Chhaya et al. (2020). This data-aware selection improves representativeness over uniform sampling and provides a foundation for more advanced sketching-based approaches.

***k*-Means Coreset** We include a k-means-based coreset baseline using the Lightweight Coreset method Bachem et al. (2018); Shit et al. (2022), which combines uniform and sensitivity-based sampling to select representative points with replacement. This efficiently approximates the data's clustering structure and scales better than exact k-means on large datasets.

Stochastic Variational Gaussian Processes (SVGP) SVGP Hensman et al. (2013) is a variational inference framework for scalable GPR that optimizes an evidence lower bound (ELBO) via stochastic gradients. It supports mini-batch training and inducing point learning, and is widely regarded as a state-of-the-art method for large-scale Gaussian Processes.

IterGP IterGP (Wenger et al., 2022) introduces a computation-aware framework for Gaussian Process inference that explicitly models both *mathematical uncertainty* (due to finite data) and *computational uncertainty* (due to approximate inference). Unlike standard approximations such as SVGP or CG-based

solvers that ignore the uncertainty introduced by limited compute, IterGP provides a combined posterior whose covariance decomposes into mathematical and computational components. This guarantees convergence of the posterior mean in RKHS norm and offers a worst-case bound on the approximation error.

Nystrom Approximations We compare three Nystrom based kernel approximation methods that differ in how the inducing points (columns of the kernel matrix) are selected. The uniform variant samples columns uniformly at random with replacement Williams & Seeger (2001), while the leverage score variant uses sampling probabilities proportional to the standard leverage scores Gittens & Mahoney (2016). Our method employs ridge leverage score sampling, which incorporates the regularization parameter and provides a data-aware, theoretically grounded alternative for constructing Nystrom approximations in GPR.

6.4 Evaluation Metrics

Model performance is evaluated using the following metrics: (i) predictive mean error, (ii) predictive variance error, (iii) root mean squared error (RMSE), (iv) Negative Log Predictive Density (NLPD), (v) Mean Standardized Log Loss (MSLL), and (vi) Negative log likelihood (NLL). Predictive mean and variance errors are computed as the relative ℓ_2 norm difference with respect to the full Gaussian Process model, defined as $\|\mu_{\text{full}} - \mu_{\text{sketch}}\|/\|\mu_{\text{full}}\|$ and $\|\sigma^2_{\text{full}} - \sigma^2_{\text{sketch}}\|/\|\sigma^2_{\text{full}}\|$, respectively.

Negative Log Predictive Density (NLPD) To evaluate the quality of uncertainty estimates in Gaussian Process Regression (GPR), we report the Negative Log Predictive Density (NLPD). NLPD measures how well the predicted Gaussian distribution aligns with the true targets, penalizing both misestimated means and variances. Formally, for test data $\{(x_i, y_i)\}_{i=1}^n$ with predictive mean μ_i and variance σ_i^2, NLPD is computed as:

$$\text{NLPD} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(y_i - \mu_i)^2}{2\sigma_i^2} + \frac{1}{2}\log(2\pi\sigma_i^2)\right) \tag{12}$$

Lower values indicate better predictive performance and better-calibrated uncertainty. NLPD is a proper scoring rule and is widely used in evaluating probabilistic regression models Artemev et al. (2021); Rasmussen & Williams (2005).

Mean Standardized Log Loss (MSLL). Unlike standard error metrics such as RMSE, MSLL evaluates how well the predictive distribution improves over a simple baseline model (typically the empirical mean and variance of the training targets). Formally, MSLL is defined as

$$\text{MSLL} = \frac{1}{n}\sum_{i=1}^{n}\left[\log p(y_i \mid x_i, \mathcal{D}_{\text{train}}) - \log p_{\text{baseline}}(y_i \mid x_i)\right], \tag{13}$$

where $p(y_i \mid x_i, \mathcal{D}_{\text{train}})$ denotes the model's predictive density and $p_{\text{baseline}}(y_i \mid x_i)$ corresponds to the baseline predictive distribution. A negative MSLL indicates that the model outperforms the baseline in terms of log predictive density. MSLL is particularly useful because it standardizes performance across datasets with different output scales and provides a more interpretable measure of probabilistic performance than raw log likelihood (Rasmussen & Williams, 2005).

6.5 Results and Analysis

Our main results, presented in the figures below and detailed in the tables in the appendices, demonstrate that the proposed Nystrom ridge leverage sketching method consistently outperforms all considered baselines across datasets, most notably on strong metrics such as NLPD and MSLL. Despite SVGP and IterGp being a state-of-the-art approach for scalable GPs, particularly in probabilistic modeling, our method achieves superior performance, especially in terms of Negative Log Predictive Density (NLPD), which is a proper scoring rule sensitive to both prediction accuracy and uncertainty calibration. Our method also achieves consistently lower predictive mean and variance errors, indicating that it more accurately approximates the true posterior distribution of the Gaussian Process Regression. Notably, the NLPD gains are achieved

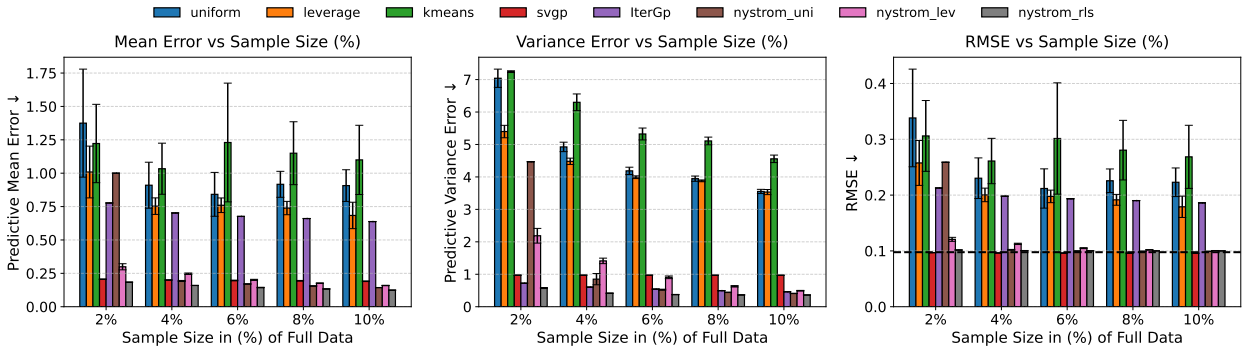

Figure 1: **Results on UCI Elevators Dataset.** Evaluation of Gaussian Process Regression methods on the UCI Elevators dataset using the **RBF kernel**. Predictive mean error, predictive variance error, and RMSE are plotted versus subset size. Ridge Leverage based GPR yields the best tradeoff across metrics. All results are averaged over 5 random trials with standard deviation shown as error bars. The dashed horizontal line indicates the performance of the full-dataset (exact GP) model.

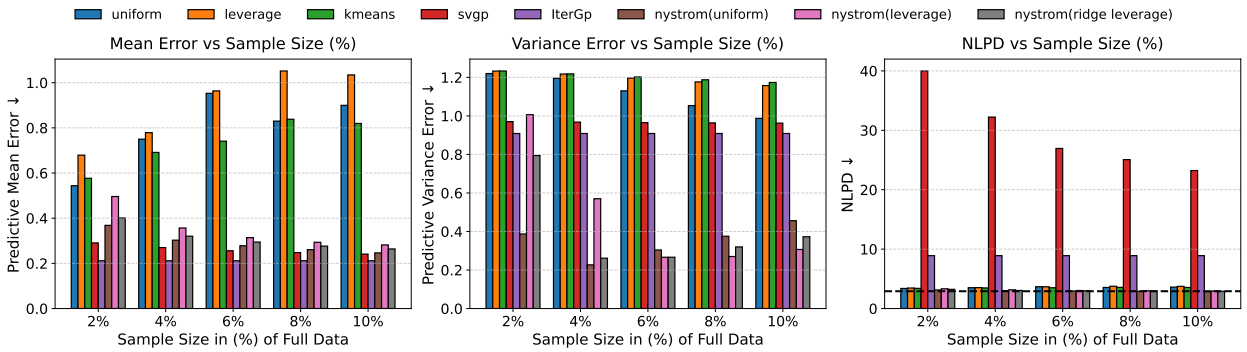

Figure 2: **Results on Protein Dataset.** Performance comparison of various Gaussian Process Regression (GPR) methods on the `Protein` dataset using the **RBF kernel**. The ridge leverage−based sketching method achieves superior predictive variance and NLPD compared to uniform, IterGp, SVGP and other baselines, demonstrating its robustness on this high-dimensional, large-scale regression task. The reported NLPD and uncertainty metrics are averaged over 5 random trials, with error bars representing standard deviations across runs. The dashed horizontal line indicates the performance of the full-dataset (exact GP) model.

using an efficient, approximate version of ridge leverage score computation Algorithm 2, showcasing the scalability and effectiveness of our approach. The best-performing results are shown in bold. We focus on these representative baselines to cover the most widely used paradigms for scalable GPR subset selection, Nystrom approximation, iterative approximation, and variational inference.

While several variational approaches to scalable Gaussian Process inference exist, we include SVGP as it remains the most widely adopted and well-established representative of this class. Additionally, we compare against IterGP, a recent state-of-the-art scalable GPR method, to benchmark our approach against the strongest contemporary baselines.

For the California Housing dataset, we were unable to include the Nystrom (leverage) baseline, as the resulting kernel matrix approximation was not positive semi-definite, which caused instability during model training. More results are included in the appendix.

Training Time and Dataset Scale Justification. In our experiments, the SVGP baseline often required *longer training time* than exact GPR despite its theoretical scalability. This effect is prominent for moderate-scale datasets ($n \approx 15$K to 30K), as GPyTorch's exact GPR leverages efficient conjugate gradient routines and optimizes only a few kernel hyperparameters, whereas SVGP jointly learns kernel and variational parameters

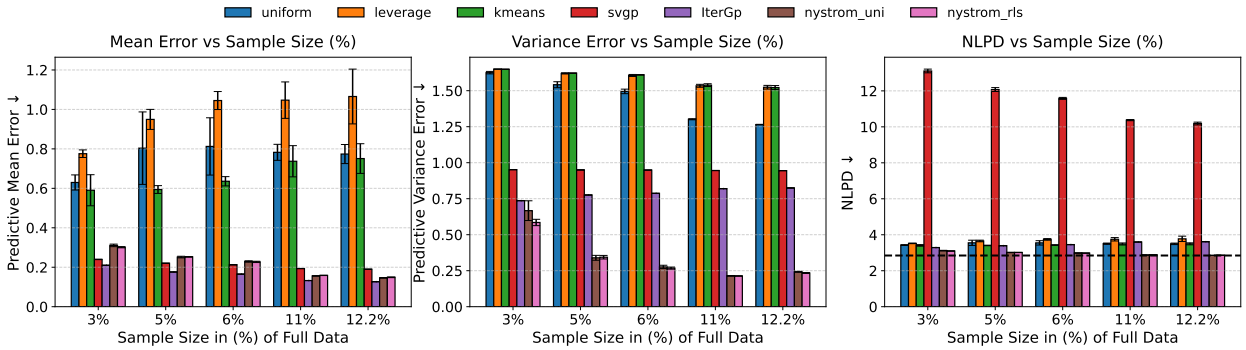

Figure 3: **Results on Protein Dataset.** Performance comparison of various Gaussian Process Regression (GPR) methods on the `Protein` dataset using the **Matern kernel**. The ridge leverage−based sketching method achieves superior predictive variance and NLPD compared to uniform, IterGp, SVGP and other baselines, demonstrating its robustness on this high-dimensional, large-scale regression task. The reported NLPD and uncertainty metrics are averaged over 5 random trials, with error bars representing standard deviations across runs. The dashed horizontal line indicates the performance of the full-dataset (exact GP) model.

