Revisiting the Sample Complexity of Sparse Spectrum Approximation of Gaussian Processes

Quang Minh Hoang
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213
qhoang@andrew.cmu.edu

Trong Nghia Hoang
MIT-IBM Watson AI Lab
IBM Research
Cambridge, MA 02142
nghiaht@ibm.com

Hai Pham
Language Technologies Institute
Carnegie-Mellon University
Pittsburgh, PA 15213
htpham@cs.cmu.edu

David P. Woodruff
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213
dwoodruf@cs.cmu.edu

Abstract

We introduce a new scalable approximation for Gaussian processes with provable guarantees which hold simultaneously over its entire parameter space. Our approximation is obtained from an improved sample complexity analysis for sparse spectrum Gaussian processes (SSGPs). In particular, our analysis shows that under a certain data disentangling condition, an SSGP's prediction and model evidence (for training) can well-approximate those of a full GP with low sample complexity. We also develop a new auto-encoding algorithm that finds a latent space to disentangle latent input coordinates into well-separated clusters, which is amenable to our sample complexity analysis. We validate our proposed method on several benchmarks with promising results supporting our theoretical analysis.

1 Introduction

A Gaussian process (GP) [36] is a popular probabilistic kernel method for regression which has found applications across many scientific disciplines. Examples of such applications include meteorological forecasting, such as precipitation and sea-level pressure prediction [2]; sensing and monitoring of ocean and freshwater phenomena such as temperature and plankton bloom [7, 12]; traffic flow and mobility demand predictions over urban road networks [9, 10, 29]; flight delay predictions [15, 19, 20]; and persistent robotics tasks such as localization and filtering [43]. The broad applicability of GPs is in part due to their expressive Bayesian non-parametric nature which provides a closed-form prediction [36] in the form of a Gaussian distribution with formal measures of predictive uncertainty, such as entropy and mutual information criteria [27, 39, 44]. Such expressiveness makes GPs not only useful as predictive methods but also a go-to representation for active learning applications [24, 23, 27, 44] or Bayesian optimization [38, 45, 22, 16] that need to optimize for information gain while collecting training data.

Unfortunately, the expressive power of a GP comes at a cost of poor scalability (i.e., cubic time [36]) in the size of the training data (see Section 2.1 below), hence limiting its use to small datasets. This prevents GPs from being applied more broadly to modern settings with increasingly growing volumes of data [15, 19, 20]. To sidestep this limitation, a prevalent research trend is to impose sparse structural assumptions [33, 34] on the GP's kernel matrix to reduce its multiplication and inversion cost, which comprises the main bulk of the training and inference complexity. This results in a broad family of

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

sparse Gaussian processes [15, 17, 19, 28, 37, 40] that are not only computationally efficient but also amenable to various forms of parallelism [7, 29] and distributed computation [13, 20, 21, 18], further increasing their efficiency.

Despite such advantages, the sparsification components at the core of these methods are heuristically designed and do not come with provable guarantees that explicitly characterize the interplay between approximation quality and computational complexity. This motivates us to develop a more robust, theoretically-grounded approximation scheme for GPs that is both provable and amenable to the many fast computation schemes mentioned above. More specifically, our contributions include:

1. An analysis of a new approximation scheme that generates a sparse spectrum approximation of a GP with provable bounds on its sample complexity, which practically becomes significantly small when the input data exhibits a certain clustering structure. Furthermore, the impact of the approximation on the resulting training and inference qualities is also formally analyzed (Section 3.1).

2. A data partitioning algorithm inspired from the above analysis, which learns a cluster embedding that reorients the input distribution while ensuring reconstructability of the original distribution (Section 3.3). We show that using sparse spectrum Gaussian processes (SSGP) [28] on the embedded space requires fewer samples to achieve the same level of approximation quality. This also induces a linear feature map which enables efficient training and inference of GPs.

3. An empirical study on benchmarks that demonstrates the efficiency of the proposed method over existing works in terms of its approximation quality versus computational efficiency (Section 4).

2 Related Work

In this section we provide an overview of Gaussian processes (Section 2.1), followed by a succinct summary of their spectral representations (Section 2.2).

2.1 Gaussian Processes (GPs)

A Gaussian process [36] defines a probabilistic prior over a random function $g(x)$ defined by mean function $m(x) = 0$ [1] and kernel function $k(x, x')$. These functions induce a marginal Gaussian prior over the evaluations $g = [g(x_1) \cdots g(x_n)]^\top$ on an arbitrary finite subset of inputs $\{x_1, \ldots, x_n\}$. Let x_* be an unseen input whose corresponding output $g_* = g(x_*)$ we wish to predict. The Gaussian prior over $[g(x_1) \cdots g(x_n)\ g(x_*)]^\top$ implies the following conditional distribution:

$$g_* \mid g \sim \mathcal{N}\left(k_*^\top K^{-1} g,\ k(x_*, x_*) - k_*^\top K^{-1} k_*\right), \tag{1}$$

where $k_* = [k(x_*, x_1) \cdots k(x_*, x_n)]^\top$ and K denotes the Gram matrix induced by $k(x, x')$ on $\{x_1, \ldots, x_n\}$ for which $K_{ij} = k(x_i, x_j)$. For a noisy observation y perturbed by Gaussian noise such that $y \sim \mathcal{N}(g(x), \sigma^2)$, Eq. (1) above can be integrated with $\mathcal{N}(g, \sigma^2 I)$ to yield:

$$g_* \mid y \sim \mathcal{N}\left(k_*^\top (K + \sigma^2 I)^{-1} y,\ k(x_*, x_*) - k_*^\top (K + \sigma^2 I)^{-1} k_*\right). \tag{2}$$

which explicitly forms the predictive distribution of a Gaussian process. The defining parameter of $k(x, x')$ (see Section 2.2) is crucial to the predictive performance and needs to be optimized via minimizing the negative log likelihood of y:

$$\ell(\theta) = \frac{1}{2} \log \left|K_\theta + \sigma^2 I\right| + \frac{1}{2} y^\top \left(K_\theta + \sigma^2 I\right)^{-1} y, \tag{3}$$

where we now use the subscript θ to indicate that K_θ is a function of θ. In practice, both training and prediction incur $O(n^3)$ processing cost, which prevents direct use of Gaussian processes on large datasets that might contain more than tens of thousands of training inputs.

2.2 Sparse Spectrum Gaussian Processes

Sparse spectrum Gaussian processes (SSGP) [14, 17, 28] exploit Theorem 1 below to re-express the Gaussian kernel $k(x, x') = \exp(-0.5 (x - x')^\top \Sigma_\ell^{-1} (x - x'))$ (where $\Sigma_\ell = \mathrm{diag}[\ell_1^2 \cdots \ell_d^2]$) as an integration over a spectrum of cosine functions such that the integrating distribution (over the frequencies that parameterize these functions) is a multivariate Gaussian.

[1] For simplicity, we assume a zero mean function since we can always re-center the training outputs around 0.

Theorem 1 (Bochner Theorem). Let $k(x; x')$ denote a Gaussian kernel defined above and let $p(r) \propto N(0; (4\pi^2)^{-1})$. It follows that:

$$k(x; x') = E_{r \sim p(r)}\left[\cos\left(2\pi r^\top (x - x')\right)\right], \quad (4)$$

where r is a d-dimensional random variable that parameterizes $\cos(2\pi r^\top (x - x'))$. In practice, r is often referred to as the spectral frequency.

This allows us to approximate the original Gram matrix K with a low-rank matrix $K' \approx K$ constructed by a linear kernel $K'(x; x') = \phi(x)^\top \phi(x)$ with feature map $\phi(x) = [\phi_1(x) \ldots \phi_{2m}(x)]^\top$ comprising $2m$ basis trigonometric functions [17]. Each pair of odd- and even-index basis functions has $\phi_{2i-1}(x) = \cos(2\pi r_i^\top x)$ and $\phi_{2i}(x) = \sin(2\pi r_i^\top x)$ is parameterized by the same sample of spectral parameter $r_i \sim p(r)$. For efficient computation, m is often selected to be significantly smaller than n (i.e., the number of training examples). However, to guarantee that $\|K' - K\|_2 \leq \epsilon$ with probability at least $1 - \delta$, m needs to be as large as $O(n^2/\epsilon^2 \log(n/\delta))$ [31][2], which makes the total prediction complexity much worse than that of a full GP.

Alternatively, one can use kernel sketching methods [3, 32, 35] to generate feature maps that scale more favorably with the effective dimension of the kernel matrix, which empirically tends to be on the order of $O(\log n)$. However, the pitfall of these methods is that without knowing the exact parameter configuration θ that underlies the data, they cannot sample from the true $p(r)$, which is necessary in their analyses. As such, existing random maps [3, 35] that were generated based on this spectral construction often depend on a parameter initialization and their approximation quality is only guaranteed for that particular parameter setting instead of uniformly over the entire parameter space. This motivates us to revisit the sample complexity of SSGP from a setting which specifically searches for a reorientation of the input distribution such that the reoriented data exhibits a disentangled cluster structure. Such disentanglement provides a more sample-efficient bound as we show in our analysis in Section 3.1 below.

3 Provable Approximation of SSGPs with Improved Sample Complexity

We first show how a sparse spectrum Gaussian process (SSGP) [28, 31] can be approximated well with a provably low sample complexity. This is achieved by revisiting its sample complexity which, unlike prior work [3, 31, 35], explicitly characterizes and accounts for a certain set of data disentanglement conditions. Importantly, our new analysis (Section 3.1) yields practical bounds on both an SSGP's prediction and model evidence (Section 3.2) that hold with high probability uniformly over the entire parameter space[3]. Furthermore, our analysis also inspires an encoding algorithm that finds a latent space to disentangle the encoded coordinates of data into well-separated clusters on which a sparse spectrum GP can approximate a GP provably well (Section 3.3). Our experiments show that such a latent space can be found for several real-world datasets (Section 4).

