Abstract

Gaussian processes are the gold standard for many real-world modeling problems, especially in cases where a model’s success hinges upon its ability to faithfully represent predictive uncertainty. These problems typically exist as parts of larger frameworks, wherein quantities of interest are ultimately defined by integrating over posterior distributions. These quantities are frequently intractable, motivating the use of Monte Carlo methods. Despite substantial progress in scaling up Gaussian processes to large training sets, methods for accurately generating draws from their posterior distributions still scale cubically in the number of test locations. We identify a decomposition of Gaussian processes that naturally lends itself to scalable sampling by separating out the prior from the data. Building off of this factorization, we propose an easy-to-use and general-purpose approach for fast posterior sampling, which seamlessly pairs with sparse approximations to afford scalability both during training and at test time. In a series of experiments designed to test competing sampling schemes’ statistical properties and practical ramifications, we demonstrate how decoupled sample paths accurately represent Gaussian process posteriors at a fraction of the usual cost.

1. Introduction

Gaussian processes (GPs) are a powerful framework for reasoning about unknown functions \( f \) given partial knowledge of their behaviors. In decision-making scenarios, well-calibrated predictive uncertainty is crucial for balancing important tradeoffs, such as exploration versus exploitation and long-term versus short-term rewards. Bayesian methods naturally strike this balance (Ghavamzadeh et al., 2015; Shahriari et al., 2015). While many quantities of interest defined with respect to Bayesian posteriors cannot be computed analytically (such as expectations of nonlinear functionals), they may be readily estimated via Monte Carlo methods. Depending on this sample-based estimator’s relative cost and statistical behavior, its performance may vary from state-of-the-art to method-of-last-resort.

Unlike methods for scalable training (Hensman et al., 2013; Wang et al., 2019), techniques for efficiently sampling from GP posteriors have received relatively little attention in the machine learning literature. On the one hand, naïve approaches to sampling are statistically well-behaved, but scale poorly owing to their need to solve for increasingly large linear systems at test time. On the other hand, fast approximation strategies using Fourier features (Rahimi and Recht, 2008) avoid costly matrix operations, but are prone to misrepresenting predictive posteriors (Wang et al., 2018; Mutny and Krause, 2018; Calandriello et al., 2019). Investigating their respective behaviors, we find that many of these strategies are complementary, with one often excelling where others falter. Motivated by this comparison of strengths and weaknesses, we leverage a lesser known decomposition of GP posteriors that allows us to incorporate the best of both worlds.

Our approach centers on the observation that we may implicitly condition Gaussian random variables by combining them with an explicit corrective term. Translating this intuition to GPs, we may decompose the posterior as the sum of a prior and an update. By doing so, we are able to separately represent each of these terms using a basis well-suited for sampling. This notion of “conditioning by kriging” was first presented by Matheron in the early 1970s, with various applications to geostatistics (Journel and Huijbregts, 1978; de Fouquet, 1994; Chiles and Delfiner, 2009). The concept was later rediscovered in astrophysics (Hoffman and Ribak, 1991; Van de Weygaert and Bertschinger, 1996), where it has been used to help simulate the universe as we know it.

We unite these ideas with techniques from the growing literature on approximate GPs to obtain an easy-to-use and general-purpose approach for accurately sampling from GP posteriors in linear time.
2. Review of Gaussian processes

As notation, let \( f : \mathcal{X} \rightarrow \mathbb{R} \) be an unknown function with domain \( \mathcal{X} \subseteq \mathbb{R}^d \) whose behavior is indicated by a training set consisting of \( n \) Gaussian observations \( y_i = f(x_i) + \varepsilon_i \) subject to measurement noise \( \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \).