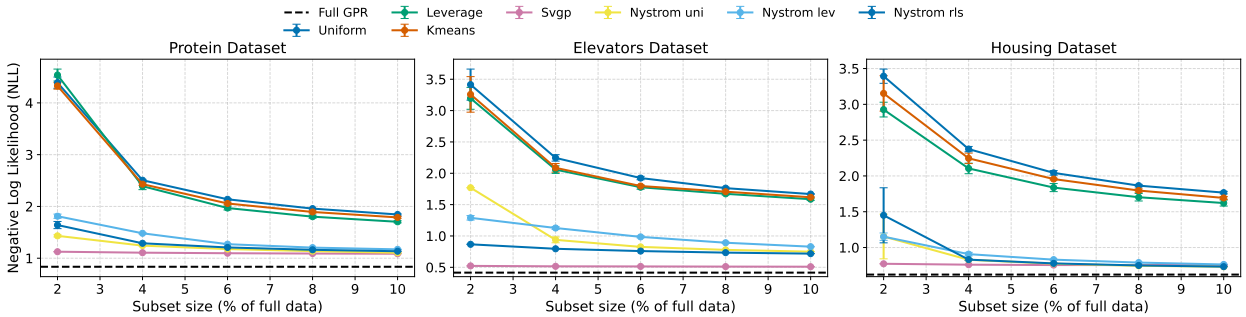

Figure 4: Comparison of Negative Log Likelihood (NLL) across different subset sizes and Gaussian Process approximation methods on the Protein, Elevators, and Housing datasets using the RBF kernel. Each subplot reports the mean and standard deviation over five random trials. The dashed horizontal line denotes the performance of the full-data (Exact GP) model.

through stochastic updates. Moreover, mini-batching and stochastic optimization introduce additional overhead at this scale, making SVGP slower in wall-clock time compared to the exact solver for moderate n, a behavior also observed in prior work (Wilson & Nickisch, 2015; Gardner et al., 2018; Pleiss et al., 2018). We therefore restrict our benchmark to the Protein dataset (n = 45,730) the largest size for which full GPR remains tractable on a 32 GB GPU. Beyond this scale, storing and inverting the full kernel ($O(n^3)$ time, $O(n^2)$ memory) becomes infeasible, preventing computation of reference quantities such as predictive mean or variance errors. This regime allows meaningful and fair comparison against scalable methods while maintaining exact GPR as a ground-truth reference (Rasmussen & Williams, 2005; Gardner et al., 2018; Wang et al., 2019).

7 Conclusion

We proposed a scalable Gaussian Process Regression method that combines Nystrom approximation with ridge leverage score sampling. While ridge leverage scores have been used in kernel ridge regression and matrix approximation, our work is the first to apply them in the Gaussian Process setting with theoretical guarantees on predictive mean, variance, and negative log-likelihood. Our analysis shows how the quality of the approximation depends on the kernel spectrum and sketch size, and our experiments demonstrate consistent improvements over existing baselines.

References

Artem Artemev, David R Burt, and Mark van der Wilk. Tighter bounds on the log marginal likelihood of gaussian process regression using conjugate gradients. In *International Conference on Machine Learning*, pp. 362–372. PMLR, 2021.

Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable k-means clustering via lightweight coresets. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1119–1127, 2018.

Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

Erdem Bıyık, Nicolas Huynh, Mykel J Kochenderfer, and Dorsa Sadigh. Active preference-based gaussian process regression for reward learning. *arXiv preprint arXiv:2005.02575*, 2020.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

David R Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. Convergence of sparse variational inference in gaussian processes regression. *Journal of Machine Learning Research*, 21(131):1–63, 2020.

C. S. S. Cai, D. J. Hand, N. M. Adams, K. Anagnostopoulos, and A. J. M. Ferreira. Protein structure dataset. `https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure`, 2003. UCI Machine Learning Repository.

Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Analysis of nyström method with sequential ridge leverage score sampling. In *Uncertainty in Artificial Intelligence*, 2016.

Yifan Chen and Yun Yang. Fast statistical leverage score approximation in kernel ridge regression. In *International Conference on Artificial Intelligence and Statistics*, pp. 2935–2943. PMLR, 2021.

Rachit Chhaya, Anirban Dasgupta, and Supratim Shit. On coresets for regularized regression. In *International conference on machine learning*, pp. 1866–1876. PMLR, 2020.

Amir Dezfouli and Edwin V Bonilla. Scalable inference for gaussian process models with black-box likelihoods. *Advances in Neural Information Processing Systems*, 28, 2015.

Kun Dong, David Eriksson, Hannes Nickisch, David Bindel, and Andrew G Wilson. Scalable log determinants for gaussian process kernel learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006a.

Petros Drineas, Michael W Mahoney, and Shan Muthukrishnan. Sampling algorithms for l 2 regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pp. 1127–1136, 2006b.

Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

Ahmed El Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. *Advances in Neural Information Processing Systems*, 28:775–783, 2015.

Tamás Erdélyi, Cameron Musco, and Christopher Musco. Fourier sparse leverage scores and approximate kernel learning. *Advances in Neural Information Processing Systems*, 33:109–122, 2020.

Christian Fiedler, Carsten W Scherer, and Sebastian Trimpe. Practical and rigorous uncertainty bounds for gaussian process regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 7439–7447, 2021.

Jacob R. Gardner, Geoff Pleiss, Kilian Q. Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Alex Gittens and Michael W Mahoney. Revisiting the nyström method for improved large-scale machine learning. *The Journal of Machine Learning Research*, 17(1):3977–4041, 2016.

Brooks T. H., D. S. Pope, and M. A. Marcolini. Airfoil self-noise data set, 1999. UCI Machine Learning Repository.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Kohei Hayashi, Masaaki Imaizumi, and Yuichi Yoshida. On random subsampling of gaussian process regression: A graphon-based analysis. In *International Conference on Artificial Intelligence and Statistics*, pp. 2055–2065. PMLR, 2020.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th international conference on computer vision*, pp. 1–8. IEEE, 2007.

Jihao Andreas Lin. Scalable gaussian processes: Advances in iterative methods and pathwise conditioning. *arXiv preprint arXiv:2507.06839*, 2025.

Jihao Andreas Lin, Sebastian Ament, Maximilian Balandat, David Eriksson, José Miguel Hernández-Lobato, and Eytan Bakshy. Scalable gaussian processes with latent kronecker structure. *arXiv preprint arXiv:2506.06895*, 2025.

Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When gaussian process meets big data: A review of scalable gps. *IEEE transactions on neural networks and learning systems*, 31(11):4405–4423, 2020.

Yifan Lu, Jiayi Ma, Leyuan Fang, Xin Tian, and Junjun Jiang. Robust and scalable gaussian process regression and its applications. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21950–21959, 2023.

Jayesh Malaviya, Anirban Dasgupta, and Rachit Chhaya. Simple weak coresets for non-decomposable classification measures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14289–14296, 2024.

Cameron Musco and Christopher Musco. Recursive sampling for the nyström method. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297, 1997.

Misha Padidar, Xinran Zhu, Leo Huang, Jacob Gardner, and David Bindel. Scaling gaussian processes with derivative information using variational inference. *Advances in Neural Information Processing Systems*, 34: 6442–6453, 2021.

Mert Pilanci and Martin J Wainwright. Randomized sketches of convex programs with sharp guarantees. *IEEE Transactions on Information Theory*, 63(9):5753–5783, 2017.

Geoff Pleiss, Jacob Gardner, Kilian Weinberger, and Andrew Gordon Wilson. Constant-time predictive distributions for gaussian processes. In *International Conference on Machine Learning*, pp. 4114–4123. PMLR, 2018.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*. MIT press, Cambridge, MA, 2005.

Alessandro Rudi and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Jens Schreiter, Duy Nguyen-Tuong, Mona Eberts, Bastian Bischoff, Heiner Markert, and Marc Toussaint. Safe exploration for active learning with gaussian processes. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 133–149. Springer, 2015.

Jiaxin Shi, Mohammad Emtiyaz Khan, and Jun Zhu. Scalable training of inference networks for gaussian-process models. In *International Conference on Machine Learning*, pp. 5758–5768. PMLR, 2019.

Supratim Shit, Anirban Dasgupta, Rachit Chhaya, and Jayesh Choudhari. Online coresets for parameteric and non-parametric bregman clustering. *Transactions on Machine Learning Research*, 2022.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pp. 1257–1264, 2006.

The DELVE Team. Individual household electric power consumption data set (elevators), 1996. UCI Machine Learning Repository.

Jörn Tebbe, Christoph Zimmer, Ansgar Steland, Markus Lange-Hegermann, and Fabian Mies. Efficiently computable safety bounds for gaussian processes in active learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1333–1341. PMLR, 2024.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

Ke Alexander Wang, Geoff Pleiss, Jacob R. Gardner, Stephen Tyree, Kilian Q. Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Jonathan Wenger, Geoff Pleiss, Marvin Pförtner, Philipp Hennig, and John P Cunningham. Posterior and computational uncertainty in gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pp. 682–688, 2001.

Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning (ICML)*, 2015.

Andrew Gordon Wilson, Christoph Dann, and Hannes Nickisch. Thoughts on massively scalable gaussian processes. *arXiv preprint arXiv:1511.01870*, 2015.

David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Zhitong Xu, Haitao Wang, Jeff M Phillips, and Shandian Zhe. Standard gaussian process is all you need for high-dimensional bayesian optimization. *arXiv preprint arXiv:2402.02746*, 2024.

Yan Zheng and Jeff M Phillips. Coresets for kernel regression. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 645–654, 2017.

A Appendix

In this section, we present our main theoretical results for the predictive mean (Theorem 5), variance (Theorem 7), and negative log-likelihood (Theorem 6) under ridge leverage score–based sketching. We begin by outlining the setup, notation, and assumptions common to all three theorems.