3.1 Practically Improved Sample Complexity for Sparse Spectrum Gaussian Processes

This section derives a new data-oriented feature map to approximate a Gaussian process parameterized with a Gaussian kernel. Unlike existing work which assumes knowledge of the true kernel parameters [3, 32, 35], our derivation remains oblivious to such parameters, and therefore holds universally over their entire candidate space. We assume that the GP prior of interest is of the form $g(x) \sim GP(0; k(x; x'))$ where $k(x; x')$ represents its Gaussian kernel in Section 2.2.

We give our analysis in three parts: (1) the spectral sampling scheme and a notion of approximation loss; (2) a set of practical data conditions which can be either observed from a raw data distribution or approximately imposed on the data via a certain embedding; (3) a theoretical analysis that delivers our key result that establishes an improved sample complexity when our data conditions are met.

3.1.1 Spectral Sampling Scheme and Spectral Loss

We show that $g(x)$ can be approximated by $g'(x) = \sum_{i=1}^{p} g_i(x)$ with provable data-oriented guarantees where $g_i(x) \sim GP(0; (1/p)\, k_i(x; x'))$. To achieve this, we first establish in Lemma 1

[2] See Theorem 6.28 in Chapter 6 of [31].

[3] In contrast, existing literature often generates bounds on either an SSGP's prediction or its model evidence (for training) for a single parameter configuration, which makes such an analysis only heuristic.

that the induced Gram matrix K of $k(x; x^0)$ on any dataset can be represented as an expectation over a space of induced Gram matrices for $\{K_i\}_{i=1}^p$ produced by a corresponding space of random kernels $\{k_i(x; x^0)\}_{i=1}^p$.

Lemma 1. Let $k(x; x^0)$ and K denote a Gaussian kernel parameterized by Σ (Section 2.2) and its induced Gram matrix on an arbitrary set of training inputs, respectively. There exists a space of random kernels $(x; x^0)$ and a $$-independent distribution over K for which $K = E[K]$ where K denotes the induced Gram matrix of $$ on the same set of training inputs.

This follows directly from Theorem 1 above which states $k(x; x^0) = E[\cos(2 r^\top (x - x^0))]$ where $r \sim N(0; (4^2)^{-1})$. We can choose $(x; x^0;) = \cos(^\top 0.5(x - x^0))$ where $\sim N(0; I)$ which implies $k(x; x^0) = E[(x; x^0;)]$. Thus, $K = E[K]$ where the $$-independent parameter indexes and K is the induced Gram matrix of . Leveraging the result of Lemma 1, a naïve analysis [31] using worst-case concentration bounds to derive a conservative estimate for a sufficient number of samples would require a prohibitively expensive sample complexity of $O(n^2 \log n)$.

Such analyses, however, often ignore the input distribution, which can be used to sample more selectively, thereby significantly reducing the sample complexity. This is demonstrated below in Theorem 2 which shows that when the input distribution exhibits a certain degree of compactness and separation (as defined in Condition 1–3), we only require $O((\log^2 n/^2) \log\log(n/))$ sampled kernels $\{_i\}_{i=1}^p$ indexed by $\{_i\}_{i=1}^p$ to produce an average Gram matrix $K^0 = \frac{1}{p}\sum_{i=1}^p K_{_i}$ that is sufficiently close to K in spectral norm (see Definition 1) with probability at least 1 .

Definition 1 (Spectral Closeness). Given > 0, the symmetric matrices K and K^0 are $$-close if $\|K - K^0\|_2$ where $\|K - K^0\|_2 = \max(K - K^0)$ denotes the largest eigenvalue of $K - K^0$.

Thus, parameterizing the GP prior with K^0 instead of K allows us to derive an upper bound on the expected difference between their induced model evidence (for learning kernel parameters) and prediction losses (for testing) with respect to the same parameter setup (Theorem 3). Theorem 3 importantly exploits the fact that the bound in Theorem 2 holds universally over the entire space of parameters, which allows us to bound the prediction difference between the original and approximated GPs with respect to their own optimized parameters (that are not necessarily the same).

3.1.2 Practical Conditions on Data Distributions

We now outline key practical data conditions, which can be satisfied approximately via an encoding algorithm that transforms the input data into a latent space where such conditions are met. These conditions are necessary for deriving a practically improved sample complexity in Section 3.1.3.

Condition 1. For each parameter configuration $= \mathrm{diag}[^2_1; \ldots; ^2_d]$, there exists a mixture distribution $M(x; = (_1; \ldots; _b); = (_1; \ldots; _b); c = (c_1; \ldots; c_b))$ with at most $b = O(\log n)$ Gaussian components $N(x; c_i; ^2_i \ ^1)$ over the data space with the mixing weights $/ 2^{\frac{i}{2}}$ and variances $_i = O(\frac{1}{\sqrt{d}})$ that generate the observed data in d-dimensional space.

Condition 2. The i^{th} Gaussian component as defined in Condition 1 above was used to generate 2^i data points of the observed dataset. This can be substantiated easily with high probability given the above setup in Condition 1 that assigns selection probability $/ 2^{\frac{i}{2}}$ to the i^{th} -component.

Condition 3. For each parameter configuration $= \mathrm{diag}[^2_1; \ldots; ^2_d]$, the mixture distribution of data in Condition 1 has sufficiently separated cluster centers. That is, for $i \neq j$:

$$^{1=2}(c_i - c_j)^2_2 > \frac{3}{2}\log\frac{2^a}{2^a - 1} \quad \text{where} \quad a = \frac{1}{\log 2}\log\frac{n^4}{n^4 - ^4} : \qquad (5)$$

These conditions impose that the observed data can be separated into a number of clusters with exponentially growing sizes and concentration (see the small variances defined in Condition 1 and the imposed sizes of Condition 2). Intuitively, this means data points that belong to clusters with high concentration are responsible for kernel entries with high values whereas those in clusters with low concentration generate entries with low values. This is easy to see since high concentration reduces the distance between data points, thus increasing their kernel values and vice versa.

Furthermore, as imposed by Condition 2, clusters with high concentration also have denser population and induce kernel entries with high value. In addition, Condition 3 requires that clusters are well-separated, which implies that a large number of kernel entries are small and therefore can be

approximated cheaply. Together, these conditions form the foundations of our reduced complexity analysis for SSGP in Theorem 2. Interestingly, we show that such conditions also inspire the development of a probabilistic algorithm that finds an encoding of the input that (approximately) satisfies these conditions while preserving the statistical properties of the input (Section 3.3). This results in an improved sample complexity for SSGPs in practice (see Section 3.1.3).

3.1.3 Main Results

To understand the intuition why an improved sample complexity can be obtained, we note that when data is partitioned in clusters with different concentrations and sizes, the kernel entries are also partitioned into multiple value-bands with narrow width (i.e., low variance). Exploiting this, we can calibrate a significantly lower sample complexity for each band using concentration inequalities that improve with lower variance [11, 25].

Then, to combine these in-band sample complexities efficiently, we further exploit the data conditions in Section 3.1.2 to show that statistically, value bands with smaller width also tend to be populated more densely[4]. This allows us to aggregate these in-band sample costs into an overall sample complexity with low cost. In practice, this also inspires an embedding algorithm (Section 3.3) that transforms the data in such a way that the distribution of their induced kernel entries will be denser in narrower bands, which is advantageous in our analysis.

Formally, let $\mathcal{C}$ be the set of all kernel entries indexed by (u, v) in the Gram matrix K such that x_u and x_v belong to the same cluster and $\mathcal{C}'$ be its complement. Also, let $\mathcal{C}$ be partitioned[5] into b value-bands $\mathcal{C}_i = \{(u,v) \in \mathcal{C} \mid (1 - \mathcal{O}(2^{1-i}))\, K_{uv}^4 \le 1 - \mathcal{O}(2^{-i})\}$ for $i \in [1:\dots:b]$ and let $\mathcal{C}_0 = \{(u,v) \in \mathcal{C} \mid K_{uv}^4 \le 1 - \mathcal{O}(2^{-b})\}$ be a band that is only populated by very large kernel entries. Theorem 2 below shows that we can construct a spectral approximation of K with arbitrarily high probability and low sample complexity.

Theorem 2. *For any $\delta \in (0, 1 - \mathcal{O}(\exp(-b\sqrt{d}))]$, if the training data has n data points and satisfies Conditions 1–3 above with respect to δ, then with probability at least $1 - \delta/2$, the approximation $K' = (1/p)\sum_{i=1}^{p} K_{\omega_i}$ where $\omega_i \sim \mathcal{N}(0, I)$ is ε-spectral close to K.*

Proof Sketch. Our proof strategy is outlined below. The formal statements are spelled out in Appendix A.

First, with a proper choice of a clustering partition, the cross-cluster entries of K are guaranteed to be sufficiently small so as to be well-approximated by zero. We can then show with high probability that any kernel entry that corresponds to a pair of unique data points from the same cluster can be well-approximated with a sample complexity that scales favorably with the cluster's variance. In particular, we show that kernel values induced by data points generated by lower-variance clusters (see Condition 1) will have smaller approximation variances than those generated by data from higher-variance clusters and therefore require fewer samples to produce the same level of approximation.

Second, for certain configurations of mixture weights, Condition 2 asserts that the number of data points from each cluster is inversely proportional to the cluster variance, which implies that a small sample complexity is enough to approximate the majority of kernel entries. More specifically, Lemma 3 shows that when the input points are distributed into clusters with certain choices of variances $\{\sigma_i\}_{i=1}^{b}$ and at an inversely proportional ratio $\mathcal{O}(\sigma_i^{-1})$, then with high probability, over all clusters, the kernel entries (excluding those on the diagonal) associated with pairs in the i-th cluster belong to their corresponding band $\mathcal{C}_i$.