A Gaussian process is a random function \( f : \mathcal{X} \rightarrow \mathbb{R} \) such that, for any finite set of locations \( \mathbf{x}_* \subset \mathcal{X} \), the random vector \( \mathbf{f}_* = \mathbf{f}(\mathbf{x}_*) \) follows a Gaussian distribution. In particular, if \( f \sim \mathcal{GP}(\mu, \mathbf{K}) \), then \( \mathbf{f}_* \sim \mathcal{N}(\mathbf{\mu}_*, \mathbf{K}_{**,*}) \) is multivariate normal with covariance \( \mathbf{K}_{**,*} = \mathbf{k}(\mathbf{x}_*, \mathbf{x}_*) \) specified by a kernel \( k \). Henceforth, we assume a zero-mean prior \( \mu(\cdot) = 0 \) and continuous, stationary covariance function \( k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}') \).

Given \( n \) observations \( \mathbf{y} \), the GP posterior at \( \mathbf{x}_* \) is defined as \( \mathbf{f}_* \mid \mathbf{y} \sim \mathcal{N}(\mathbf{m}_{*|n}, \mathbf{K}_{**,*|n}) \), where we have defined

\[
\begin{align*}
\mathbf{m}_{*|n} &= \mathbf{K}_{*n}(\mathbf{K}_{nn} + \sigma^2 \mathbf{I})^{-1}\mathbf{y} \\
\mathbf{K}_{**,*|n} &= \mathbf{K}_{**,*} - \mathbf{K}_{*n}\mathbf{K}_{nn}(\mathbf{K}_{nn} + \sigma^2 \mathbf{I})^{-1}\mathbf{K}_{n*}.
\end{align*}
\]

(1)

When using (1) to help guide reinforcement learning agents (Kuss and Rasmussen, 2004), black-box optimizers (Snook et al., 2012), and other complex algorithms, we often rely on samples to estimate quantities of interest. The standard way of generating these samples is via a location-scale transform of Gaussian random variables \( \zeta \sim \mathcal{N}(0, \mathbf{I}) \), namely

\[
\mathbf{f}_* \mid \mathbf{y} = \mathbf{m}_{*|n} + \mathbf{K}^{1/2}_{**,*|n}\zeta,
\]

(2)

where \((\cdot)^{1/2}\) denotes a matrix square root, such as a Cholesky factor. Since this scheme is exact up to numerical error, we take it to be the gold standard against which the sample quality of alternatives will be judged. Unfortunately, this sampling strategy is also one of the least scalable, since the cost of computing \( \mathbf{K}^{1/2}_{**,*|n} \) is already \( \mathcal{O}(n^3) \).

The first column of Figure 1 visualizes sampling from a GP posterior given varying amounts of training data \( n \). Since matrices on the right-hand-side of (1) grow as training sets increases in size, this method of sampling can be seen to accumulate little to no error as \( n \) increases. However, this growth requires us to invert increasingly large matrices both during training and at test time, causing standard GP inference and sampling methods to scale poorly in \( n \).

2.1. Function-space approximations to GPs

The preceding interpretation of GPs, as distributions over functions with Gaussian marginals, is commonly known as the function-space view (Rasmussen and Williams, 2006). From this perspective, a natural way of approximating GPs is to represent \( f \) in terms of its behavior \( \mathbf{u} = \mathbf{f}(\mathbf{Z}) \) at a carefully chosen set of inducing locations \( \mathbf{Z} = \{ z_1, \ldots, z_m \} \). In line with this function-space intuition of reasoning about \( f \) via a small set of locations, this family of approximations is commonly referred to as sparse Gaussian processes.

Rather than directly conditioning on observations \( \mathbf{y} \), sparse GPs begin by defining \( \text{inducing distributions} \ q(\mathbf{u}) \) that explain for the data. Over the years, distinct iterations of sparse GPs have proposed different inducing paradigms (Snelson and Ghahramani, 2006; Titsias, 2009; Hensman et al., 2017). In this work, we remain agnostic regarding the choice of \( q(\mathbf{u}) \) and simply assume access to draws \( \mathbf{u} \sim q(\mathbf{u}) \).