A.1 Common Theoretical Setup:

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be the dataset, where $x_i \in \mathbb{R}^d$ are the input points and $y_i \in \mathbb{R}$ are the corresponding outputs. Let $k(x, x')$ be the kernel function used in the Gaussian process regression, and let K be the kernel matrix $n \times n$ such that $K_{ij} = k(x_i, x_j)$. Let the eigenvalue decomposition of the kernel matrix, $K = U\Sigma U^T$.

Let $S \in \mathbb{R}^{n\times m}$ be a sketching matrix (obtained in Algorithm 1) so that $S_{ij} = \sqrt{\frac{1}{mp_i}}$ if $i = i_j$ else 0, where $m \ll n$ is obtained by probability distribution $(p_i)_{1\le i\le n}$ such that $\forall i \in \{1, \cdots, n\}, \quad p_i \ge \beta \cdot l_i(\sigma_\xi^2)/\sum_{i=1}^n l_i(\sigma_\xi^2)$ for some $\beta \in (0, 1]$.
We define the sketch of a kernel matrix $L_\gamma = KS(S^\top KS + \gamma I)^{-1}S^\top K$ as the submatrix of K where $\gamma > 0$.

Moreover, let

$$D = \Phi - \Phi^{1/2}U^\top SS^\top U\Phi^{1/2}$$

with $\Phi = \Sigma(\Sigma + \gamma I)^{-1}$.

Following the result of El Alaoui & Mahoney (2015), we assume that the sketching matrix $S \in \mathbb{R}^{n\times m}$ satisfies the spectral condition

$$\lambda_{\max}(D) \le t \quad \text{for some } t \in (0, 1)$$

where $\lambda_{\max}$ denotes the maximum eigenvalue. Under this condition, they show that the approximation error between the original kernel matrix K and its sketched version L_γ is bounded as

$$0 \preceq K - L_\gamma \preceq \left(\frac{\gamma}{1-t}\right) I$$

To ensure this bound holds, we follow the **sketch size guarantee** provided in Theorem 2 of Appendix B in El Alaoui & Mahoney (2015), which characterizes the **required number of samples** m based on the ridge leverage score distribution. This setup is used as the basis for all theoretical results presented in the subsequent sections.

For the mean and variance inference on test data we also assume that $k_* = U\alpha$ (i.e., $\alpha_i = u_i^T k_*$) where, $\alpha \in R^n$, and $U = [u_1, \ldots, u_n]$ is the eigenvector matrix of K.

B Predictive Mean Estimation using Sketching

Theorem 5 (Predictive Mean Approximation under σ_ξ^2- Ridge Leverage Score Sketching)**.** *For the notations and assumptions defined in our **Setup** let*

$$\mu(x^*) = k_*^T(K + \sigma_\xi^2 I)^{-1}y$$

be the predictive mean of Gaussian Process Regression at a new point x^* for the full kernel matrix. Here $k_* = [k(x_1, x^*), k(x_2, x^*), \ldots, k(x_n, x^*)]^T$ and σ_ξ^2 is the noise. For the sketch of a kernel matrix L_γ, the predictive mean is,

$$\mu_S(x^*) = k_*^T(L_\gamma + \sigma_\xi^2 I)^{-1}y$$

For the L_γ obtained using Algorithm 1 we have

$$|\mu(x^*) - \mu_S(x^*)| \le \left(\frac{\gamma}{1-t}\right)\sqrt{\sum_{i=1}^n \frac{\alpha_i^2}{(\Sigma_{i,i} + \sigma_\xi^2)^2}} \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D)$$

where, $\Delta_D = \left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma + \gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}$, holds with probability at least $1 - \delta$, if the sketch size m is set so that

$$m \geq 8\left(\frac{d_{\mathit{eff}}}{\beta} + \frac{1}{6}\right)\log\left(\frac{n}{\delta}\right)$$

Proof.

$$\begin{aligned}
|\mu(x^*) - \mu_S(x^*)| &= \left|k_*^T(K+\sigma_\xi^2 I)^{-1}y - k_*^T(L_\gamma+\sigma_\xi^2 I)^{-1}y\right| \\
&= \left|k_*^T[(K+\sigma_\xi^2 I)^{-1} - (L_\gamma+\sigma_\xi^2 I)^{-1}]y\right| \\
&\leq \left\|k_*^T[(K+\sigma_\xi^2 I)^{-1} - (L_\gamma+\sigma_\xi^2 I)^{-1}]\right\|_2 \cdot \|y\|_2 \\
&= \left\|k_*^T[(K+\sigma_\xi^2 I)^{-1}(L_\gamma - K)(L_\gamma+\sigma_\xi^2 I)^{-1}]\right\|_2 \cdot \|y\|_2 \\
&= \left\|k_*^T[(K+\sigma_\xi^2 I)^{-1}(K - L_\gamma)(L_\gamma+\sigma_\xi^2 I)^{-1}]\right\|_2 \cdot \|y\|_2 \\
&\leq \left\|k_*^T[(K+\sigma_\xi^2 I)^{-1}(K - L_\gamma)]\right\|_2 \left\|(L_\gamma+\sigma_\xi^2 I)^{-1}\right\|_{op} \cdot \|y\|_2 \\
&\leq \left(\frac{\gamma}{1-t}\right)\left\|k_*^T(K+\sigma_\xi^2 I)^{-1}\right\|_2 \left\|(L_\gamma+\sigma_\xi^2 I)^{-1}\right\|_{op} \cdot \|y\|_2 \qquad (\text{As, } 0 \preceq K - L_\gamma \preceq \left(\tfrac{\gamma}{1-t}\right) I) \\
&\leq \left(\frac{\gamma}{1-t}\right)\left\|k_*^T(K+\sigma_\xi^2 I)^{-1}\right\|_2 \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D) \\
&\qquad\qquad (\text{Putting bound on, } \left\|(L_\gamma+\sigma_\xi^2 I)^{-1}\right\|_{op} \text{ from equation 14 below}) \\
&= \left(\frac{\gamma}{1-t}\right)\left\|k_*^T(U\Sigma U^\top+\sigma_\xi^2 I)^{-1}\right\|_2 \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D) \\
&= \left(\frac{\gamma}{1-t}\right)\left\|k_*^\top(U(\Sigma+\sigma_\xi^2 I)U^\top)^{-1}\right\|_2 \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D) \\
&= \left(\frac{\gamma}{1-t}\right)\left\|k_*^\top(U(\Sigma+\sigma_\xi^2 I)^{-1}U^\top)\right\|_2 \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D) \\
&= \left(\frac{\gamma}{1-t}\right)\left\|(U\alpha)^\top(U(\Sigma+\sigma_\xi^2 I)^{-1}U^\top)\right\|_2 \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D) \qquad (\text{Substitute, } k_* = U\alpha) \\
&= \left(\frac{\gamma}{1-t}\right)\left\|\alpha^\top U^\top U(\Sigma+\sigma_\xi^2 I)^{-1}U^\top\right\|_2 \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D) \\
&= \left(\frac{\gamma}{1-t}\right)\left\|\alpha^\top(\Sigma+\sigma_\xi^2 I)^{-1}\right\|_2 \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D)
\end{aligned}$$

The spectral norm decomposes as:

$$\left\|\alpha^T(\Sigma+\sigma_\xi^2 I)^{-1}\right\|_2 = \sqrt{\sum_{i=1}^n \alpha_i^2\left(\frac{1}{(\Sigma_{i,i}+\sigma_\xi^2)}\right)^2}$$

Therefore, in the final bound we have

$$|\mu(x^*) - \mu_S(x^*)| \leq \left(\frac{\gamma}{1-t}\right)\sqrt{\sum_{i=1}^n \frac{\alpha_i^2}{(\Sigma_{i,i}+\sigma_\xi^2)^2}} \cdot \|y\|_2 \cdot \lambda_{max}(\Delta_D)$$

□

C Negative Log Marginal Likelihood Approximation

Theorem 6 (Negative Log Marginal Likelihood Approximation under σ_ξ^2- Ridge Leverage Score Sketching)**.** *Let $K \in \mathbb{R}^{n\times n}$ be a symmetric positive semi-definite kernel matrix and $y \in \mathbb{R}^n$ the response vector. For $\sigma_\xi^2 > 0$, the negative log marginal likelihood (NLML) be,*

$$\mathcal{L}(K) = \frac{1}{2} y^\top (K + \sigma_\xi^2 I)^{-1} y + \frac{1}{2} \log\det(K + \sigma_\xi^2 I) + \frac{n}{2}\log(2\pi)$$

The corresponding approximate NLML for the sketch of the kernel matrix L_γ obtained using Algorithm 1 is given as,

$$\mathcal{L}(L_\gamma) = \frac{1}{2} y^\top (L_\gamma + \sigma_\xi^2 I)^{-1} y + \frac{1}{2} \log\det(L_\gamma + \sigma_\xi^2 I) + \frac{n}{2}\log(2\pi)$$

Then, for any $0 \leq \delta \leq 1$, if

$$m \geq 8\left(\frac{d_{\mathit{eff}}}{\beta} + \frac{1}{6}\right)\log\left(\frac{n}{\delta}\right)$$

then, with probability at least $1-\delta$ following inequality holds,

$$|\mathcal{L}(K) - \mathcal{L}(L_\gamma)| \leq \frac{\gamma}{2(1-t)}\left(\frac{1}{\lambda_{min}(K + \sigma_\xi^2 I)}\right) \cdot \|y\|_2^2 \cdot \lambda_{max}(\Delta_D) + \frac{\gamma}{2(1-t)}\mathrm{Tr}(\Delta_D)$$

where, $\Delta_D = \left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma + \gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}$.