Lemma 5 shows that for $p = \mathcal{O}(\log^2 n/\varepsilon^2 \log(\log n/\varepsilon))$, with probability $1 - \delta/b$, the total approximation error of all kernel entries in the $\mathcal{C}_0$ will be at most $\varepsilon^2/4b$, which implies with probability $1 - \delta$, the total approximation cost for items in $\mathcal{C}_0$ is at most $\varepsilon^2/4$. Next, Lemma 2 establishes that with the above data distribution, $\mathcal{C}_1$ accounts for $n^2/4$ entries while $\mathcal{C}_0$ accounts for $3n^2/4$ entries, which needs to be approximated with error at most $\varepsilon^2/4$.

Finally, Lemma 4 shows that when the clusters are sufficiently well-separated (see Condition 3), any kernel value corresponding to an arbitrary data pair with points belonging to different clusters is guaranteed to be smaller than ε^2/n^2, which then guarantees a total error of at most $3\varepsilon^2/4$ when they are uniformly approximated with zero. Putting these together yields a total error of ε^2 with probability

[4] The intuition here is that kernel entries in narrower bands are cheaper (in term of sample cost) to approximate.

[5] The exact bounds defining the band can be found in Appendix A.

$1 \quad 2$, which implies K and K' are $$-spectrally close since $\|K - K'\|_2 \leq \|K - K'\|_F$. Please see Appendix A for details.

3.2 Approximation Loss for Prediction and Model Evidence

In terms of prediction and model evidence approximation, our result holds simultaneously for all parameter con gurations and is thus oblivious to the choice of parameters (see Theorem 3). While existing kernel sketch methods [3, 32] generically achieve near-linear complexity for the approximate feature map[6], they often require knowledge of the parameters to construct the kernel approximations. In contrast, our result in Theorem 2 can be leveraged to bound the same prediction discrepancy when the original and approximated GPs use their own optimized parameter con gurations, as shown in Theorem 4 below. To establish Theorem 4, however, we rst establish an intermediate result that bounds the prediction and model evidence in the case when both the original and approximated GPs use the same parameter con gurations.

Theorem 3. Let $<$ 1 be a user-speci ed con dence as de ned previously in Theorem 2 and K' let be an approximation to K for which $\|K - K'\|_2^2 \leq \sigma^2$ with probability 1, uniformly over the entire parameter space. Then, with probability 1, the following hold:

$$\mathbb{E}[g(x)] = 1 - \frac{}{\sigma^2}\,\mathbb{E}[g'(x)] \quad \text{and} \quad \mathbb{V}[g(x)] = 1 - \frac{}{\sigma^2}\,\mathbb{V}[g'(x)] - \frac{}{\sigma^2} \tag{6}$$

where σ^2 is the noise of the variance (Eq. (2)), and $g(x)$, $g'(x)$ respectively denote the predictive distributions of the full GP and the approximated GP pertaining to an arbitrary test input x.

Proof. This follows directly from Lemma 7 and Lemma 8 in Appendix B. □

Finally, Theorem 4 analyzes how close the approximated predictive mean is to the full GP predictive mean when both are evaluated at the optimizer of their respective training objective.

Theorem 4. Let $<$ 1 be a user-speci ed con dence as de ned in Theorem 2. Let K' denote an approximation to K for which $\|K - K'\|_2^2 \leq \sigma^2$ with probability at least 1, uniformly over the entire parameter space. Let and $'$ denote the optimal hyperparameters obtained by respectively minimizing the negative log likelihood of a full GP and the approximated GP. With probability 1, the following holds:

$$\mathbb{E}[g'(x\,;\,')] = 1 - (\,;\,;\,;\,')\;\mathbb{E}[g(x\,;\,)] + \}\,(\,;\,;\,;\,') \tag{7}$$

where $(\,;\,;\,;\,')$ and $\}\,(\,;\,;\,;\,')$ are constants with respect to $;\,;\,'$.

Proof. This follows immediately from Lemma 11 in Appendix B, which was built on the result of Theorem 3 above. This completes our loss analysis for SSGPs. □

3.3 Optimizing Feature Map Complexity

We next present a practical probabilistic embedding algorithm that transforms the input data to meet the requirements of Conditions 1-3. Our method is built on the rich literature of variational auto-encoders (VAE) [26], which is a broad class of deep generative models that combine the rigor of Bayesian methods and rich parameterization of (deep) neural networks to discover (non-linear) low-dimensional embeddings of data while preserving their statistical properties. We rst provide a short review on VAEs below, followed by an augmentation that aims to achieve the impositions in Conditions 1-3 above.

3.3.1 Variational Auto-Encoders (VAEs)

Let x be a random variable with density function $p(x)$. We want to learn a latent variable model $p\,(x;z) = p(z)p\,(z|x)$ that captures this generative process. The latent variable model comprises a xed latent prior $p(z)$ and a parametric likelihood $p\,(z|x)$. To learn , we maximize the variational evidence lower-bound (ELBO) $L(x;\,;\,)$ of $\log p\,(x)$:

$$L(x;\,;\,) \;,\; \mathbb{E}_{z\sim q}\left[\log p\,(x|z)\right] - \mathrm{KL}\left[q\,(z|x)\,\|\,p(z)\right] \tag{8}$$

[6] [3, 32] achieves a complexity of $\mathcal{O}(nm^2)$ where m scales with the effective dimension of the kernel matrix.

with respect to an arbitrary posterior surrogate $q(z|x) \approx p(z|x)$ over the latent variable z. The ELBO is always a lower-bound on $\log p(x)$ regardless of our choice of $q(z|x)$. This is due to the non-negativity of the KL divergence as seen in the first part of the above equation.

This can be viewed as a stochastic auto-encoder with $p(x|z)$ and $q(z|x)$ acting as the encoder and decoder, respectively. Here, θ and ϕ characterize the neural network parameterization of these models. Their learning is enabled via a re-parameterization of $q(z|x)$ that enables stochastic gradient ascent.

3.3.2 Re-configuring Data via an Augmenting Variational Auto-Encoder

To augment the above VAE framework [26, 30] to account for the impositions in Conditions 1 and 2, we ideally want to configure the parameterization of the above generative process to guarantee that the marginal posterior $q(z) = \int_x q(z|x)p(x)dx$ will manifest itself in the form of a mixture of Gaussians with the desired concentration and population densities as stated in Condition 1.

However, it is often difficult to make such an imposition directly given that we typically have no prior knowledge of $p(x)$. We instead impose the desired structure on the latent prior $p(z)$ and then penalize the divergence between $q(z)$ and $p(z)$ while optimizing for the above ELBO in Eq (8). That is, we parameterize $p(z) = \pi_1 \mathcal{N}(z; c_1, \frac{2}{1}\ ^{1}) + \ldots + \pi_b \mathcal{N}(z; c_b, \frac{2}{b}\ ^{1})$ where $\pi_i \propto 2^{i=2}$ (see Condition 2), which encodes the desired clustering structure. This is then reflected on the marginal posterior $q(z)$ via augmenting the above ELBO as,

$$\mathcal{L}(x; \theta; \phi) := \mathbb{E}_{z \sim q}\left[\log p(x|z)\right] - \mathrm{KL}\left(q(z|x)||p(z)\right) - \beta\,\mathrm{KL}\left(q(z)||p(z)\right); \quad (9)$$

where the penalty term $\mathrm{KL}(q(z)||p(z))$ serves as an incentive to encourage $q(z)$ to assume the same clustering structure as $p(z)$. The parameter β can be manually set to adjust the strength of the incentive. To encourage separation among learned clusters (see Condition 3), we also add an extra penalty term to the above augmented ELBO,

$$\mathcal{L}_{\gamma}(x; \theta; \phi) := \mathcal{L}(x; \theta; \phi) + \gamma \sum_{i \neq j} \mathrm{KL}\left(\mathcal{N}(z; c_i, \tfrac{2}{i}\ ^{1}I)\,||\,\mathcal{N}(z; c_j, \tfrac{2}{j}\ ^{1}I)\right). \quad (10)$$

Once these clusters are learned, we can use the resulting encoding network $q(z|x)$ to transform each training input x into its latent projection and subsequently train an SSGP on the latent space of z (instead of training it on the original data space). Our previous analysis can then be applied on z to give the desired sample complexity. The empirical efficiency of the proposed method is demonstrated in Section 4 below. Note that the cost of training the embedding is linear in the number of data points and therefore does not noticeably affect our overall running time.

4 Experiments

Datasets. This section presents our empirical studies on two real datasets: (a) the ABALONE dataset [42] with 3000 data points which was used to train a model that predicts the age of abalone (number of rings on its shell) from physical measurements such as length, diameter, height, whole weight, shucked weight, viscera weight and shell weight; and (b) the GAS SENSOR dataset with 4 million data points [5, 6] which was used to train a model that predicts the CO concentration (ppm) from measurements of humidity, temperature, flow rate, heater voltage and the resistant measures of 14 gas sensors.

In both settings, we compare our revised SSGP method with the traditional SSGP on both datasets to demonstrate its sample efficiency. In particular, our SSGP method is applied on the embedded space of data which was generated and configured using the auto-encoding method in Section 3.3.2 to approximately meet the aforementioned Conditions 1-3.

The detailed parameterization of our entire algorithm[7] is provided in Appendix C. The traditional SSGP method on the other hand was applied directly to the data space. The prediction root-mean-square-error (RMSE) achieved by each method is reported at different sample complexities in Figure 1 below. All reported performances were averaged over 5 independent runs on a computing server with a Tesla K40 GPU with 12GB RAM.