Given \( q(\mathbf{u}) \), we approximate posterior distributions as

\[
p(\mathbf{f}_* \mid \mathbf{y}) \approx \int_{\mathbb{R}^m} p(\mathbf{f}_* \mid \mathbf{u})q(\mathbf{u}) \, d\mathbf{u}.
\]

(3)

If \( \mathbf{u} \sim \mathcal{N}(\mu_u, \Sigma_u) \), we compute this integral analytically to obtain a Gaussian distribution with mean and covariance

\[
\begin{align*}
\mathbf{m}_{*|m} &= \mathbf{K}_{*,m} \mathbf{K}_{m,m}^{-1}\mu_u \\
\mathbf{K}_{**,*|m} &= \mathbf{K}_{*,*,*} - \mathbf{K}_{*,m}\mathbf{K}_{m,m}^{-1}(\Sigma_u - \mathbf{K}_{m,m})\mathbf{K}_{m,m}^{-1}\mathbf{K}_{m,*}.
\end{align*}
\]

(4)

By virtue of explaining for \( n \) observations using \( m \) inducing variables, sparse GPs can be trained with \( \mathcal{O}(\tilde{n}m^2) \) time complexity, where the choice of batch size \( 1 \leq \tilde{n} \leq n \) depends on the particular algorithm. Since high-quality approximations can be constructed using \( m \ll n \) (Burt et al., 2019), sparse GPs drastically improve upon their exact counterparts’ \( \mathcal{O}(n^3) \) scaling.

While posterior moments (4) may be computed at reduced cost, this benefit does not carry over when sampling. The standard procedure for sampling from sparse GPs is the same as in (2) and incurs \( \mathcal{O}(n^3) \) cost.1 When used to drive Monte-Carlo-based algorithms, sparse GPs can therefore be fast during training but slow during deployment. The middle column of Figure 1 depicts samples from a sparse GP posterior with \( m = 8 \) inducing locations.

2.2. Weight-space approximations to GPs

In the function-space view of GPs, we reason about \( f \) in terms of the values it may assume at locations \( x \in \mathcal{X} \). We now turn to the weight-space view, where we will reason about \( f \) as a weighted sum of basis functions. Per the kernel trick (Schölkopf and Smola, 2001), \( k \) can be viewed as the inner product in a reproducing kernel Hilbert space (RKHS) \( \mathcal{H} \) equipped with a feature map \( \varphi : \mathcal{X} \rightarrow \mathcal{H} \). If \( \mathcal{H} \) is separable, we may approximate this inner product as

\[
k(x, x') = \langle \varphi(x), \varphi(x') \rangle_\mathcal{H} \approx \phi(x)^\top \phi(x'),
\]

(5)

where \( \phi : \mathcal{X} \rightarrow \mathbb{R}^{\ell} \) is a finite-dimensional feature map (Rasmussen and Williams, 2006). For stationary covariance functions, Bochner’s theorem implies that a suitable

1Select inducing point methods allow fast sampling from degenerate posteriors, see Quiñonero-Candela and Rasmussen (2005).
$f(x)$

$\ell$-dimensional feature map can be constructed via a set of random Fourier features (RFF) (Rahimi and Recht, 2008).

In this case, we have $\phi_i(x) = \sqrt{2/\ell} \cos(\theta_i^T x + \tau_i)$, where $\theta_i$ are sampled proportional to the kernel’s spectral density and $\tau_i \sim U(0, 2\pi)$. By defining the Bayesian linear model

$$f(\cdot) = \sum_{i=1}^{\ell} w_i \phi_i(\cdot) \quad w_i \sim \mathcal{N}(0, 1),$$

we obtain an $\ell$-dimensional GP approximation. As in previous sections, $f$ is now a random function with Gaussian marginals. However, this stochasticity is now entirely controlled by the distribution of weights $w$.

For Gaussian likelihoods, the posterior weight distribution $w \mid y \sim \mathcal{N}(\mu_{w|y}, \Sigma_{w|y})$ is Gaussian with moments

$$\mu_{w|y} = (\Phi^T \Phi + \sigma^2 I)^{-1} \Phi^T y$$

$$\Sigma_{w|y} = (\Phi^T \Phi + \sigma^2 I)^{-1},$$

where $\Phi = \phi(X)$ is an $n \times \ell$ feature matrix. In both cases, we may solve for the right-hand side at $\mathcal{O}(\min\{\ell, n\}^3)$ cost by applying the Woodbury matrix identity.