Proof. To analyze the negative log marginal likelihood (NLL), we omit the additive constant term involving $\frac{n}{2}\log(2\pi)$, as it does not affect the optimization or approximation. The resulting expression consists of two principal components: a quadratic term and a log-determinant term. We derive separate bounds for each of these components and then combine them to obtain an overall bound on the NLML approximation error.

$$\mathcal{L}(K) = \frac{1}{2} y^\top (K + \sigma_\xi^2 I)^{-1} y + \frac{1}{2} \log\det(K + \sigma_\xi^2 I) + \frac{n}{2}\log(2\pi)$$

Since, we have $0 \preceq K - L_\gamma \preceq \left(\frac{\gamma}{1-t}\right) I$ bound from El Alaoui & Mahoney (2015),

$$\begin{aligned}
\frac{1}{2}\left|y^T(K + \sigma_\xi^2 I)^{-1} y - y^T(L_\gamma + \sigma_\xi^2 I)^{-1} y\right| &= \frac{1}{2}\left|y^T[(K + \sigma_\xi^2 I)^{-1} - (L_\gamma + \sigma_\xi^2 I)^{-1}]y\right| \\
&\leq \frac{1}{2}\left\|(K + \sigma_\xi^2 I)^{-1} - (L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op} \cdot \|y\|_2^2 \\
&= \frac{1}{2}\left\|(K + \sigma_\xi^2 I)^{-1}(L_\gamma - K)(L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op} \cdot \|y\|_2^2 \\
&= \frac{1}{2}\left\|(K + \sigma_\xi^2 I)^{-1}(K - L_\gamma)(L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op} \cdot \|y\|_2^2 \\
&\leq \frac{1}{2}\left\|(K + \sigma_\xi^2 I)^{-1}(K - L_\gamma)\right\|_{op}\left\|(L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op} \cdot \|y\|_2^2 \\
&\leq \frac{\gamma}{2(1-t)}\left\|(K + \sigma_\xi^2 I)^{-1}\right\|_{op}\left\|(L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op} \cdot \|y\|_2^2 \\
&\qquad\qquad \text{(As, } 0 \preceq K - L_\gamma \preceq \left(\tfrac{\gamma}{1-t}\right) I\text{)} \\
&\leq \frac{\gamma}{2(1-t)}\left\|(K + \sigma_\xi^2 I)^{-1}\right\|_{op} \cdot \|y\|_2^2 \cdot \lambda_{max}(\Delta_D) \\
&\qquad \text{(Putting bound on, } \left\|(L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op} \text{ from equation 14 below)} \\
&\leq \frac{\gamma}{2(1-t)}\left(\frac{1}{\lambda_{min}(K + \sigma_\xi^2 I)}\right) \cdot \|y\|_2^2 \cdot \lambda_{max}(\Delta_D)
\end{aligned}$$

For, $\left\|(L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op}$ we can get upper bound like following,

$$L_\gamma = KS\left(S^T KS + \gamma I\right)^{-1} S^T K$$

With $K = U\Sigma U^\top$ and $R = \Sigma^{1/2}U^\top S$, $\bar{L}_\gamma = R(R^\top R + \gamma I)^{-1}R^\top$, we have

$$L_\gamma = U\Sigma^{1/2}\bar{L}_\gamma\Sigma^{1/2}U^\top$$

Due to the matrix inversion lemma, we have

$$\begin{aligned}
\bar{L}_\gamma &= RR^\top(RR^\top + \gamma I)^{-1} \\
&= I - \gamma(RR^\top + \gamma I)^{-1} \\
&= I - \gamma(\Sigma + \gamma I + RR^\top - \Sigma)^{-1} \\
&= I - \gamma(\Sigma + \gamma I)^{-1/2}(I - D)^{-1}(\Sigma + \gamma I)^{-1/2}
\end{aligned}$$

with

$$\begin{aligned}
D &= (\Sigma + \gamma I)^{-1/2}(\Sigma - RR^\top)(\Sigma + \gamma I)^{-1/2} \\
&= \Phi - \Phi^{1/2}U^\top SS^\top U\Phi^{1/2}
\end{aligned}$$

and $\Phi = \Sigma(\Sigma + \gamma I)^{-1}$.

If the sketching matrix S satisfies $\lambda_{max}\left(\Phi - \Phi^{1/2}U^\top SS^\top U\Phi^{1/2}\right) = \lambda_{max}\left(D\right) \leq t$ for $t \in (0,1)$ where λ_{max} denotes the maximum eigenvalue we can derive the lower bound for $\bar{L_\gamma}$ as the following,

$$\begin{aligned}
\bar{L}_\gamma &= I - \gamma(\Sigma + \gamma I)^{-1/2}(I - D)^{-1}(\Sigma + \gamma I)^{-1/2} \\
&\succeq I - \gamma(\Sigma + \gamma I)^{-1/2}\left(\frac{1}{1-t}\right)I(\Sigma + \gamma I)^{-1/2} && \text{(As, } (I-D)^{-1} \preceq \left(\tfrac{1}{1-t}\right)I\text{)} \\
&= I - \left(\frac{\gamma}{1-t}\right)(\Sigma + \gamma I)^{-1}
\end{aligned}$$

Now, lets put this $\bar{L}_\gamma$ into L_γ,

$$\begin{aligned}
L_\gamma &= U\Sigma^{1/2}\bar{L}_\gamma\Sigma^{1/2}U^T \\
&\succeq U\Sigma^{1/2}\left[I - \left(\frac{\gamma}{1-t}\right)(\Sigma + \gamma I)^{-1}\right]\Sigma^{1/2}U^T \\
&= U\left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma + \gamma I)^{-1}\right]U^T \\
&= L_\gamma^{'}
\end{aligned}$$

Therefore, $L_\gamma^{'} = U\left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma + \gamma I)^{-1}\right]U^T$ is the lower bound for L_γ.

For the upper bound of $\left\|(L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op}$ term we need lower bound on L_γ which is $L_\gamma^{'}$,

$$\begin{aligned}\left\|(L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op} &\leq \left\|(L_\gamma^{'} + \sigma_\xi^2 I)^{-1}\right\|_{op} \\ &= \left\|\left(U\left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma+\gamma I)^{-1}\right]U^T + \sigma_\xi^2 I\right)^{-1}\right\|_{op} \\ &= \left\|\left(U\left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma+\gamma I)^{-1} + \sigma_\xi^2 I\right]U^T\right)^{-1}\right\|_{op}\end{aligned}$$

Using the eigendecomposition $K = U\Sigma U^\top$, and orthogonality of U, the operator norm simplifies as following, Given the eigendecomposition of the kernel matrix $K = U\Sigma U^\top$, where $\Sigma = \text{diag}(\lambda_1, \ldots, \lambda_n)$, the following expression arises in the analysis,

$$\left\|\left(U\left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma+\gamma I)^{-1} + \sigma_\xi^2 I\right]U^\top\right)^{-1}\right\|_{\text{op}} = \left\|\left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma+\gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}\right\|_{\text{op}}$$

Lets denote, $\Delta_D = \left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma+\gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}$ then we have,

$$\left\|(L_\gamma + \sigma_\xi^2 I)^{-1}\right\|_{op} \leq \lambda_{max}(\Delta_D) \tag{14}$$

Now, lets bound log determinant term,

$$\frac{1}{2}\left|log\left|(K + \sigma_\xi^2 I)\right| - log\left|(L_\gamma + \sigma_\xi^2 I)\right|\right|$$

Now, we use proposition Bhatia (2013); Boyd & Vandenberghe (2004) which says for any two symmetric positive semi-definite matrix, $A \succ 0$ and $0 \preceq \Delta \preceq \alpha I$ following inequality holds,

$$|log\,|A + \Delta| - log\,|A|| \leq Tr(A^{-1}\Delta) \leq \alpha Tr(A^{-1})$$

In above proposition if, $A = L_\gamma + \sigma_\xi^2 I$, $\Delta = K - L_\gamma$ and $\alpha = \left(\frac{\gamma}{1-t}\right)$ then,

$$
\begin{aligned}
|log\,|A+\Delta| - log\,|A|| &= \left|log\left|(L_\gamma + \sigma_\xi^2 I) + (K - L_\gamma)\right| - log\left|L_\gamma + \sigma_\xi^2 I\right|\right| \\
&= \left|log\left|(K + \sigma_\xi^2 I)\right| - log\left|(L_\gamma + \sigma_\xi^2 I)\right|\right| \\
&\leq Tr\left[(L_\gamma + \sigma_\xi^2 I)^{-1}(K - L_\gamma)\right] \\
&\leq Tr\left[(L_\gamma + \sigma_\xi^2 I)^{-1}\left(\frac{\gamma}{1-t}\right) I\right] \\
&= \left(\frac{\gamma}{1-t}\right) Tr\left[(L_\gamma + \sigma_\xi^2 I)^{-1}\right] \\
&\leq \left(\frac{\gamma}{1-t}\right) Tr\left[(L_\gamma^{'} + \sigma_\xi^2 I)^{-1}\right] \qquad \text{(Since, } L_\gamma \succeq L_\gamma^{'} \text{ and } L_\gamma \text{ is PSD)} \\
&= \left(\frac{\gamma}{1-t}\right) Tr\left[\left(U\left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma+\gamma I)^{-1}\right]U^T + \sigma_\xi^2 I\right)^{-1}\right] \\
&= \left(\frac{\gamma}{1-t}\right) Tr\left[\left(U\left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma+\gamma I)^{-1} + \sigma_\xi^2 I\right]U^\top\right)^{-1}\right] \\
&= \left(\frac{\gamma}{1-t}\right) \mathrm{Tr}\left[\left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma+\gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}\right] \\
&\qquad\qquad \text{(Using the orthogonality of } U\text{)}
\end{aligned}
$$

Therefore bound for log determinant term is,

$$
\frac{1}{2}\left|log\left|(K + \sigma_\xi^2 I)\right| - log\left|(L_\gamma + \sigma_\xi^2 I)\right|\right| \leq \frac{\gamma}{2(1-t)}\mathrm{Tr}\left[\left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma+\gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}\right]
$$

Since we are saying, $\Delta_D = \left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma+\gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}$, we have,

$$
\frac{1}{2}\left|log\left|(K + \sigma_\xi^2 I)\right| - log\left|(L_\gamma + \sigma_\xi^2 I)\right|\right| \leq \frac{\gamma}{2(1-t)}\mathrm{Tr}(\Delta_D)
$$

Now, combining both terms bound we will get overall bound,

$$
|\mathcal{L}(K) - \mathcal{L}(L_\gamma)| \leq \frac{\gamma}{2(1-t)}\left[\left(\frac{1}{\lambda_{min}(K + \sigma_\xi^2 I)}\right)\cdot \|y\|_2^2 \cdot \lambda_{max}(\Delta_D) + \mathrm{Tr}(\Delta_D)\right]
$$

□

D Predictive Variance Estimation using Sketching

Theorem 7 (Predictive Variance Approximation under σ_ξ^2– Ridge Leverage Score Sketching)**.** *For the notations and assumptions defined in our* ***Setup*** *let*

$$
Var(x^*) = k(x^*, x^*) - k_*^\top (K + \sigma_\xi^2 I)^{-1} k_*
$$

be the predictive variance of Gaussian Process Regression at a new point x^*. Here, $k_* = [k(x_1, x^*), k(x_2, x^*), \ldots, k(x_n, x^*)]^T$ and σ_ξ^2 is the noise variance. For the sketch of a kernel matrix L_γ, the predictive variance is,

$$
Var_S(x^*) = k(x^*, x^*) - k_*^\top (L_\gamma + \sigma_\xi^2 I)^{-1} k_*
$$

For the L_γ obtained using Algorithm 1 we have

$$|Var(x^*) - Var_S(x^*)| \leq \left(\frac{\gamma}{1-t}\right) \left\|\alpha^\top \Delta_D\right\|_2 \cdot \sqrt{\sum_{i=1}^{n} \alpha_i^2 \left(\frac{1}{(\Sigma_{i,i} + \sigma_\xi^2)}\right)^2}$$

where, $\Delta_D = \left(\Sigma\left[I - \frac{\gamma}{1-t}(\Sigma + \gamma I)^{-1}\right] + \sigma_\xi^2 I\right)^{-1}$, holds with probability at least $1-\delta$ if the sketch size m is set so that

$$m \geq 8\left(\frac{d_{\mathit{eff}}}{\beta} + \frac{1}{6}\right)\log\left(\frac{n}{\delta}\right)$$