Results and Discussions. It can be observed from the results that at all levels of sample complexity, the revised SSGP achieves substantially better performance than its vanilla SSGP counterpart. This

[7] Our experimental code is released at https://github.com/hqminh/gp_sketch_nips .

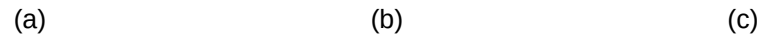


Figure 1: Graphs of performance comparisons between our revised SSGP and the traditional SSGP on the ABALONE dataset [42] at varying sample complexities (see Theorem 2): $m = 2^4, 2^5$ and 2^6.

is expected since our revised SSGP is guaranteed to require many fewer samples than the vanilla SSGP when the data is recon gured to exhibit a certain clustering structure (see Condition 1 and Theorem 2). As such, when both are set to operate at the same level of sample complexity, one would expect the revised SSGP to achieve better performance since SSGP generally performs better when its sample complexity is set closer to the required threshold. On the larger GAS SENSOR dataset (which contains approximately 4M data points), we also observe the same phenomenon from the performance comparison graph as shown in Figure 2a below: A vanilla SSGP needs to increase its number of samples to marginally improve its predictive performance while our revisited SSGP is able to outperform the former with the least number of samples ($m = 16$).

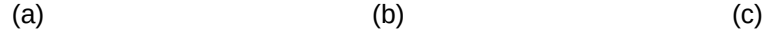


Figure 2: Graphs of (a) performance comparison between our revisited SSGP's (with sample complexity $p = 16$) and the vanilla SSGP's (with sample complexity $p = 16, 32, 64$) on the GAS SENSOR dataset [41]; and visualizations of (b) original and (c) recon gured data distributions of GAS SENSOR data on a 2-dimensional latent space generated by our auto-encoding algorithm in Section 3.3.

Furthermore, a closer look into the data distribution (visualized on a 2D space in Fig. 2b) and the data recon gured data distribution (visualized on a 2D space in Fig. 2c) also corroborates our hypothesis earlier that a well-separated data partition with high in-cluster concentration (in the form of a mixture of clusters – see Condition 1) can be found (by our embedding algorithm in Section 3.3) to recon gure our data distribution to (approximately) meet the necessary technical conditions that enable our sample-complexity enhancement analysis (see Section 3.1). Due to limited space, interested readers are referred to Appendix D for more detailed empirical studies and demonstrations.

5 Conclusion

We present a new method and analysis for approximating Gaussian processes. We obtain provable guarantees for both training and inference, which are the rst to hold simultaneously over the entire space of kernel parameters. Our results complement existing work in kernel approximation that often assumes knowledge of its de ning parameters. Our results also reveal important (practical) insights that allow us to develop an algorithmic handle on the tradeoff between approximation quality and sample complexity, which is achieved via nding an embedding that disentangles the latent coordinates of data. Our empirical results show for many datasets, such a disentangled embedding space can be found, which leads to a signi cantly reduced sample complexity of SSGP.

6 Statement of Broader Impact

Our work focuses on approximating Gaussian processes using a mixture of practical methods and theoretical analysis to recon gure data in ways that reduce their approximation complexity. As such, it could have signi cant broader impact by allowing users to more accurately solve practical problems such as the ones discussed in our introduction, while still providing concrete theoretical guarantees. While applications of our work to real data could result in ethical considerations, this is an indirect (and unpredictable) side-effect of our work. Our experimental work uses publicly available datasets to evaluate the performance of our algorithms; no ethical considerations are raised.

7 Acknowledgement

T. N. Hoang is supported by the MIT-IBM Watson AI Lab, IBM Research. Q. M. Hoang is supported by the Gordon and Betty Moore Foundation's Data-Driven Discovery Initiative through Grant GBMF4554, by the US National Science Foundation (DBI-1937540), by the US National Institutes of Health (R01GM122935), and by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program. D. Woodruff is supported by National Institute of Health grant 5R01 HG 10798-2, Of ce of Naval Research grant N00014-18-1-2562, and a Simons Investigator Award.

References

[1] Rakshit Allamraju and Girish Chowdhary. Communication ef cient decentralized gaussian process fusion for multi-uas path planning. *American Control Conference*, pages 4442–4447, 05 2017.

[2] T. J. Ansell et al. Daily mean sea level pressure reconstructions for the European-North Atlantic region for the period 1850-2003. *J. Climate*, 19(12):2717–2742, 2006.

[3] H. Avron, M. Kapralov, C. Musco, C. Musco, A. Velingker, and A. Zandieh. Random fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. *Proc. In ICML*, pages 253–262, 2017.

[4] Javier Burgues. Gas Sensor Array Temperature Modulation Dataset, howpublished = https //archive.ics.uci.edu/ml/machine-learning-databases/00487/ .

[5] Javier Burgués, Juan Manuel Jiménez-Soto, and Santiago Marco. Estimation of the limit of detection in semiconductor gas sensors through linearized calibration models. *Analytica chimica acta*, 1013:13–25, 2018.

[6] Javier Burgués and Santiago Marco. Multivariate estimation of the limit of detection by orthogonal partial least squares in temperature-modulated mox sensors. *Analytica chimica acta*, 1019:49–64, 2018.

[7] N. Cao, K. H. Low, and J. M. Dolan. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. *Proc. AAMAS*, pages 7–14, 2013.

[8] J. Chen, N. Cao, K. H. Low, R. Ouyang, C. K.-Y. Tan, and P. Jaillet. Parallel Gaussian process regression with low-rank covariance matrix approximations. *Proc. UAI*, pages 152–161, 2013.

[9] J. Chen, K. H. Low, C. K.-Y. Tan, A. Oran, P. Jaillet, J. M. Dolan, and G. S. Sukhatme. Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traf c phenomena. *Proc. UAI*, pages 163–173, 2012.

[10] Jie Chen, Kian Hsiang Low, and Colin Tan. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. *Robotics: Science and Systems*, 06 2013.

[11] H. Chernoff. A measure of asymptotic ef ciency for tests of hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[12] J. M. Dolan, G. Podnar, S. Stancliff, K. H. Low, A. Elfes, J. Higinbotham, J. C. Hosler, T. A. Moisan, and J. Moisan. Cooperative aquatic sensing using the telesupervised adaptive ocean sensor fleet. In Proc. SPIE Conference on Remote Sensing of the Ocean, Sea Ice, and Large Water Regions, volume 7473, 2009.

[13] Y. Gal, M. van der Wilk, and C. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In Proc. NIPS, 2014.

[14] Yarin Gal and Richard Turner. Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. 2015.

[15] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In Proc. UAI, pages 282–290, 2013.

[16] M. Hoang and C. Kingsford. Optimizing dynamic structures with bayesian generative search. In Internation Conference on Machine Learning, 2020.

[17] Q. M. Hoang, T. N. Hoang, and K. H. Low. A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In Proc. AAAI, pages 2007–2014, 2017.

[18] Q. M. Hoang, T. N. Hoang, K. H. Low, and C. Kingsford. Collective model fusion for multiple black-box experts. In Proc. ICML, 2019.

[19] T. N. Hoang, Q. M. Hoang, and K. H. Low. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In Proc. ICML, pages 569–578, 2015.

[20] T. N. Hoang, Q. M. Hoang, and K. H. Low. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In Proc. ICML, pages 382–391, 2016.

[21] T. N. Hoang, Q. M. Hoang, K. H. Low, and J. P. How. Collective online learning of Gaussian processes in massive multi-agent systems. In Proc. AAAI, 2019.

[22] T. N. Hoang, Q. M. Hoang, O. Ruofei, and K. H. Low. Decentralized high-dimensional bayesian optimization with factor graphs. In Proc. AAAI, 2018.

[23] T. N. Hoang, K. H. Low, P. Jaillet, and M. Kankanhalli. Nonmyopic ϵ-Bayes-optimal active learning of Gaussian processes. In Proc. ICML, pages 739–747, 2014.

[24] T. N. Hoang, K. H. Low, P. Jaillet, and M. S. Kankanhalli. Active learning is planning: Nonmyopic ϵ-Bayes-optimal active learning of Gaussian processes. In Proc. ECML-PKDD Nectar Track, pages 494–498, 2014.

[25] W. Hoeffding. Probability inequalities for the sum of bounded random variables. Journal of the American Statistical Association, 58:13–30, 1963.

[26] D. Kingma and M. Welling. Auto-Encoding Variational Bayes. In Proc. ICLR, 2013.

[27] A. Krause and C. Guestrin. Nonmyopic active learning of Gaussian processes: An exploration-exploitation approach. In Proc. ICML, pages 449–456, 2007.

[28] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse spectrum Gaussian process regression. Journal of Machine Learning Research, pages 1865–1881, 2010.

[29] K. H. Low, J. Yu, J. Chen, and P. Jaillet. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In Proc. AAAI, pages 2821–2827, 2015.

[30] Emile Mathieu, Tom Rainforth, Siddharth Narayanaswamy, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. ICML, 2019.

[31] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of Machine Learning. MIT press, 2018.

[32] C. Musco and C. Musco. Recursive sampling for the nystrom method. In *Proc. NIPS*, 2016.

[33] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

[34] J. Quiñonero-Candela, C. E. Rasmussen, and C. K. I. Williams. Approximation methods for gaussian process regression. *Large-Scale Kernel Machines*, pages 203–223, 2007.

[35] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proc. NIPS*, 2007.

[36] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[37] M. Seeger, C. K. I. Williams, and N. D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Proc. AISTATS*, 2003.

[38] J. Snoek, L. Hugo, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Proc. NIPS*, pages 2960–2968, 2012.

[39] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. ICML*, pages 1015–1022, 2010.

[40] M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proc. AISTATS*, 2009.

[41] L. J. P. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[42] Sam Waugh. Abalone Dataset, howpublished = https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/ .