Approximating the posterior $f \mid y$ as weighted sums of basis functions in (6) is particularly advantageous for purposes of sampling. As before, we may generate draws from (7) by first computing $\Sigma_{w|y}$ at $\mathcal{O}(\ell^3)$ cost. Unlike before, we now sample weight vectors rather than function values and each draw now defines an actual function evaluable at arbitrary locations $x \in \mathcal{X}$. These methods have recently attracted attention in Bayesian optimization (Hernández-Lobato et al., 2014; Shahriari et al., 2015), where the ability to fine-tune test locations $X$, by differentiating through samples is particularly valuable (Wilson et al., 2018). Unfortunately, these efficiency gains are counterbalanced by loss in expressivity. GP approximations equipped with covariance functions arising from finite-dimensional feature maps are well-known to exhibit undesirable pathologies at test time, see Rasmussen and Quinonero-Candela (2005). In the case of Fourier-feature-based approximations, this tendency manifests as variance starvation, whereby their extrapolatory predictions become increasingly ill-behaved as $n$ increases (Wang et al., 2018; Mutny and Krause, 2018; Calandriello et al., 2019). Intuitively, this occurs because the Fourier basis is only efficient at representing stationary covariance functions with finite-dimensional feature maps. The posterior, however, is generally nonstationary. This tendency is evident in the right column of Figure 1: samples from the posterior clearly deteriorate in quality as we transition from low to high-data regimes.

**Motivation.** Prior to presenting our primary contributions, we briefly pause to restate key trends discussed above and shown in Figure 1. Sampling from sparse GPs accommodates large amounts of training data $n = |X|$, but scales poorly with the number of test locations $*$ = |X*|. Conversely, sampling from Fourier-feature-based weight-space approximations scales gracefully with $*$, but results in high approximation error as $n$ increases. Function- and weight-space approaches to sampling from GP posteriors therefore exhibit opposing strengths and weaknesses. Hence, the question: *can we obtain the best of both worlds?*
3. Sampling with Matheron’s rule

Our approach to designing an improved sampling scheme, which doubles as a rough outline for this section, is as follows: (i) analyze the shortcomings of existing methods, (ii) identify a decomposition of GPs that isolates these issues, (iii) represent the different terms using bases that address their corresponding issues. We begin by reviewing Matheron’s rule for Gaussian random variables (Journel and Huijbregts, 1978; Chiles and Delfiner, 2009; Doucet, 2010), which is central to our analysis.

**Theorem 1** (Matheron’s Rule). Let \( a \) and \( b \) be jointly Gaussian random variables. Then the random variable \( a \) conditional on \( b = \beta \) is equal in distribution to

\[
(a \mid b = \beta) \overset{d}{=} a + \text{Cov}(a, b) \text{Cov}(b, b)^{-1}(\beta - b). \tag{8}
\]

**Proof.** Follows immediately by computing the mean and variance of both sides. \( \square \)

Intuitively, Matheron’s rule tells us that conditional random variable \( a \mid b \) can be broken up into a term representing the prior \( p(a, b) \) and a term that communicates the error in the prior upon observing that \( b = \beta \). Hence, we may sample \( a \mid b \) by drawing \( a \) and \( b \) together from the prior and, subsequently, updating \( a \) to account for the residuals \( \beta - b \) as in (8). The corresponding statement for GPs is as follows.

**Corollary 2.** For a Gaussian process \( f \sim \mathcal{GP}(0, k) \) with marginal \( f_m = f(Z) \), the process conditioned on \( f_m = u \) admits, in distribution, the representation

\[
(f \mid u)(\cdot) \overset{d}{=} f(\cdot) + k(\cdot, Z)K_{m,m}^{-1}(u - f_m). \tag{9}
\]

**Proof.** By Theorem 1, the corollary holds for arbitrary finite-dimensional marginals, so the claim follows. \( \square \)

Unlike (1) and (4), Corollary 2 defines a pathwise update: rather than conditioning the prior as a distribution, we update the prior as realized in terms of sample paths. As we will soon see, this ability to go from prior to posterior (function) draws without needing to compute posterior covariance matrices (and their square-roots) will be the key to unlocking fast and accurate sampling from GP posteriors.