Proof.

$$\begin{aligned}
|Var(x^*) - Var_S(x^*)| &= \left|k(x^*,x^*) - k_*^T(K + \sigma_\xi^2 I)^{-1}k_* - k(x^*,x^*) + k_*^T(L_\gamma + \sigma_\xi^2 I)^{-1}k_*\right| \\
&= \left|-k_*^T(K + \sigma_\xi^2 I)^{-1}k_* + k_*^T(L_\gamma + \sigma_\xi^2 I)^{-1}k_*\right| \\
&= \left|k_*^T(L_\gamma + \sigma_\xi^2 I)^{-1}k_* - k_*^T(K + \sigma_\xi^2 I)^{-1}k_*\right| \\
&= \left|k_*^T[(L_\gamma + \sigma_\xi^2 I)^{-1} - (K + \sigma_\xi^2 I)^{-1}]k_*\right| \\
&= \left|k_*^T[(L_\gamma + \sigma_\xi^2 I)^{-1}(K - L_\gamma)(K + \sigma_\xi^2 I)^{-1}]k_*\right| \\
&\leq \left(\frac{\gamma}{1-t}\right)\left|k_*^T[(L_\gamma + \sigma_\xi^2 I)^{-1}(K + \sigma_\xi^2 I)^{-1}]k_*\right| \qquad (\text{As, } 0 \preceq K - L_\gamma \preceq \left(\tfrac{\gamma}{1-t}\right) I) \\
&= \left(\frac{\gamma}{1-t}\right)\left|k_*^T[(L_\gamma + \sigma_\xi^2 I)^{-1}(U\Sigma U^\top + \sigma_\xi^2 I)^{-1}]k_*\right| \\
&= \left(\frac{\gamma}{1-t}\right)\left|k_*^T[(L_\gamma + \sigma_\xi^2 I)^{-1}(U(\Sigma + \sigma_\xi^2 I)U^\top)^{-1}]k_*\right| \\
&= \left(\frac{\gamma}{1-t}\right)\left|(U\alpha)^T[(L_\gamma + \sigma_\xi^2 I)^{-1}(U(\Sigma + \sigma_\xi^2 I)^{-1}U^\top)](U\alpha)\right| \qquad (\text{Substitute, } k_* = U\alpha) \\
&\leq \left(\frac{\gamma}{1-t}\right)\left|\alpha^\top U^\top (L_\gamma' + \sigma_\xi^2 I)^{-1}(U(\Sigma + \sigma_\xi^2 I)^{-1}U^\top)U\alpha\right| \qquad (\text{Since, } L_\gamma \succeq L_\gamma') \\
&= \left(\frac{\gamma}{1-t}\right)\left|\alpha^\top U^\top \left(U\left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma + \gamma I)^{-1}\right]U^T + \sigma_\xi^2 I\right)^{-1}(U(\Sigma + \sigma_\xi^2 I)^{-1}U^\top)U\alpha\right| \\
&= \left(\frac{\gamma}{1-t}\right)\left|\alpha^\top U^\top \left(U\left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma + \gamma I)^{-1} + \sigma_\xi^2 I\right]U^T\right)^{-1}(U(\Sigma + \sigma_\xi^2 I)^{-1}U^\top)U\alpha\right| \\
&= \left(\frac{\gamma}{1-t}\right)\left|\alpha^\top \left[\Sigma - \left(\frac{\gamma}{1-t}\right)\Sigma(\Sigma + \gamma I)^{-1} + \sigma_\xi^2 I\right]^{-1}(\Sigma + \sigma_\xi^2 I)^{-1}\alpha\right| \\
&= \left(\frac{\gamma}{1-t}\right)\left|\alpha^\top \Delta_D(\Sigma + \sigma_\xi^2 I)^{-1}\alpha\right| \qquad (\text{As, } \Delta_D = \left[\Sigma - \left(\tfrac{\gamma}{1-t}\right)\Sigma(\Sigma + \gamma I)^{-1} + \sigma_\xi^2 I\right]^{-1}) \\
&\leq \left(\frac{\gamma}{1-t}\right)\left\|\alpha^\top \Delta_D\right\|_2 \left\|(\Sigma + \sigma_\xi^2 I)^{-1}\alpha\right\|_2
\end{aligned}$$

The spectral norm decomposes as:

$$\left\|(\Sigma + \sigma_\xi^2 I)^{-1}\alpha\right\|_2 = \sqrt{\sum_{i=1}^{n} \alpha_i^2 \left(\frac{1}{(\Sigma_{i,i} + \sigma_\xi^2)}\right)^2}$$

Therefore, in the final bound we have

$$|Var(x^*) - Var_S(x^*)| \leq \left(\frac{\gamma}{1-t}\right) \left\|\alpha^\top \Delta_D\right\|_2 \cdot \sqrt{\sum_{i=1}^{n} \alpha_i^2 \left(\frac{1}{(\Sigma_{i,i} + \sigma_\xi^2)}\right)^2}$$

□

E Additional Experimental Details

In this section, we provide supplementary experimental results obtained using the *Matern kernel* in addition to the RBF kernel used in the main paper. These additional experiments are designed to validate the robustness of our proposed Nystrom ridge leverage score sketching method across different kernel choices.

Consistent with our findings in the main text, the results with the Matern kernel confirm that our method continues to *outperform all baseline methods* across multiple datasets and evaluation metrics. In particular, it achieves *superior performance in terms of Negative Log Predictive Density (NLPD)*, a proper scoring rule that captures both prediction accuracy and uncertainty calibration. Our approach also yields *lower predictive mean and variance errors*, indicating more accurate posterior approximations of the underlying Gaussian Process Regression model.

These trends hold across all evaluated datasets, and the improvements remain significant despite the change in kernel. Importantly, the *approximate ridge leverage score algorithm* (used to efficiently construct the sketch matrix) remains effective, demonstrating both *scalability and predictive reliability* of our framework. Results from these additional experiments are presented in accompanying tables and figures, where the best-performing entries are shown in bold. We note that while several scalable variational methods exist, *SVGP* is included as a strong and widely adopted representative of the variational family. Our method demonstrates consistently lower *NLPD* and better uncertainty calibration than *SVGP*, *IterGP*, and other baselines across all kernel configurations, highlighting its robustness and effectiveness.

As shown in Table 1, full-data GP requires approximately 23.48 GB to store the kernel matrix for the Protein dataset (46K samples), reflecting its $O(n^2)$ memory complexity. In contrast, our Nystrom (Ridge Leverage) sketch avoids forming the full kernel, using only 2.25 to 5.64 GB depending on the sketch size.

Table 1: Memory usage (MB) across subset sizes on the Protein dataset (around 46K samples). All values are averaged over five random trials with standard deviations shown after ±. Full-data GPR requires around 23.48GB.

Method	**2%**	**4%**	**6%**	**8%**	**10%**
Full Data (Exact GP)			**23477.29**		
Uniform	6709.6 ± 3678.6	4109.7 ± 23.1	4178.4 ± 7.8	4302.0 ± 10.6	4424.3 ± 13.7
Leverage	4121.3 ± 19.1	4109.2 ± 23.5	4178.3 ± 7.6	4302.0 ± 10.6	4424.1 ± 13.6
KMeans	4121.9 ± 18.7	4109.8 ± 23.2	4178.3 ± 7.6	4302.0 ± 10.6	4424.3 ± 13.7
SVGP	$\mathbf{1756.8 \pm 1657.6}$	$\mathbf{1121.5 \pm 73.2}$	$\mathbf{1734.7 \pm 73.7}$	2372.0 ± 73.3	3036.2 ± 73.9
IterGP	2034.7 ± 0.0	2034.7 ± 0.0	2034.7 ± 0.0	$\mathbf{2034.7 \pm 0.0}$	$\mathbf{2034.7 \pm 0.0}$
Nyström (Uniform)	1953.9 ± 75.5	2842.3 ± 79.6	3753.2 ± 84.8	4687.5 ± 89.2	5644.0 ± 93.7
Nyström (Leverage)	2255.6 ± 347.5	2843.8 ± 79.4	3754.7 ± 84.9	4689.1 ± 89.0	5645.5 ± 93.6
Nyström (Ridge Leverage)	2253.5 ± 347.7	2842.4 ± 79.8	3753.7 ± 84.9	4687.7 ± 89.2	5644.3 ± 93.7

Table 2: Comparison of negative log predictive density (NLPD), mean standardized log loss (MSLL) and runtime (in seconds) for Gaussian Process Regression (GPR) methods on the UCI Elevators dataset using the RBF kernel. Each experiment is repeated over 5 random seeds, and we report the mean and standard deviation across runs. Lower values are better for all metrics.