[43] N. Xu, K. H. Low, J. Chen, K. K. Lim, and E. B. Özgül. GP-Localize: Persistent mobile robot localization using online sparse Gaussian process observation model. In *Proc. AAAI*, pages 2585–2592, 2014.

[44] Y. Zhang, T. N. Hoang, K. H. Low, and M. Kankanhalli. Near-optimal active learning of multi-output Gaussian processes. In *Proc. AAAI*, pages 2351–2357, 2016.

[45] Y. Zhang, T. N. Hoang, K. H. Low, and M. Kankanhalli. Information-based multi- delity bayesian optimization. In *NIPS Workshop BayesOpt*, 2017.

A Intermediate Results for Theorem 2

Let $\Delta(x_u, x_v) = |K(x_u, x_v) - K'(x_u, x_v)|$ where $K'(x_u, x_v) = (1/p)\sum_{i=1}^{p} K_i(x_u, x_v)$, and where $\omega_i \sim N(0, I)$ as defined in Lemma 1 above. We will first measure the approximation loss across different value-bands of $K(x_u, x_v)$, thereby deriving tight sample bounds for each band. Combining these with the union bound allows us to establish a much cheaper overall sample complexity as compared to the naïve $O(n^2 \log n)$ bound.

Lemma 2. Suppose the data distribution follows Condition 1-3 above. Let $c(x_u)$ denote the cluster index of each data point x_u. Let $C = \{u, v \mid c(x_u) = c(x_v)\}$ and $C' = \{u, v \mid c(x_u) \neq c(x_v)\}$ denote the sets of in-cluster and out-cluster kernel entries, respectively, where $|C| = \frac{n^2}{4}$ and $|C'| = \frac{3n^2}{4}$.

Proof. By Condition 2, since n data points are scattered across b clusters and each cluster i has $2^{i/2}$ points, it follows that:

$$n = \sum_{i=1}^{b} 2^{\frac{i}{2}} = \frac{\sqrt{2^{b+1}} - \sqrt{2}}{\sqrt{2} - 1}$$

$$\Rightarrow |C| = \sum_{i=1}^{b} 2^i = 2^{b+1} - 1 = n\left(\sqrt{2} - 1 + \sqrt{2}^{\,2} - 1\right) \simeq \frac{n^2}{4}$$

$$\Rightarrow |C'| = n^2 - |C| \simeq \frac{3n^2}{4}. \tag{11}$$

This also implies that $b = O(\log n)$, which is consistent with Condition 1 above. □

Lemma 3. Let $C_i = \{(u, v) \in C \mid c(x_u) = c(x_v) = i\}$ for $i \in [1 \ldots b]$. Then with probability at least $1 - \delta$, for $\delta = O(\exp(\log n - \sqrt{d}))$, the following holds for all i and $(u, v) \in C_i$ for which $u \neq v$:

$$1 - \frac{1}{2^{a+i-1}}^{\frac{1}{4}} \leq K(x_u, x_v) < 1 - \frac{1}{2^{a+i}}^{\frac{1}{4}} \quad \text{where} \quad a = \frac{1}{\log 2} \log \frac{n^4}{n^4 - 4} \tag{12}$$

Proof. If x_u and x_v are both generated from component i of the data distribution as defined in Condition 1, it follows that $\sqrt{1/2}(x_u - x_v) \sim N(c_i, \sigma_i^2 I)$. Therefore, by standard chi-squared tail bounds, with probability at least $1 - 2e^{-t}$, we have:

$$\left\| \sqrt{1/2}(x_u - x_v) \right\|_2^2 = \sigma_i^2 d \pm O\left(\sigma_i^2 \sqrt{dt}\right); \tag{13}$$

where d is the data dimension. Using this, we can then figure out a setting for σ_i^2 such that $K(x_u, x_v)$ follows the above condition in Eq. 12. In particular, set

$$L(i) = \log \frac{2^{a+i}}{2^{a+i-1}} \quad \text{and} \quad U(i) = \log \frac{2^{a+i-1}}{2^{a+i-1} - 1}. \tag{14}$$

We can then choose:

$$\sigma_i^2 = \frac{1}{4d}\left(U(i) + L(i)\right) \quad \text{and} \quad t = \sqrt{d}\,\frac{U(i) - L(i)}{U(i) + L(i)} = O\left(\sqrt{d}\right); \tag{15}$$

so that by plugging these choices in Eq. 13 above, we have with probability at least $1 - 2e^{-t}$:

$$\left\| \sqrt{1/2}(x_u - x_v) \right\|_2^2 \in \left[\frac{1}{2}\log \frac{2^{a+i}}{2^{a+i}-1}, \frac{1}{2}\log \frac{2^{a+i-1}}{2^{a+i-1}-1}\right]$$

$$\Rightarrow K(x_u, x_v) \in \left[1 - \frac{1}{2^{a+i-1}}^{\frac{1}{4}}, 1 - \frac{1}{2^{a+i}}^{\frac{1}{4}}\right]. \tag{16}$$

Now, note that for any i for which $\delta \geq 4^b e^{-\sqrt{d}} \geq 4^i e^{-\sqrt{d}}$ $\forall i \leq b$, we have $\delta = 4^i \cdot 2e^{-t}$ since $t = O(\sqrt{d})$. This also means $\delta = O(\exp(\log n - \sqrt{d}))$ since $b = O(\log n)$.

That is, Eq.(16) and hence, Eq.(12), hold with probability at least $1 - 2e^{-t} - 1/4^i$ for each entry in C_i. For each cluster i, even though there are up to 2^i kernel entries, by the triangle inequality it is easy to see that we only need to apply a union bound over at most 2^{i-2} (carefully selected) entries (excluding the entries on the diagonal) to meet (12) with probability at least $1 - 2^i(1/4^i) = 1 - 1/2^i$.

Subsequently, applying a union bound over all clusters gives us that with probability at least $\prod_{i=1}^{b}(1 - 1/2^i) \geq 1 - \sum_{i=1}^{b} 1/2^i \geq 1/2$, all kernel entries within the i-th cluster satisfy Eq.(12) simultaneously for $1 \leq i \leq b$. □

Lemma 4. For all $(u, v) \in C^0$, $\{u, v \mid c(x_u) \neq c(x_v)\}$, we have $K(x_u, x_v) < \left(1 - \frac{1}{2^a}\right)^{\frac{1}{4}}$ where $a = \frac{1}{\log 2} \log \frac{n^4}{n^4 - 4}$ as defined in Lemma 3 above.

Proof. For any (u, v) for which $c(x_u) = i$ and $c(x_v) = j$ and $i \neq j$, we have:

$$\|x_u - x_v\|_2^{1/2} \geq \|c_i - c_j\|_2^{1/2} - \|x_u - c_i\|_2^{1/2} - \|x_v - c_j\|_2^{1/2}$$

$$\|c_i - c_j\|_2^{1/2} \leq \frac{1}{2}\log\frac{2^{a+i}}{2^{a+i} - 1} + \frac{1}{2}\log\frac{2^{a+j}}{2^{a+j} - 1}$$

$$\|c_i - c_j\|_2^{1/2} \leq \log\frac{2^a}{2^a - 1}$$

$$\Rightarrow K(x_u, x_v) = \exp\left(-\frac{1}{2}\|x_u - x_v\|_2^{1/2}\right) \cdot \exp\left(-\frac{1}{2}\|c_i - c_j\|_2^{1/2} + \frac{1}{2}\log\frac{2^a}{2^a - 1}\right) < \left(1 - \frac{1}{2^a}\right)^{\frac{1}{4}}$$

since for all (i, j), by Condition 3:

$$\|c_i - c_j\|_2^{1/2} > \frac{3}{2}\log\frac{2^a}{2^a - 1}. \tag{17}$$

This completes our proof for the stated result of Lemma 4. □

Corollary 1. With probability at least $1 - \delta$, there are exactly n entries that are greater than $1 - 2^{-(a+b)}$ where $a = \frac{1}{\log 2} \log \frac{n^4}{n^4 - 4}$. These are the diagonal entries $K(x_u, x_u)$ with $1 \leq u \leq n$.

Proof. Lemma 3 asserts that with probability $1 - \delta$, all kernel entries $K(x_u, x_v)$, where $c(x_u) = c(x_v) = i$, belong to their respective bands $\mathcal{B}_i = \{(u, v) \mid 1 - 1/2^{a+i-1} \leq K^4(x_u, x_v) \leq 1 - 1/2^{a+i}\}$. When this happens, all in-cluster entries (except the diagonal entries) will have values between $1 - 1/2^a$ and $1 - 1/2^{a+b}$ (since there are b bands) and as such, off-cluster entries will either be smaller than $1 - 1/2^a$ or larger than $1 - 1/2^{a+b}$. But then Lemma 4 further guarantees that all off-cluster entries are smaller than $1 - 1/2^a$, following Condition 3. Thus, it follows that the only entries that are larger than $1 - 1/2^{a+b}$ are the diagonal items and there are exactly n of them. □

Lemma 5. Let $\mathcal{B}_i = \{(u, v) \mid 1 - 1/2^{a+i-1} \leq K^4(x_u, x_v) < 1 - 1/2^{a+i}\}$. It follows that for each $i \in [1 : : : b]$, with probability at least $1 - \delta/b$:

$$\sum_{(u,v)\in\mathcal{G}_i} \|\phi(x_u, x_v) - \phi'(x_u, x_v)\|_2^2 \leq \frac{\epsilon^2}{b}; \tag{18}$$

if the kernel approximation $K'(x_u, x_v) = \frac{1}{p}\sum_{t=1}^{p} K_t(x_u, x_v)$ is formed using at least $p = \frac{b|\mathcal{B}_i|}{\epsilon^2 2^{a+i}} \log \frac{b|\mathcal{B}_i|}{\delta} = O\left(\frac{\log^2 n}{\epsilon^2} \log \frac{\log n}{\delta}\right)$ samples.