We are not the first to have realized this fact. This approach to simulating Gaussian conditionals is implicit in Matheron’s pioneering work in the field of geostatistics, where it was subsequently popularized by Journel and Huijbregts (1978). Decades later, (9) was rediscovered in the context of \( N \)-body simulations by Hoffman and Ribak (1991). We combine these ideas with modern machine learning methods (such as sparse GPs and random Fourier features) to create a more efficient approach to sampling.

3.1. Pathwise updates in weight- and function-spaces

Rewriting the standard formulae for sparse and exact GP posteriors, respectively, as pathwise updates in accordance with Theorem 1, we obtain

\[
f_* \mid u \overset{d}{=} f_* + K_{*,m}K_{m,m}^{-1}(u - f_m) \tag{10}
\]

\[
f_* \mid y \overset{d}{=} f_* + K_{*,n}(K_{n,n} + \sigma^2 I)^{-1}(y - f - \varepsilon). \tag{11}
\]

When sampling from sparse GPs in (10), we draw \( f_* \) and \( f_m \) together from the prior, and independently generate target values \( u \sim q(u) \). When sampling from exact GPs in (11), we again begin by jointly drawing \( f_* \) and \( f \) from the prior. Here however, we no longer need to generate targets \( u = y \). Instead, we combine \( f \) with noise variates \( \varepsilon \sim \mathcal{N}(0, \sigma^2 I) \) such that \( f + \varepsilon \) constitutes a draw from the prior distribution of \( y \).

Turning to the weight-space setting, the analogous pathwise update given an initial weight vector \( w \sim \mathcal{N}(0, I) \) is then

\[
w \mid y \overset{d}{=} w + \Phi^\top(\Phi\Phi^\top + \sigma^2 I)^{-1}(y - \Phi^\top w - \varepsilon). \tag{12}
\]

At first glance, it appears that sampling via Theorem 1 does not improve over standard methods. Whereas (12) is of modest practical interest (it allows us to sample at \( \mathcal{O}(\min\{\ell, n\}) \) cost without resorting to an eigen-decomposition), (10) and (11) are actually more expensive than their standard counterparts.

At the same time, however, Theorem 1 allows us to view GP posteriors from a different perspective. In particular, separating the effect of the prior from that of the data allows us to better diagnose the different sampling scheme’s shortcomings. For function-space approaches, we see that \( \mathcal{O}(*) \) time complexity is specific to the prior, since the update is linear in \( * \). For weight-space methods, we see that erratic extrapolations stem from difficulty representing the data (i.e., the update), since stationary priors are well-behaved under the Fourier basis. Equipped with a better understanding of why these methods fail, we now demonstrate how to address these issues.

3.2. Pathwise updates with decoupled bases

So far, we have implicitly assumed a unified view of GP posteriors: when sampling in weight-space and in function-space, we sought to generate draws from conditional distributions over weight vectors and function values, respectively. Several recent works (Cheng and Boots, 2017; Salimbeni et al., 2018; Shi et al., 2020) have introduced decompositions that separately represent different aspects of GPs via different bases, such as RKHS subspaces and their orthogonal complements. There, the authors exploit the different bases’ properties to better approximate the overarching process. We will do the same, but our goal will be to efficiently sample from the accompanying posteriors.
Corollary 2 is a pathwise update for Gaussian random variables that doubles as a decomposition of the posterior. To further build on this distinction, we restate Corollary 2 using a weight-space approximation to the prior

\[
(f | u)(\cdot) \approx \frac{1}{d} \sum_{i=1}^{d} w_i \phi_i(\cdot) + \sum_{j=1}^{m} v_j k(\cdot, z_j),
\]

where we have defined \( v = K_{m,m}^{-1} (u - \Phi^T \omega) \). The equivalent expression for exact GPs with Gaussian observations is obtained by adding noise \( \varepsilon \sim \mathcal{N}(0, \sigma^2 I) \) to \( \Phi^T w \) and replacing \( Z, u, \) and \( K_{m,m}^{-1} \) with \( X, y, \) and \( (K_{n,n} + \sigma^2 I)^{-1} \).