Method	**Subset %**	**NLPD**	**MSLL**	**Time (s)**
	Full Data	-0.7540	-0.8224	38.45
Uniform	2%	0.4791 ± 0.1192	0.4107 ± 0.1192	5.6802 ± 0.0415
Leverage	2%	0.2869 ± 0.0543	0.2185 ± 0.0543	5.6877 ± 0.0433
Kmeans	2%	0.4359 ± 0.0824	0.3675 ± 0.0824	$\mathbf{5.6594 \pm 0.0470}$
Svgp	2%	40.7380 ± 0.2973	40.6696 ± 0.2973	114.2088 ± 0.3616
Itergp	2%	-0.2166 ± 0.0000	-0.2850 ± 0.0000	8.7699 ± 0.2120
Nystrom(uniform)	2%	0.2323 ± 0.0000	0.1639 ± 0.0000	16.4537 ± 0.0354
Nystrom(leverage)	2%	-0.1789 ± 0.0352	-0.2473 ± 0.0352	17.0823 ± 0.0296
Nystrom(ridge leverage)	2%	$\mathbf{-0.5551 \pm 0.0034}$	$\mathbf{-0.6235 \pm 0.0034}$	21.1613 ± 0.1409
Uniform	4%	0.2258 ± 0.0428	0.1574 ± 0.0428	$\mathbf{6.1125 \pm 0.0073}$
Leverage	4%	0.1558 ± 0.0174	0.0874 ± 0.0174	6.1252 ± 0.0429
Kmeans	4%	0.3369 ± 0.0516	0.2685 ± 0.0516	6.1207 ± 0.0113
Svgp	4%	41.6886 ± 0.2967	41.6202 ± 0.2967	116.2481 ± 0.1822
Itergp	4%	-0.3124 ± 0.0000	-0.3808 ± 0.0000	22.2070 ± 0.1170
Nystrom(uniform)	4%	-0.4713 ± 0.0492	-0.5397 ± 0.0492	18.6045 ± 0.0160
Nystrom(leverage)	4%	-0.3172 ± 0.0174	-0.3856 ± 0.0174	19.0833 ± 0.0375
Nystrom(ridge leverage)	4%	$\mathbf{-0.6380 \pm 0.0032}$	$\mathbf{-0.7064 \pm 0.0032}$	21.7761 ± 0.0655
Uniform	6%	0.1517 ± 0.0509	0.0833 ± 0.0509	6.4596 ± 0.0220
Leverage	6%	0.1157 ± 0.0173	0.0473 ± 0.0173	6.4745 ± 0.0210
Kmeans	6%	0.3647 ± 0.1634	0.2963 ± 0.1634	$\mathbf{6.4564 \pm 0.0371}$
Svgp	6%	40.0423 ± 0.1234	39.9739 ± 0.1234	118.1078 ± 0.2941
Itergp	6%	-0.3415 ± 0.0000	-0.4099 ± 0.0000	43.3009 ± 0.0840
Nystrom(uniform)	6%	-0.5925 ± 0.0074	-0.6609 ± 0.0074	20.4320 ± 0.0431
Nystrom(leverage)	6%	-0.4420 ± 0.0122	-0.5104 ± 0.0122	20.9021 ± 0.0344
Nystrom(ridge leverage)	6%	$\mathbf{-0.6815 \pm 0.0021}$	$\mathbf{-0.7499 \pm 0.0021}$	22.7822 ± 0.1542
Uniform	8%	0.1517 ± 0.0308	0.0833 ± 0.0308	$\mathbf{7.1314 \pm 0.0217}$
Leverage	8%	0.0987 ± 0.0140	0.0303 ± 0.0140	7.2971 ± 0.0233
Kmeans	8%	0.3060 ± 0.0824	0.2376 ± 0.0824	7.2365 ± 0.0169
Svgp	8%	38.7838 ± 0.2665	38.7154 ± 0.2665	122.1149 ± 0.1595
Itergp	8%	-0.3629 ± 0.0000	-0.4313 ± 0.0000	70.7370 ± 0.2089
Nystrom(uniform)	8%	-0.6465 ± 0.0031	-0.7149 ± 0.0031	20.5632 ± 0.0910
Nystrom(leverage)	8%	-0.5343 ± 0.0088	-0.6027 ± 0.0088	21.2114 ± 0.2373
Nystrom(ridge leverage)	8%	$\mathbf{-0.7115 \pm 0.0021}$	$\mathbf{-0.7799 \pm 0.0021}$	22.5875 ± 0.1829
Uniform	10%	0.1202 ± 0.0399	0.0518 ± 0.0399	$\mathbf{7.1129 \pm 0.0575}$
Leverage	10%	0.0541 ± 0.0274	-0.0143 ± 0.0274	7.2908 ± 0.0144
Kmeans	10%	0.2613 ± 0.0984	0.1929 ± 0.0984	7.2396 ± 0.0129
Svgp	10%	37.5595 ± 0.0773	37.4911 ± 0.0773	127.6543 ± 0.3152
Itergp	10%	-0.3798 ± 0.0000	-0.4482 ± 0.0000	104.1623 ± 0.1610
Nystrom(uniform)	10%	-0.6804 ± 0.0022	-0.7488 ± 0.0022	22.1774 ± 0.0991
Nystrom(leverage)	10%	-0.5982 ± 0.0048	-0.6666 ± 0.0048	22.7742 ± 0.0430
Nystrom(ridge leverage)	10%	$\mathbf{-0.7323 \pm 0.0024}$	$\mathbf{-0.8007 \pm 0.0024}$	23.9634 ± 0.0825

Table 3: Comparison of negative log predictive density (NLPD), mean standardized log loss (MSLL) and runtime (in seconds) for Gaussian Process Regression (GPR) methods on the UCI California Housing dataset using the Matern kernel. Each experiment is repeated over 5 random seeds, and we report the mean and standard deviation across runs. Lower values are better for all metrics.

Method	**Subset %**	**NLPD**	**MSLL**	**Time (s)**
	Full Data	0.8987	−0.6563	45.83
Uniform	2%	2.0318 ± 0.1416	0.4768 ± 0.1416	5.4882 ± 0.0558
Leverage	2%	1.8690 ± 0.0424	0.3140 ± 0.0424	5.4729 ± 0.0127
Kmeans	2%	2.0137 ± 0.1869	0.4587 ± 0.1869	$\mathbf{5.4678 \pm 0.0423}$
Svgp	2%	10.9006 ± 0.1611	9.3456 ± 0.1611	117.4531 ± 1.7228
Itergp	2%	$\mathbf{1.0109 \pm 0.0000}$	$\mathbf{-0.5442 \pm 0.0000}$	10.6045 ± 1.0680
Nystrom(uniform)	2%	1.1239 ± 0.0189	-0.4311 ± 0.0189	22.7375 ± 0.4177
Nystrom(ridge leverage)	2%	1.0696 ± 0.0072	-0.4854 ± 0.0072	24.4552 ± 0.3453
Uniform	4%	2.0952 ± 0.1466	0.5402 ± 0.1466	6.1049 ± 0.0349
Leverage	4%	1.9214 ± 0.0978	0.3663 ± 0.0978	$\mathbf{6.0918 \pm 0.0116}$
Kmeans	4%	2.0195 ± 0.1899	0.4645 ± 0.1899	6.1105 ± 0.0348
Svgp	4%	10.7327 ± 0.1574	9.1777 ± 0.1574	118.4384 ± 0.3412
Itergp	4%	0.9691 ± 0.0000	-0.5859 ± 0.0000	36.8662 ± 0.4266
Nystrom(uniform)	4%	0.9720 ± 0.0076	-0.5830 ± 0.0076	24.5613 ± 0.2247
Nystrom(ridge leverage)	4%	$\mathbf{0.9479 \pm 0.0075}$	$\mathbf{-0.6071 \pm 0.0075}$	26.3090 ± 0.1460
Uniform	6%	2.0469 ± 0.0724	0.4918 ± 0.0724	$\mathbf{6.9879 \pm 0.0088}$
Leverage	6%	1.8013 ± 0.0543	0.2462 ± 0.0543	7.1414 ± 0.0070
Kmeans	6%	1.9992 ± 0.1683	0.4441 ± 0.1683	7.1341 ± 0.0030
Svgp	6%	10.3749 ± 0.1324	8.8199 ± 0.1324	121.4746 ± 0.2119
Itergp	6%	0.9521 ± 0.0000	-0.6029 ± 0.0000	83.7735 ± 1.2646
Nystrom(uniform)	6%	0.9040 ± 0.0074	-0.6510 ± 0.0074	26.6897 ± 0.1512
Nystrom(ridge leverage)	6%	$\mathbf{0.8791 \pm 0.0059}$	$\mathbf{-0.6759 \pm 0.0059}$	28.5945 ± 0.1527
Uniform	8%	1.9365 ± 0.0553	0.3815 ± 0.0553	$\mathbf{7.0778 \pm 0.0072}$
Leverage	8%	1.7552 ± 0.0312	0.2002 ± 0.0312	7.1491 ± 0.0216
Kmeans	8%	1.9944 ± 0.1438	0.4394 ± 0.1438	7.1731 ± 0.0027
Svgp	8%	10.0673 ± 0.1195	8.5122 ± 0.1195	127.7320 ± 0.4636
Itergp	8%	0.9474 ± 0.0000	-0.6076 ± 0.0000	145.2249 ± 11.9784
Nystrom(uniform)	8%	0.8518 ± 0.0039	-0.7032 ± 0.0039	30.2436 ± 0.1127
Nystrom(ridge leverage)	8%	$\mathbf{0.8387 \pm 0.0017}$	$\mathbf{-0.7164 \pm 0.0017}$	25.1814 ± 3.6309
Uniform	12%	1.8548 ± 0.0359	0.2998 ± 0.0359	7.1411 ± 0.0161
Leverage	12%	1.6833 ± 0.0524	0.1283 ± 0.0524	$\mathbf{7.1168 \pm 0.0035}$
Kmeans	12%	1.8868 ± 0.0508	0.3318 ± 0.0508	7.1536 ± 0.0042
Svgp	12%	9.6147 ± 0.1137	8.0597 ± 0.1137	135.3741 ± 1.4913
Itergp	12%	0.9388 ± 0.0000	-0.6162 ± 0.0000	296.5839 ± 83.5047
Nystrom(uniform)	12%	0.8121 ± 0.0012	-0.7429 ± 0.0012	39.2689 ± 0.0813
Nystrom(ridge leverage)	12%	$\mathbf{0.8103 \pm 0.0009}$	$\mathbf{-0.7447 \pm 0.0009}$	30.3013 ± 0.3381

Table 4: Comparison of negative log predictive density (NLPD), mean standardized log loss (MSLL) and memory usage (in megabytes) for Gaussian Process Regression (GPR) methods on the UCI Airfoil Self-Noise dataset using the Matern kernel. Results are averaged over 5 random seeds, with standard deviations reported.