Proof. For all (u,v), we have $K_t(x_u,x_v) = \cos(\omega_t^\top (x_u - x_v))$ where $\omega_t \sim N(0,I)$ and,

$$K_t(x_u,x_v) = \cos\left(\sum_{\ell=1}^{d} \omega_t^\ell \frac{x_u^\ell - x_v^\ell}{\sigma^\ell}\right), \quad \cos z_{uv}^t \quad (19)$$

Since $\omega_t^\ell \sim N(0,1)$, z_{uv}^t is then a weighted sum of Gaussian random variables and $z_{uv}^t \sim N(0, \gamma_{uv}^t)$, where $\gamma_{uv}^t = (x_u - x_v)^\top \Sigma^{-1}(x_u - x_v)$, which in turn implies:

$$E[\cos(z_{uv}^t)] = \exp(-0.5\gamma_{uv}^t) = K(x_u,x_v);$$

$$V[\cos(z_{uv}^t)] = \frac{1}{2}(1 - E[\cos(z_{uv}^t)]^2)^2 = \frac{1}{2}(1 - K^2(x_u,x_v))^2 \le \frac{1}{2^{a+i}}; \quad (20)$$

where the last inequality follows from the choice of $(u,v) \in \Omega_i$ and the definition of the Ω_i above. Next, applying the Chernoff-Hoeffding inequality and union bounding over Ω_i, we have:

$$\Pr\left[\forall (u,v) \in \Omega_i : (\Sigma^{-1})^2(x_u,x_v) \le \frac{\epsilon}{p}\right] \ge 1 - 2|\Omega_i|\exp\left(\frac{-\epsilon^2}{4\sum_{t=1}^{p} V[\cos(z_{uv}^t)]}\right)$$

$$\Rightarrow \Pr\left[\sum_{(u,v)\in\Omega_i} (\Sigma^{-1})^2(x_u,x_v) \le \frac{|\Omega_i|\epsilon}{p^2}\right] \ge 1 - 2|\Omega_i|\exp\left(\frac{-\epsilon^2 2^{a+i}}{8p}\right). \quad (21)$$

Thus, setting $\epsilon^2 = \frac{\epsilon^2 p^2}{4b|\Omega_i|}$ and $p \ge \frac{32b|\Omega_i|}{\epsilon^2 2^{a+i}} \log\left(\frac{2b|\Omega_i|}{\epsilon}\right)$ yields:

$$\Pr\left[\sum_{(u,v)\in G_i} (\Sigma^{-1})^2(x_u,x_v) \le \frac{\epsilon^2}{4b}\right] \ge 1 - 2|\Omega_i|\exp\left(\frac{-\epsilon^2 p\, 2^{a+i}}{32b|\Omega_i|}\right) \ge 1 - \frac{\delta}{b}. \quad (22)$$

where the last inequality follows from the above choice of p. Since $|\Omega_i| = 2^i$ by Condition 2, we further have $p \ge \frac{32b}{\epsilon^2 2^a} \log\left(\frac{b\, 2^{b+1}}{\delta}\right) = O\left(\frac{\log^2 n}{\epsilon^2} \log\left(\frac{\log n}{\delta}\right)\right)$. □

Lemma 5 thus establishes a very strong sample complexity bound of $O(\log^2 n \log\log n)$ for approximating all kernel entries within a narrow band of values, which is significantly cheaper than the sample complexity of $O(n^2 \log n)$ we would get if we were to ignore the distribution of kernel values in different bands. This is made clear in Corollary 2 below, which combines Lemmas 3, 4 and 5 to establish an overall sample complexity resulting in only a small approximation loss accumulated over all bands.

Corollary 2. If a kernel approximation K' of K is formed such that $K'(x_u,x_v) = \frac{1}{p}\sum_{t=1}^{p} K_t(x_u,x_v)$ for all in-cluster entries $(u,v) \in C$ using $p = O\left((\log^2 n/\epsilon^2)\log(\log n/\delta)\right)$ samples and $K'(x_{u'},x_{v'}) = 0$ for all off-cluster entries $(u',v') \in C'$, then,

$$\|K - K'\|_2^2 \le \|K - K'\|_F^2 \le \epsilon^2;$$

with probability at least $1 - \delta/2$ with $O(\exp(\log n - \sqrt{d}))$. This immediately guarantees that K' is spectrally close to K using the notion of ϵ-closeness (see Definition 1).

Proof. By Lemma 3, with probability $1 - \delta/2$, $|\Omega_i| = |G_i|$ simultaneously for all i. Thus, applying a union bound over this event and the results obtained in Lemma 5 for all clusters, we have the following bound on the total approximation loss over in-cluster entries C with probability $1 - \delta/2$:

$$\sum_{(u,v)\in C} (\Sigma^{-1})^2(x_u,x_v) \le \frac{\epsilon^2}{4}. \quad (23)$$

Furthermore, by Lemma 4, we also have the following bound on the total approximation loss over off-cluster entries in C' (which were approximated uniformly by zero):

$$\sum_{(u,v)\in C'} (\Sigma^{-1})^2(x_u,x_v) \le \frac{3n^2}{4} K(x_u,x_v) \le \frac{3n^2}{4}\sqrt{1 - \frac{1}{2^a}} \le \frac{3\epsilon^2}{4}; \quad (24)$$

when the last inequality is due to the facts (established in Lemma 4) that $K^4(x_u; x_v) \quad 1 \quad 1=2^a$ and that $a = \frac{1}{\log 2} \log \frac{n^4}{n^4 \quad ^4}$. Finally, combining these yields:

$$kK \quad K^0k_2^2 \quad k \; K \quad K^0k_F^2 = \sum_{(u;v)2C} {}^2(x_u; x_v) + \sum_{(u;v)2C^0} {}^2(x_u; x_v) \quad \frac{1}{4} \; {}^2 + \frac{3}{4} \; {}^2 = \quad {}^2 \qquad (25)$$

□

B Intermediate Results for Theorem 3

Lemma 6. Let K and K^0 be positive semide nite matrices in $R^{n \quad n}$ such that $I \quad K \quad K^0 \quad I$, $Q \; , \; K + \; ^2I$ and $Q^0 \; , \; K^0 + \; ^2I$ for some $; \; > \; 0$, then:

$$kQ^0 \; ^1k_2 = \quad 1 \quad \frac{}{2} \quad kQ \; ^1k_2 : \qquad (26)$$

Proof. By de nition of the spectral norm, we have $\forall x \; 2 \; R^n$:

$$K \quad K^0 \qquad I ; \qquad (27)$$

which implies

$$\begin{aligned} Q \qquad & K^0 + (\; ^2 + \;)I \\ & (\; ^2 + \;)K^0 + (\; ^2 + \;)I \\ = \qquad & 1 + \frac{}{2} \quad Q^0 : \end{aligned} \qquad (28)$$

where $\quad$ and $\quad$ denote the Loewner inequality operators. Likewise, by symmetry, we also have:

$$Q^0 \qquad 1 + \frac{}{2} \quad Q : \qquad (29)$$

Let $A \; , \; (1 + \; = \; ^2)Q^0$ and $B \; , \; Q$. Since A and B are symmetric and positive semide nite, there exist $U ; V$ with orthogonal rows and columns and diagonal matrices $; \; ^0$ for which $A \; = \; U \; U \; ^>$ and $B \; = \; V \quad ^0V^>$. We further let $A \quad ^{1=2} \; , \; U \quad ^{1=2}$ and $B \quad ^{1=2} \; , \; V \quad ^{0 \; 1=2}$.

Then, we can rewrite Eq. (28) as:

$$\begin{aligned} & A \quad B \qquad 0 \\ ) \quad & B \quad ^{1=2}(A \quad B)B \quad ^{1=2} \qquad 0 \\ ) \quad & B \quad ^{1=2}AB \quad ^{1=2} \quad I \qquad 0 \\ ) \quad A \quad ^{1=2}B^{1=2}(B \quad ^{1=2}AB \quad ^{1=2})B \quad ^{1=2}A^{1=2} \qquad & A \quad ^{1=2}B^{1=2}B \quad ^{1=2}A^{1=2} \\ ) \quad & A^{1=2}B \quad ^1A^{1=2} \qquad I \\ ) \quad A \quad ^{1=2}(A^{1=2}B \quad ^1A^{1=2})A \quad ^{1=2} \qquad & A \quad ^{1=2}A \quad ^{1=2} \\ ) \quad & B \quad ^1 \qquad A \quad ^1 \\ ) \quad & Q \quad ^1 \qquad \frac{^2}{^2 + }Q^{0 \; 1} \\ ) \quad 1 + \frac{}{2} \quad Q \quad ^1 \qquad & Q^{0 \; 1} : \end{aligned} \qquad (30)$$

Again, by symmetry, we can rewrite Eq. 29 as:

$$\begin{aligned} Q^{0 \; 1} \qquad & \frac{^2}{^2 + }Q \quad ^1 \qquad 1 \quad \frac{}{^2 + } \quad Q \quad ^1 \\ & 1 \quad \frac{}{2} \quad Q \quad ^1 : \end{aligned} \qquad (31)$$

Therefore, we have $kQ^0 \; ^1k_2 = (1 \quad = \; ^2)kQ \; ^1k_2$. □

Let $g(\mathbf{x})$ and $g^0(\mathbf{x})$ respectively denote the predictive distributions of full GP and the approximated GP pertaining to an arbitrary test input $\mathbf{x}$. We then state the following lemmas:

Lemma 7. *Let $\mathbf{K}^0$ denote an approximation that is ϵ-close to the original kernel $\mathbf{K}$. The induced predictive mean of $\mathbf{K}^0$ is bounded by a factor of $1 - \epsilon/\sigma^2$ times the original predictive mean.*

$$\mathrm{E}[g(\mathbf{x})] \;=\; 1 - \frac{\epsilon}{\sigma^2}\,\mathrm{E}[g^0(\mathbf{x})]\,. \tag{32}$$