Figure 2 acts as a visual guide for decoupled sampling, showing the progression from prior (6) to posterior (13). Stepping through this example: (i) we draw a function \( f \) from an approximate prior, (ii) we construct an update function to account for the residuals \( u - f(Z) \) produced by an independent sample \( u \sim g(u) \), (iii) we add these functions together to obtain a function drawn from an approximate posterior (13) that we may freely evaluate anywhere in \( \mathcal{X} \).

In (13), we obtain an efficient approximation by separately discretizing the prior using Fourier basis functions \( \phi_i(\cdot) \) and the update using canonical basis functions \( k(\cdot, z_j) \). While other decompositions exist (see Appendix A), this particular decoupling directly capitalizes upon each basis’ strengths: the Fourier basis is well-suited for representing the prior (Rahimi and Recht, 2008) and the canonical basis is well-suited for representing the data (Burt et al., 2019).

By combining these bases as in (13), we therefore inherit the best of both worlds. As in weight-space methods, we may efficiently approximate draws from the prior using an \( \ell \)-dimensional Bayesian linear model \( f(\cdot) = \phi(\cdot)^T w \), where weights \( w \) are standard normal (owing to the assumed stationarity of kernel \( k \)). As in function-space methods, we may faithfully represent the data since basis functions \( k(\cdot, z_j) \) are in one-to-one correspondence with inducing locations \( z_j \in Z \). This retention of statistical propriety is evident on the right-hand side of Figure 2: despite using half as many basis functions as the weight-space method (see Figure 1), decoupled sampling’s statistical properties mirror those of the gold standard.

Expanding upon these properties, we note the following intuitive behaviors. The update function’s role of “correcting” for residuals \( u - f(Z) \) subsumes that of representing the posterior mean: replacing the prior draw \( f \) with the prior mean \( \mathbb{E}[f] \) reduces (13) to the standard expression for the conditional expectation \( \mathbb{E}[f | u] \). Since this task is performed in the canonical basis, the expected value of decoupled sample paths is guaranteed to coincide with that of (sparse) GP’s posterior. As a result, decoupled sampling becomes increasingly well-behaved as the number of training (inducing) locations grows and uncertainty decreases. Conversely, we are guaranteed to revert to the prior as we move away from the data, assuming local basis functions \( k(\cdot, z) \) (see the center column of Figure 2).

Decoupled sampling complements these desiderata with function draws’ inherent strengths. The immediate implication here is that decoupled sampling scales linearly with respect to the number of test locations \( X \). A more subtle point is that these functions are pathwise differentiable with respect to \( x \)—an affordance with significant consequences when seeking to understand Gaussian processes’ extrema.

\footnote{This point was not lost on Hoffman and Ribak (1991), who similarly approximated stationary priors using spectral methods.}
Appendix B. As a convenient shorthand, we refer to the Gaussian processes are often compared via a suitable notion of similarity on the space of probability distributions (Gibbs and Su, 2002). We focus on the 2-Wasserstein distances between an exact and, therefore, facilitate meaningful performance comparisons. Moreover, 2-Wasserstein distances between finite-dimensional approximations thereof are finite and, therefore, facilitate meaningful performance comparisons. Additionally, Wasserstein distances between an exact GP and finite-dimensional approximations thereof are finite dimensional, and therefore, facilitate meaningful performance comparisons. Moreover, 2-Wasserstein distances between finite-dimensional Gaussian marginal distributions can be computed efficiently as a proxy for distances between processes themselves. For DSGP, we may bound this distance as follows.