Method	**Subset %**	**NLPD**	**MSLL**	**Memory (MB)**
	Full Data	2.4025	−0.9750	51.12
Uniform	2%	3.9157 ± 0.2636	0.5383 ± 0.2636	26.6922 ± 5.0963
Leverage	2%	3.8053 ± 0.0263	0.4278 ± 0.0263	26.1291 ± 0.1068
Kmeans	2%	3.8010 ± 0.1398	0.4236 ± 0.1398	24.1977 ± 0.1063
Svgp	2%	4.7646 ± 0.0660	1.3871 ± 0.0660	19.9855 ± 2.2387
Itergp	2%	3.0549 ± 0.0000	-0.3225 ± 0.0000	$\mathbf{19.2832 \pm 0.0000}$
Nystrom(uniform)	2%	3.1717 ± 0.1845	-0.2058 ± 0.1845	20.7281 ± 0.1014
Nystrom(leverage)	2%	3.5400 ± 0.0000	0.1625 ± 0.0000	22.7260 ± 0.4168
Nystrom(ridge leverage)	2%	$\mathbf{2.9451 \pm 0.0147}$	$\mathbf{-0.4323 \pm 0.0147}$	20.9984 ± 0.4168
Uniform	4%	3.8199 ± 0.1304	0.4425 ± 0.1304	24.2454 ± 0.0131
Leverage	4%	3.6805 ± 0.0497	0.3030 ± 0.0497	26.1022 ± 0.0133
Kmeans	4%	3.6971 ± 0.0458	0.3196 ± 0.0458	24.2459 ± 0.0131
Svgp	4%	4.6540 ± 0.0517	1.2765 ± 0.0517	20.4402 ± 0.1234
Itergp	4%	2.8807 ± 0.0000	-0.4968 ± 0.0000	$\mathbf{19.2832 \pm 0.0000}$
Nystrom(uniform)	4%	2.8716 ± 0.0090	-0.5059 ± 0.0090	22.2289 ± 0.0959
Nystrom(leverage)	4%	3.0367 ± 0.0244	-0.3408 ± 0.0244	22.7094 ± 0.0959
Nystrom(ridge leverage)	4%	$\mathbf{2.8157 \pm 0.0144}$	$\mathbf{-0.5618 \pm 0.0144}$	22.2401 ± 0.0959
Uniform	6%	3.7120 ± 0.0882	0.3345 ± 0.0882	24.3877 ± 0.0225
Leverage	6%	3.6156 ± 0.0349	0.2381 ± 0.0349	26.1542 ± 0.0227
Kmeans	6%	3.7218 ± 0.0667	0.3443 ± 0.0667	24.3882 ± 0.0225
Svgp	6%	4.3329 ± 0.0323	0.9554 ± 0.0323	22.1729 ± 0.1336
Itergp	6%	2.8622 ± 0.0000	-0.5153 ± 0.0000	$\mathbf{19.2832 \pm 0.0000}$
Nystrom(uniform)	6%	2.7624 ± 0.0061	-0.6151 ± 0.0061	23.7790 ± 0.1021
Nystrom(leverage)	6%	2.8627 ± 0.0060	-0.5148 ± 0.0060	23.8132 ± 0.1021
Nystrom(ridge leverage)	6%	$\mathbf{2.7159 \pm 0.0120}$	$\mathbf{-0.6615 \pm 0.0120}$	23.7907 ± 0.1021
Uniform	8%	3.6108 ± 0.0535	0.2333 ± 0.0535	24.5618 ± 0.0307
Leverage	8%	3.5720 ± 0.0161	0.1945 ± 0.0161	26.2272 ± 0.0309
Kmeans	8%	3.6473 ± 0.1022	0.2698 ± 0.1022	24.5623 ± 0.0307
Svgp	8%	4.2478 ± 0.0368	0.8703 ± 0.0368	23.9900 ± 0.1410
Itergp	8%	2.8677 ± 0.0000	-0.5098 ± 0.0000	$\mathbf{19.2832 \pm 0.0000}$
Nystrom(uniform)	8%	2.6889 ± 0.0075	-0.6886 ± 0.0075	25.3638 ± 0.1064
Nystrom(leverage)	8%	2.7807 ± 0.0060	-0.5968 ± 0.0060	25.3979 ± 0.1064
Nystrom(ridge leverage)	8%	$\mathbf{2.6559 \pm 0.0113}$	$\mathbf{-0.7216 \pm 0.0113}$	25.3755 ± 0.1064
Uniform	10%	3.5457 ± 0.0460	0.1682 ± 0.0460	24.7744 ± 0.0400
Leverage	10%	3.5409 ± 0.0136	0.1634 ± 0.0136	26.3231 ± 0.0402
Kmeans	10%	3.5957 ± 0.0854	0.2182 ± 0.0854	24.7749 ± 0.0400
Svgp	10%	4.0887 ± 0.0631	0.7112 ± 0.0631	25.9065 ± 0.1508
Itergp	10%	2.8104 ± 0.0000	-0.5671 ± 0.0000	$\mathbf{19.2832 \pm 0.0000}$
Nystrom(uniform)	10%	2.6252 ± 0.0045	-0.7523 ± 0.0045	26.9956 ± 0.1123
Nystrom(leverage)	10%	2.7220 ± 0.0029	-0.6555 ± 0.0029	27.0298 ± 0.1123
Nystrom(ridge leverage)	10%	$\mathbf{2.6130 \pm 0.0244}$	$\mathbf{-0.7645 \pm 0.0244}$	27.0073 ± 0.1123

Table 5: Comparison of negative log predictive density (NLPD), mean standardized log loss (MSLL) and memory usage (in megabytes) for Gaussian Process Regression (GPR) methods on the UCI Elevators dataset using the Matern kernel. Results are averaged over 5 random seeds, with standard deviations reported.

Method	**Subset %**	**NLPD**	**MSLL**	**Memory (MB)**
	Full Data	-0.8485	-0.9169	**4055.47**
Uniform	3%	0.2850 ± 0.0241	0.2166 ± 0.0241	990.2746 ± 539.8686
Leverage	3%	0.2107 ± 0.0137	0.1423 ± 0.0137	721.6564 ± 2.7777
Kmeans	3%	0.3394 ± 0.0787	0.2710 ± 0.0787	721.6564 ± 2.7777
Svgp	3%	4.5856 ± 0.0548	4.5172 ± 0.0548	$\mathbf{161.6910 \pm 16.6343}$
Itergp	3%	-0.1158 ± 0.0000	-0.1842 ± 0.0000	224.2437 ± 0.0000
Nystrom(uniform)	3%	-0.3056 ± 0.0063	-0.3740 ± 0.0063	483.4021 ± 16.9635
Nystrom(ridge leverage)	3%	$\mathbf{-0.4026 \pm 0.0099}$	$\mathbf{-0.4710 \pm 0.0099}$	514.8188 ± 45.6582
Uniform	5%	0.2149 ± 0.0229	0.1465 ± 0.0229	742.7208 ± 4.5334
Leverage	5%	0.1754 ± 0.0113	0.1070 ± 0.0113	742.7237 ± 4.5334
Kmeans	5%	0.3074 ± 0.0845	0.2390 ± 0.0845	742.7237 ± 4.5334
Svgp	5%	4.5543 ± 0.0655	4.4859 ± 0.0655	272.8118 ± 11.1136
Itergp	5%	-0.1522 ± 0.0000	-0.2206 ± 0.0000	$\mathbf{224.2437 \pm 0.0000}$
Nystrom(uniform)	5%	-0.3952 ± 0.0037	-0.4636 ± 0.0037	674.9846 ± 12.0901
Nystrom(ridge leverage)	5%	$\mathbf{-0.4389 \pm 0.0057}$	$\mathbf{-0.5073 \pm 0.0057}$	675.0925 ± 12.0901
Uniform	6%	0.1786 ± 0.0247	0.1102 ± 0.0247	757.2636 ± 2.4379
Leverage	6%	0.1606 ± 0.0136	0.0922 ± 0.0136	757.2670 ± 2.4379
Kmeans	6%	0.3217 ± 0.1206	0.2533 ± 0.1206	757.2670 ± 2.4379
Svgp	6%	4.5257 ± 0.0321	4.4573 ± 0.0321	327.4325 ± 5.6037
Itergp	6%	-0.1639 ± 0.0000	-0.2323 ± 0.0000	$\mathbf{224.2437 \pm 0.0000}$
Nystrom(uniform)	6%	-0.4211 ± 0.0065	-0.4895 ± 0.0065	771.6616 ± 6.2375
Nystrom(ridge leverage)	6%	$\mathbf{-0.4577 \pm 0.0015}$	$\mathbf{-0.5261 \pm 0.0015}$	771.6937 ± 6.1928
Uniform	9%	0.1461 ± 0.0159	0.0777 ± 0.0159	745.4193 ± 5.8440
Leverage	9%	0.1261 ± 0.0153	0.0577 ± 0.0153	745.4242 ± 5.8440
Kmeans	9%	0.2655 ± 0.0595	0.1971 ± 0.0595	745.4242 ± 5.8440
Svgp	9%	4.3862 ± 0.0311	4.3178 ± 0.0311	494.8212 ± 15.9441
Itergp	9%	-0.1921 ± 0.0000	-0.2605 ± 0.0000	$\mathbf{224.2437 \pm 0.0000}$
Nystrom(uniform)	9%	-0.4750 ± 0.0044	-0.5434 ± 0.0044	1056.8524 ± 19.7632
Nystrom(ridge leverage)	9%	$\mathbf{-0.4933 \pm 0.0015}$	$\mathbf{-0.5617 \pm 0.0015}$	1057.1027 ± 19.4862
Uniform	10%	0.1282 ± 0.0165	0.0598 ± 0.0165	749.2726 ± 0.9944
Leverage	10%	0.1052 ± 0.0175	0.0368 ± 0.0175	749.2779 ± 0.9944
Kmeans	10%	0.2356 ± 0.0455	0.1672 ± 0.0455	749.2779 ± 0.9944
Svgp	10%	4.3341 ± 0.0123	4.2657 ± 0.0123	556.8342 ± 4.8722
Itergp	10%	-0.1979 ± 0.0000	-0.2663 ± 0.0000	$\mathbf{224.2437 \pm 0.0000}$
Nystrom(uniform)	10%	-0.4861 ± 0.0030	-0.5545 ± 0.0030	1169.6672 ± 7.5045
Nystrom(ridge leverage)	10%	$\mathbf{-0.5036 \pm 0.0035}$	$\mathbf{-0.5720 \pm 0.0035}$	1169.7800 ± 7.5045

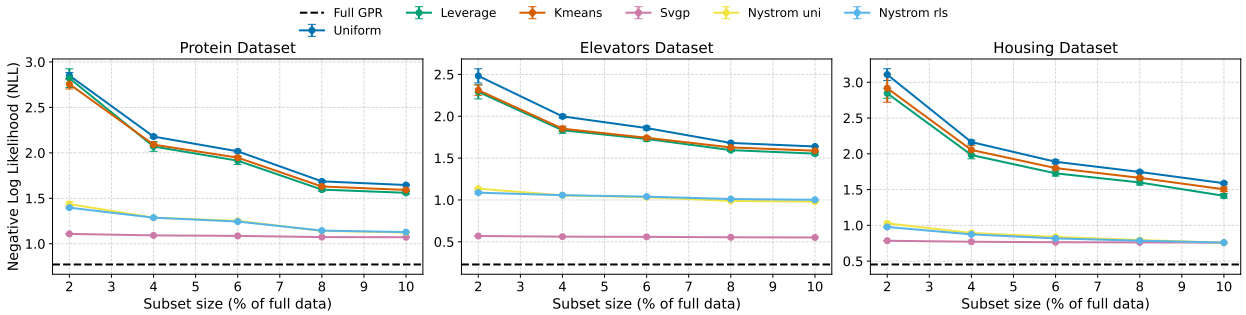

Figure 5: Comparison of Negative Log Likelihood (NLL) across different subset sizes and Gaussian Process approximation methods on the Protein, Elevators, and Housing datasets using the Matern kernel. Each subplot reports the mean and standard deviation over five random trials. The dashed horizontal line denotes the performance of the full-data (Exact GP) model.