Proof. Let $\mathbf{k} \triangleq [k(\mathbf{x}, \mathbf{x}_i)]_{i=1}^n$ where $\mathbf{x}_i$ denotes the i-th training data point. We have:

$$\begin{aligned}
\mathrm{E}[g(\mathbf{x})] &= \frac{1}{2}\,(\mathbf{k} + \mathbf{y})^\top \mathbf{Q}^{-1}(\mathbf{k} + \mathbf{y}) - \mathbf{k}^\top \mathbf{Q}^{-1}\mathbf{k} - \mathbf{y}^\top \mathbf{Q}^{-1}\mathbf{y} \\
&= \frac{1}{2}\,\Big(1 - \frac{\epsilon}{\sigma^2}\Big)\,(\mathbf{k} + \mathbf{y})^\top \mathbf{Q}'^{-1}(\mathbf{k} + \mathbf{y}) - \mathbf{k}^\top \mathbf{Q}'^{-1}\mathbf{k} - \mathbf{y}^\top \mathbf{Q}'^{-1}\mathbf{y} \\
&= 1 - \frac{\epsilon}{\sigma^2}\,\mathrm{E}[g^0(\mathbf{x})]\,;
\end{aligned} \tag{33}$$

where the first and third equations follow from adding and subtracting the same terms to the expression of $g(\mathbf{x})$ – see Eq. (2) – while the second equation follows from applying Lemma 6 above. □

Lemma 8. *Let $\mathbf{K}^0$ denote an approximation that is ϵ-close to the original kernel $\mathbf{K}$. The induced predictive variance of $\mathbf{K}^0$ is bounded by a factor of $1 - \epsilon/\sigma^2$ of the original predictive variance up to a constant bias of ϵ/σ^2,*

$$\mathrm{V}[g(\mathbf{x})] \;=\; 1 - \frac{\epsilon}{\sigma^2}\,\mathrm{V}[g^0(\mathbf{x})] - \frac{\epsilon}{\sigma^2}\,. \tag{34}$$

Proof. Following the definition of the Gaussian kernel, we assume that the signal of the SE (Squared Exponential) kernel is unitary [8]. As such,

$$\begin{aligned}
\mathrm{V}[g(\mathbf{x})] &= 1 - \mathbf{k}^\top \mathbf{Q}^{-1}\mathbf{k} \\
&= 1 - \Big(1 - \frac{\epsilon}{\sigma^2}\Big)\,\mathbf{k}^\top \mathbf{Q}'^{-1}\mathbf{k} \\
&= 1 - \frac{\epsilon}{\sigma^2} - \Big(1 - \frac{\epsilon}{\sigma^2}\Big)\,\mathbf{k}^\top \mathbf{Q}'^{-1}\mathbf{k} + \frac{\epsilon}{\sigma^2} \\
&= 1 - \frac{\epsilon}{\sigma^2} - \mathrm{V}[g^0(\mathbf{x})] + \frac{\epsilon}{\sigma^2}\,;
\end{aligned} \tag{35}$$

where (again) the above equation follows straightforwardly from applying Lemma 6 and standard algebraic manipulation. Lemma 7 and Lemma 8 thus provide an explicit bound on the difference between the original and approximated predictive distributions. We will now establish another bound on the difference between the original and approximated negative log likelihoods (i.e., the training objectives) in Lemma 9 and Lemma 10 below. □

Lemma 9. *Let $\mathbf{K}^0$ denote an approximation that is ϵ-close to the original kernel $\mathbf{K}$. Let $\mathbf{Q} = \mathbf{K} + \sigma^2\mathbf{I}$ and $\mathbf{Q}^0 = \mathbf{K}^0 + \sigma^2\mathbf{I}$. We have:*

$$\log|\mathbf{Q}^0| \;=\; 1 - \gamma_{\epsilon,\sigma}(\mathbf{K})\,\log|\mathbf{Q}|\,. \tag{36}$$

where the spectral constant $\gamma_{\epsilon,\sigma}(\mathbf{K})$ of $\mathbf{K}$ is defined below:

$$\gamma_{\epsilon,\sigma}(\mathbf{K}) \;\triangleq\; \frac{\max\;\log\,1 + \epsilon/\sigma^2\;;\;\log\,1 - \epsilon/\sigma^2}{\min\;\log(\lambda_{\min}(\mathbf{K}) + \sigma^2)\;;\;\log(\lambda_{\max}(\mathbf{K}) + \sigma^2)}\,. \tag{37}$$

[8] This simplifies the analysis and does not restrict the expressiveness of the kernel since we can either normalize the output or absorb it into the length-scales (i.e., the ℓ_i).

Proof. Let $_1 \quad _2 \quad \quad _n$ and $^0_1 \quad ^0_2 \quad \quad ^0_n$ be the eigenvalues of K and K^0 respectively. Applying the Courant-Fischer theorem on the result obtained in Lemma 6, we have:

$$^0_i + ^2 = 1 \quad \overline{}_2 \quad _i + ^2 : \tag{38}$$

This implies:

$$\begin{aligned}
\log jQ^0j &\quad \sum_{i=1} \log(^0_i + ^2) = \sum_{i=1} \log(_i + ^2) + \log 1 \quad \overline{}_2 \\
&\sum_{i=1} \log(_i + ^2) + \sum_{i=1} \max \log 1 + \overline{}_2 \; ; \log 1 \quad \overline{}_2 \\
&1 + _; (K) \sum_{i=1} \log(_i + ^2) = 1 + _; (K) \log jQj :
\end{aligned} \tag{39}$$

Similarly, by symmetry, we have:

$$\begin{aligned}
\log jQ^0j &\quad \sum_{i=1} \log(_i + ^2) \quad \sum_{i=1} \max \log 1 + \overline{}_2 \; ; \log 1 \quad \overline{}_2 \\
&1 \quad _; (K) \sum_{i=1} \log(_i + ^2) = 1 \quad _; (K) \log jQj :
\end{aligned} \tag{40}$$

Together, Eq. (39) and Eq. (40) imply $\log jQ^0j = 1 \quad _; (K) \log jQj$. □

Lemma 10. *Let K^0 denote an approximation that is $$-close to the original kernel K. With $_; (K)$ previously defined in Lemma 9, we have:*

$$`^0( ) = 1 \max \quad _; (K); \overline{}_2 \quad `( ) : \tag{41}$$

where $`( )$ and $`^0( )$ respectively denote the negative log likelihood of the full GP and the approximated GP evaluated at the hyper-parameters $= \text{diag}[^2_1; ^2_2 : : : ^2_d]$ as defined previously.

Proof. We have:

$$\begin{aligned}
`^0( ) &= \frac{1}{2} \log jQ^0j + \frac{1}{2} y^> Q^0 \quad ^1 y \\
&= \frac{1}{2} (1 \quad _; (K)) \log jQj + \frac{1}{2} \quad 1 \quad \overline{}_2 \quad y^> Q \quad ^1 y \\
&= 1 \max \quad _; (K); \overline{}_2 \quad \frac{1}{2} \log jQj + y^> Q \quad ^1 y \\
&= 1 \max \quad _; (K); \overline{}_2 \quad `( ) : \qquad \square
\end{aligned} \tag{42}$$

Using the result of Lemma 10 above, we can further analyze how the quality of the optimized parameter $^0 = \arg\max \; `^0( )$ of the approximated training objective compares to the true optimizer of the original objective function $= \arg\max \; `( )$ in Lemma 11 below.

Lemma 11. *Let and ^{0} denote the optimal hyper-parameters obtained by respectively minimizing the negative log likelihood of the full GP and the approximated GP. We have:*

$$`^0( ^0) = 1 \max \quad _; (K); \overline{}_2 \quad `( ) : \tag{43}$$

Proof. By Lemma 10, we have:

$$\begin{aligned}
&`^0( ^0) \quad `^0( ) \\
&\quad 1 + \max \quad _; (K); \overline{}_2 \quad `( )
\end{aligned} \tag{44}$$

and

$$\grave{}^{0}(\ ^{0}) \quad 1 \quad \max \quad _{;}(\mathsf{K}); \frac{}{2} \quad \grave{}(\ ^{0})$$

$$1 \quad \max \quad _{;}(\mathsf{K}); \frac{}{2} \quad \grave{}(\quad): \tag{45}$$

Together, these results imply $\grave{}^{0}(\ ^{0}) = 1 \quad \max \quad _{;}(\mathsf{K}); \frac{}{2} \quad \grave{}(\quad)$. □

Lemma 12. *Let* 2 $(0; 1)$ *and let* K^{0} *denote an approximation of* K *for which* $k\mathsf{K} \quad \mathsf{K}^{0}k_{2}^{2} \quad ^{2}$ *with probability at least* $1 \quad$ *uniformly over the entire parameter space. Let* $\quad$ *and* ^{0} *denote the optimal hyper-parameters obtained by respectively minimizing the negative log likelihood of the full GP and the approximated GP. Then, with probability* $1 \quad$*, the following holds:*

$$\mathrm{E}[g^{0}(x \ ; \ ^{0})] = 1 \quad (\ ; \ ; \quad ; \ ^{0}) \quad \mathrm{E}[g(x \ ; \quad)] + \}(\ ; \ ; \quad ; \ ^{0}) \tag{46}$$

where $(\ ; \ ; \quad ; \ ^{0})$ *and* $\}(\ ; \ ; \quad ; \ ^{0})$ *are constant with respect to* $\ ; \ ; \quad ; \ ^{0}$