**Proposition 3.** Assume that \( \mathcal{X} \subseteq \mathbb{R}^d \) is compact and that stationary kernel \( k \) is sufficiently regular for \( f \sim \mathcal{G}(0, k) \) to be almost surely continuous. Let \( f \mid y \) be the posterior of \( f, f'(w), f''(s), f(d) \) that of a sparse GP, and \( f(d) \) that of a corresponding DSGP defined via an approximate prior \( f(w) \). Then we have

\[
W_{2,L^2(\mathcal{X})}(f(d), f \mid y) \leq W_{2,L^2(\mathcal{X})}(f(w), f) + C_1 W_{2,C(\mathcal{X})}(f(w), f),
\]

where \( W_{2,L^2(\mathcal{X})} \) and \( W_{2,C(\mathcal{X})} \) are the 2-Wasserstein distances over \( L^2(\mathcal{X}) \) and the space of continuous functions \( C(\mathcal{X}) \) equipped with the supremum norm, respectively.

**Proof.** Appendix B.

This bound tells us that the error exhibited by DSGP sample paths cleanly separates into independent terms associated with the sparse GP and the approximate prior. In particular, the way in which error in the prior carries over to the posterior is controlled by the inducing locations \( Z \), which \( C_1 \) depends on, but not by the inducing distribution \( q(u) \).

We continue this analysis by studying DSGP’s moments. Since a DSGP’s mean is guaranteed to coincide with that of a sparse GP, we focus on the error it introduces into the posterior covariance. When using RFF to approximate the prior, this error will depend on the \( \ell \)-dimensional basis \( \phi \) given by parameters \( \tau \sim U(0, 2\pi) \) and \( \theta \sim s(\theta) \), where \( s(\cdot) \) denotes the (normalized) spectral density of \( k \). We therefore bound the expectation of this error.

**Proposition 4.** Continuing from Proposition 3, let \( k(f(w), k(s), k(d) \) respectively denote the covariance functions of processes \( f \mid y, f(w), f'(s), f(d) \). Denoting the supremum norm over continuous functions by \( \| \cdot \|_{C(\mathcal{X})} \), it follows that

\[
\mathbb{E}_\theta \| k(d) - k(f(w)) \|_{C(\mathcal{X})^2} \leq \| k(s) - k(f(w)) \|_{C(\mathcal{X})^2} + C_2 C_3 \frac{\| u \|}{\sqrt{\ell}},
\]

where the constants \( C_2 \) and \( C_3 \) are given by Sutherland and Schneider (2015) and in Appendix B, respectively.

**Proof.** Appendix B.
We investigate decoupled sampling’s behavior in a series of experiments accompanied by two practical applications, Thompson sampling and dynamical system simulation. Each of these experiments highlights different properties of decoupled sample paths: uncertainty calibration, reliability and differentiability, and computational savings.4

**Testing uncertainty calibration with the 2-Wasserstein.**
To better understand how the bounds presented in Section 3.3 manifest in the real world, we put the various sampling schemes through numerical experiments that empirically estimated the 2-Wasserstein distance bounded by (14). These tests allow us to see how this distance is affected by factors, such as the number of training points, whose effects are difficult to directly analyze. In each trial, we measured the distance between the true posterior and empirical distributions of samples generated using the various strategies introduced in the paper. To eliminate confounding variables, experiments were run using exact GPs with known hyperparameters (see Appendix C for details).

4Code: https://github.com/j-wilson/GPflowSampling

Our investigation focuses on each method’s behavior as the number of inducing locations \( m \) (equivalently, the number of training points \( n \)) increases relative to the number of basis functions employed. For fair comparison, the total number of basis functions \( b = m + \ell \) utilized by weight-space and decoupled samplers was held equal, where \( \ell \) denotes an initial allocation. For decoupled sampling, \( \ell \) specifies the number of Fourier features used to approximate the prior.

Figure 3 shows that weight-space sampling tends to deteriorate as \( m \) increases relative to \( b \). Variance starvation causes sample paths’ extrapolatory behavior to increasingly misrepresent the posterior. This issue is exacerbated as dimensionality \( d \) rises, since we can expect the (randomly chosen) test locations \( \mathbf{X}_* \) to lie further and further away from the data.