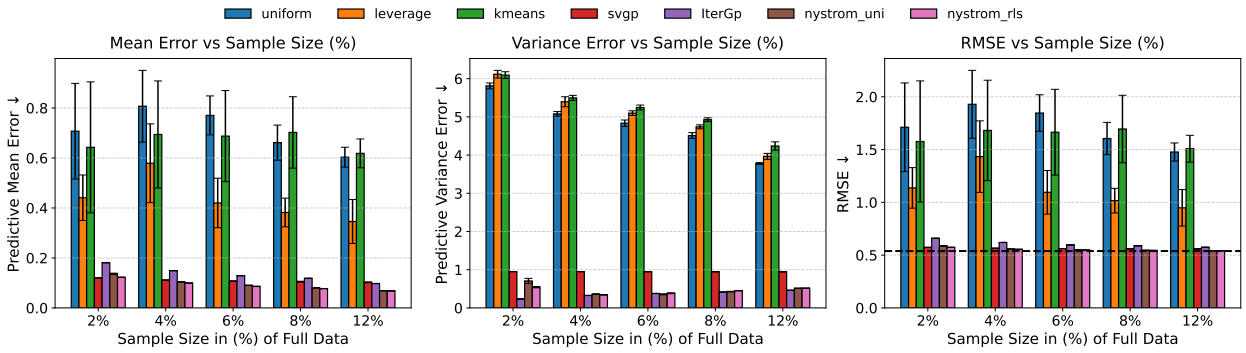

Figure 6: **Results on California Housing Dataset.** Evaluation of Gaussian Process Regression methods on the Housing dataset using the Matern kernel. Predictive mean error, predictive variance error, and RMSE are plotted versus subset size. Ridge leverage–based GPR yields the best tradeoff across metrics. All results are averaged over 5 random trials, with standard deviations shown as error bars. The dashed horizontal line indicates the performance of the full-dataset (exact GP) model.

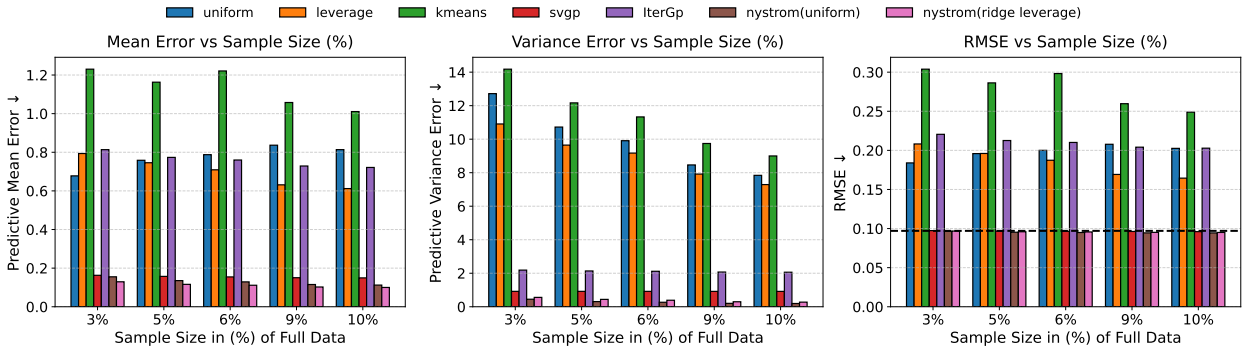

Figure 7: **Results on UCI Elevators Dataset.** Evaluation of Gaussian Process Regression methods on the UCI Elevators dataset using the Matern kernel. Predictive mean error, predictive variance error, and RMSE are plotted versus subset size. Ridge leverage–based GPR yields the best tradeoff across metrics. All results are averaged over 5 random trials, with standard deviations shown as error bars. The dashed horizontal line indicates the performance of the full-dataset (exact GP) model.

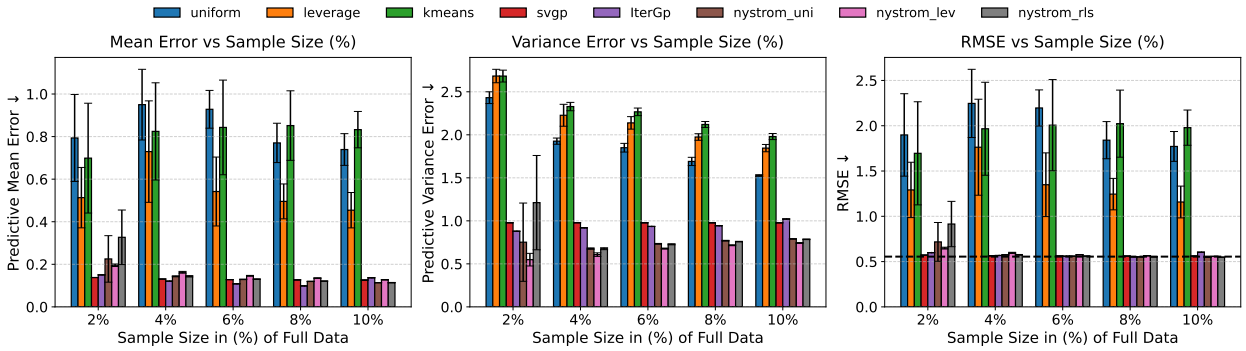

Figure 8: **Results on California Housing Dataset.** Evaluation of Gaussian Process Regression methods on the Housing dataset using the **RBF kernel**. Predictive mean error, predictive variance error, and RMSE are plotted versus subset size. Ridge Leverage based GPR yields the best tradeoff across metrics. All results are averaged over 5 random trials with standard deviation shown as error bars. The dashed horizontal line indicates the performance of the full-dataset (exact GP) model.

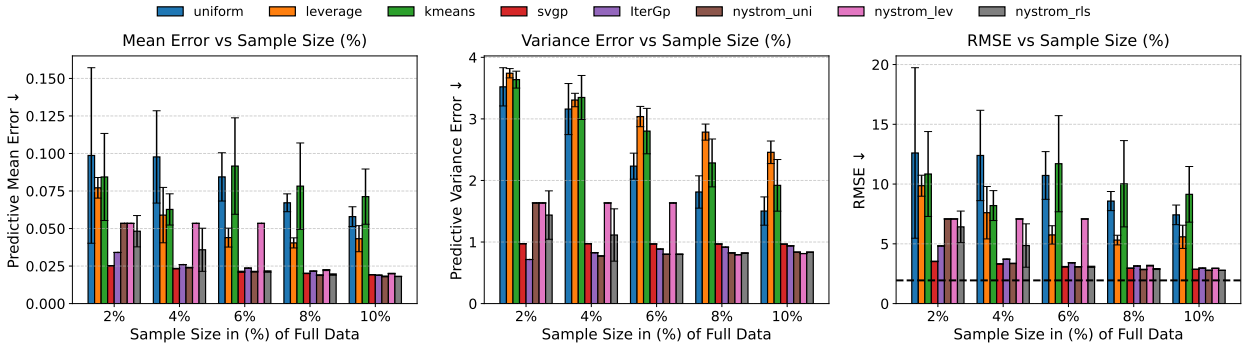

Figure 9: **Results on Airfoil Self-Noise Dataset.** Performance of various GPR methods on the Airfoil dataset using the RBF kernel. Ridge leverage−based sketching again outperforms uniform and SVGP in both predictive accuracy and variance estimation. All results are averaged over 5 random trials with standard deviation shown as error bars. The dashed horizontal line indicates the performance of the full-dataset (exact GP) model.

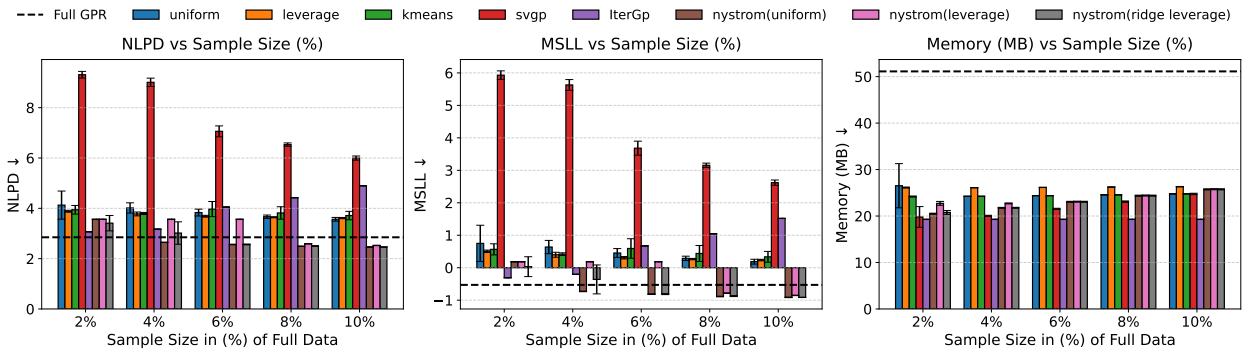

Figure 10: **Results on Airfoil Self-Noise Dataset.** Performance of various GPR methods on the *Airfoil* dataset using the RBF kernel. All results are averaged over 5 random trials, with standard deviations shown as error bars. The dashed horizontal line indicates the performance of the full-dataset (exact GP) model. Due to the relatively small dataset size, the memory usage of the proposed and baseline approximation methods remains nearly constant across subset sizes from 2% to 10%. Nevertheless, all approximation methods consume substantially less memory than the full GP, highlighting their efficiency.

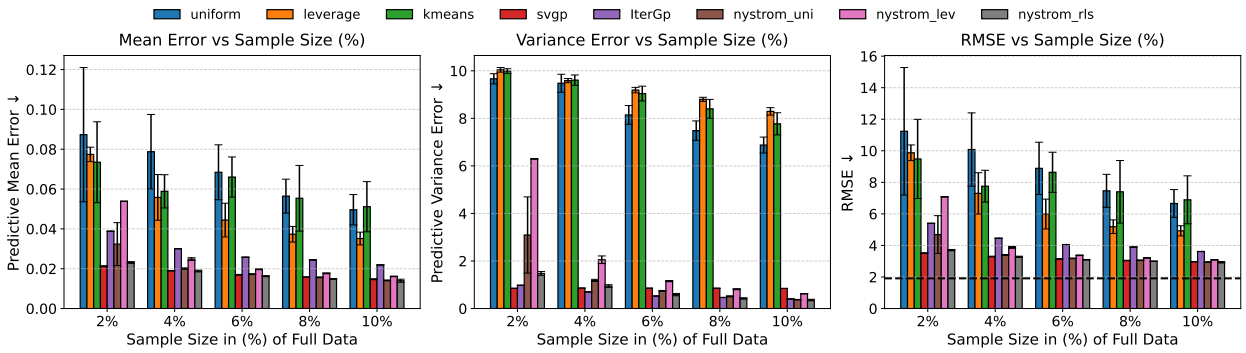

Figure 11: **Results on Airfoil Self-Noise Dataset.** Performance of various Gaussian Process Regression (GPR) methods on the UCI Airfoil Self-Noise dataset using the Matern kernel. Ridge leverage–based sketching consistently outperforms uniform sampling and SVGP in both predictive accuracy and variance estimation. All results are averaged over 5 random trials, with standard deviations shown as error bars. The dashed horizontal line indicates the performance of the full-dataset (exact GP) model.