Proof. We have:

$$\begin{aligned}\mathrm{E}[g(x \ ; \quad)] &= k^{>}Q \ ^{1}y \\ &= \frac{1}{2}\Big[(k \ + y)^{>}Q \ ^{1}(k \ + y) \quad k \ Q \ ^{1}k \ + \log jQj\Big] \quad \frac{1}{2}\grave{}(\quad) \\ &\quad \frac{1}{2}\Big[\grave{}(\quad) + 1 \sum_{i=1}^{} \log \ _{i} + \ ^{2}\Big]\end{aligned} \tag{47}$$

On the other hand, we have:

$$\begin{aligned}\mathrm{E}[g(x \ ; \quad)] &= k^{>}Q \ ^{1}y \\ &\quad \frac{1}{2}\Big[k^{>}Q \ ^{1}k \ + y^{>}Q \ ^{1}y\Big] \\ &\quad \frac{1}{2}\Big[\grave{}(\quad) + 1 \sum_{i=1}^{} \log \ _{i} + \ ^{2}\Big]\end{aligned} \tag{48}$$

Thus, we have:

$$\mathrm{E}[g(x \ ; \quad)] = \frac{1}{2}\Big[\grave{}(\quad) + 1 \sum_{i=1}^{} \log \ _{i} + \ ^{2}\Big] \tag{49}$$

and by symmetry:

$$\begin{aligned}\mathrm{E}[g^{0}(x \ ; \ ^{0})] &= \frac{1}{2}\Big[\grave{}^{0}(\ ^{0}) + 1 \sum_{i=1}^{} \log \ _{i}^{0} + \ ^{2}\Big]_{0} \\ &= 1 \quad (\ ; \ ; \quad ; \ ^{0}) \quad \mathrm{E}[g(x \ ; \quad)] + \}(\ ; \ ; \quad ; \ ^{0})\end{aligned} \tag{50}$$

where $\}(\quad ; \ ^{0})$ is a constant as defined below:

$$(\ ; \ ; \quad ; \ ^{0}) \quad , \quad \max \quad _{;}(\mathsf{K}); \ _{;}(\mathsf{K}^{0}); \frac{}{2}$$

$$\}(\ ; \ ; \quad ; \ ^{0}) \quad , \quad \Bigg[\sum_{i=1}^{} \log \frac{_{i} + \ ^{2}}{^{0}_{i} + \ ^{2}}\Bigg]_{0} \quad (\ ; \ ; \quad ; \ ^{0}) \quad 1 \sum_{i=1}^{} \log(\ _{i} + \ ^{2})$$

□

C Model Parameterization and Practical Implementation

Our embedding algorithm is based on a VAE implementation where the latent prior, posterior and likelihood of the data generation process are represented via separate mixtures of k Gaussian distributions over a 4-dimensional space. For the latent prior, we set (and fixed) the means of each Gaussian component (i.e., the prior cluster means) at k equidistant points on a 4-dimensional sphere centered at zero with an optimizable radius. For the latent posterior and likelihood, the mean and covariance entries of each component in the mixture are parameterized as outputs of their respective neural networks, which we refer to as Gaussian nets.

In turn, the Gaussian nets are parameterized separately. Each starts with a linear layer comprising of 10 neurons whose outputs are fed simultaneously to two separate hidden (linear) layers with 10 hidden neurons each. Their outputs are then used to form the mean and covariance entries of the corresponding Gaussian component. All neurons are activated by a ReLU unit, and in addition, the (batch) outputs of the first linear layer are also standardized via a learnable 1D batch-norm layer to ensure the stability of batch optimization. The mixing weights that combine such Gaussian nets in the mixtures are also parameterized as the outputs of a linear layer with $k = 8$ neurons where $k = 8$ is also the number of components in our mixture.

The above parameterized latent prior, posterior, and likelihood are then connected in the variational lower-bound (ELBO) as expressed in the first two terms of Eq. (9). This ELBO objective is then combined with two regularization terms weighted with (manually tuned) parameters $= 8.0$ and $= 1.2$ as detailed in Eq. (10). The entire function is optimized via gradient descent using the standard Adam optimizer with the default setting implemented in PyTorch.

Once learned, the outputs of the latent posterior were used as the encoded data which were fed as input to our revisited SSGP. For a practical implementation, we also found that additionally passing the encoded data to the latent likelihood generates a reconfigured version of the original data which helps to marginally improve the performance. All of our reported results below are generated with respect to this version of reconfiguration. All of our implementations of GP, SSGP and revisited SSGP that makes use of the output of this reconfiguration process, are also in PyTorch. Our experimental code is released at `https://github.com/hqminh/gp_sketch_nips`.

D Additional Empirical Results and Visualizations

This section provides additional empirical results and visualizations that complement and corroborate the reported results in the main text. In particular, we provide: (a) a more refined and comprehensive visualization of how our embedding algorithm (Section 3.3) re-configures data across different settings; and (b) an extended comparison with SSGP at different levels of sample complexity when evaluated on middle (10K data points) and large (500K data points) data sets. All data samples used in this section were extracted from the GAS SENSOR dataset [4][9].

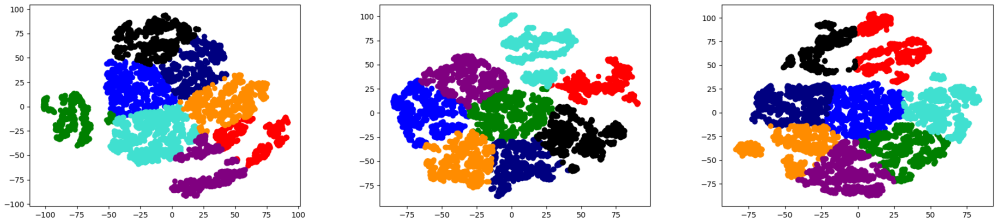
D.1 The Effect of Data Re-configuration: A Visual Demonstration

This section describes an ablation study to demonstrate the effectiveness of our data re-configuration component (i.e., to approximately meet the practical Conditions 1-3 of our refined analysis). Specifically, we demonstrate this by contrasting the scatter plots of data embeddings (see Fig. 3) before and after reconfiguration using our algorithm in Section 3.3 below. The visualizations are shown for 3 different samples of data, each of which has 10K data points.

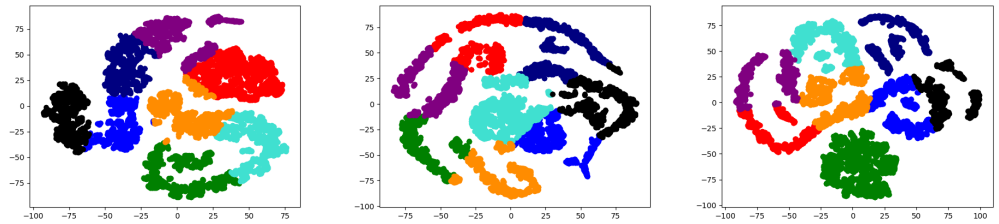
For each data sample, its embedding was clustered and re-clustered before and after its reconfiguration. Both clustering processes were generated independently using K-Means to provide an objective visual measurement of the reconfiguration effects of our algorithm.

Observing the above visual excerpts, it appears that after reconfiguration, the clusters across different data samples all became significantly more disengtangled with a visibly increased distance between their cluster centers. This provides conclusive evidence to the data disengtangling effect of our

[9] The entire GAS SENSOR dataset contains approximately 4M data points. However, in the body of this paper we only used a sample of 500K points to conduct our experiments.



(S1): Original Embedding (S2): Original Embedding (S3): Original Embedding



(S1): Reconfig. Embedding (S2): Reconfig. Embedding (S3): Reconfig. Embedding

Figure 3: Visualizations of original (top) and reconfigured (bottom) data embeddings for 3 different (randomly selected) data samples annotated with S1, S2 and S3, respectively. Each visual excerpt is annotated with different colors corresponding to the different clusters that the data belong to. All visualizations are generated using T-SNE [41].

embedding algorithm. More importantly, this demonstration further reveals a practical aspect of data that has not been investigated before in the existing literature of GP:

Data (especially experimental data) is often the manifestation of how latent concepts that underlie them were observed and depending on specific parameters of the observation process, these concepts might manifest differently in either more or less useful forms for learning. This raises the question of whether one can reorient the observation process to increase the utility of such data.

In this vein of thought, to address the above question, our data reconfiguration algorithm can be considered to be one potential solution which uses a parameterized construction of a latent space to provide a handle on how to reorient the latent concepts that underlie our data. For an intuitive example, imagine how we would look at the outside world via a narrowed pigeonhole. With different viewing angles, we would perceive the same scene outside differently and apparently, some angles provide a much better perception of that scene (thus, allowing us to interpret the scene more accurately).

In technical terms, such a reorientation is implemented in our algorithm via the regularization of the mixture composition of the latent prior while constraining the entire embedding process to have it reflected on the latent posterior – see Eq. (10) – which was used to encode data into a latent space that exhibits the desired separation effect. Such separation/disentanglement is then shown (empirically) to be richer in information and can be leveraged to improve the sample complexity of SSGP (see Section D.2), thus supporting our theoretical analysis in Appendix A.

D.2 Comparison with SSGP on Large Data

To demonstrate the effectiveness of the data disentanglement in reducing the sample complexity of SSGP, we compare the performance of SSGP and our revisited SSGP (which was instead applied on the reconfigured space of data) at different levels of sample complexity. All results were generated for two different data samples extracted from GAS-SENSOR [4]. One of these (containing 500K data points) is in fact on the same scale of the most extensive datasets used in the GP literature. All performance plots were visualized in Fig. 4 below. For each experiment, the data sample is divided into a train/test partition with an 8-2 ratio. All results were averaged over 5 independent runs.

We see that our revised SSGP consistently achieves better performance than its SSGP counterpart at all complexity levels. In particular, in all cases of the 10K setting, the performance of our revised