In contrast, decoupled sampling retains its performance, and may even improve. This reflects the fact that the data is represented in an efficient basis that grows alongside it. For sparse GPs with \( m \geq n \) (which includes exact GPs as a special case), we may always represent the data exactly: usually, however, \( m \ll n \) inducing locations (i.e., kernel basis functions) suffice (Burt et al., 2019). Since we expect posterior to contract as training sets expand, the functions drawn from these posteriors behave increasingly similar to their means. Since decoupled sample paths are guaranteed to exhibit the correct means, their statistical properties may improve. This process occurs more slowly in higher-dimensional cases. However, since away from data these function draws revert to the approximate prior, they exhibit constant error when extrapolating—the approximation error of said prior.

**Thompson Sampling with robust, differentiable draws.**
Thompson Sampling (TS) is a classic strategy for decision-making in the face of uncertainty, whereby a choice \( x \in X \) is selected according to its estimated probability of being
optimal (Thompson, 1933). When used as a vehicle for
GP-based optimization, TS evaluates a pathwise minimizer
\[ x_{n+1} = \arg \min_{x \in \mathcal{X}} (f \mid y)(x) \]  
(16)
of a function drawn \( f \mid y \) from the posterior. Upon finding
this minimizer, \( x_{n+1} \) is evaluated to obtain \( y_{n+1} \), the pair
\( (x_{n+1}, y_{n+1}) \) is added to the training set, and the process
repeats. In practice, this algorithm is (embarrassingly) par-
allelized by independently drawing \( \kappa > 1 \) functions and
evaluating a minimizer of each one (Hernández-Lobato et
al., 2017; Kandasamy et al., 2018).

We compare the performance of parallel TS equipped
with the various sampling schemes discussed in Section 3,
along with two common baselines. To help eliminate con-
founding variables, experiments were run using functions
drawn from known GP priors with fixed measurement noise
\( y_t \sim \mathcal{N}(f_t, 10^{-3}) \). Across trials, we varied both the dimen-
sionality \( d \) of search spaces \( \mathcal{X} = [0, 1]^d \) and the number
of initial basis functions \( \ell \). We set \( \kappa = d \), but this choice
was not found to greatly influence results. The total number of
basis functions allocated to weight-space and decoupled samplers
was again matched, so that \( b = m + \ell \).

Figure 4 shows that different methods of sampling from
GP posteriors dramatically influence achieved performance.
While all methods suffered from the curse of dimensionality,
TS in function-space deteriorates most aggressively, owing
to its inability to efficiently exploit gradient information and
to the prohibitive cost for generating large sample vectors
\( f_s \mid y \). Weight-space TS resolves both of these issues and,
therefore, performs competitively—so long as \( b \gg m \), in
which case it accurately approximates the posterior. On the
other hand, TS in weight-space collapses due to variance
starvation as \( m \) increases relative to \( b \), often performing
worse than simpler alternatives.

Decoupled sampling avoids these shortcomings. As function
draws, decoupled sample paths \( (f \mid y)(X_s) \) boast linear time complexity \( O(n) \) and can be minimized by path-
wise differentiating with respect to \( X_s \). Moreover, because
the canonical basis is able to efficiently represent the data,
these sample paths retain their statistical properties even
when \( b \) is comparable to \( n \) or, in the case of sparse GPs,
when \( b \ll n \).

Simulating dynamical systems in linear time. Model-
based simulators are commonly used in cases where real-
world data collection proves impractical or impossible. For
example, GP surrogates are a key component of state-of-
the-art methods for solving the types of continuous control
problems seen in robotics (Deisenroth et al., 2015; Kamthe
and Deisenroth, 2018). Without loss of generality, we as-
sume that our goal is to model a time-invariant system whose
dynamics are governed by a stochastic differential equation,
discretized according to the the Euler-Maruyama integrator
\[ \Delta s_t = s_{t+1} - s_t = f(s_t, c_t) \Delta t + \sqrt{\Delta t} \varepsilon_t, \]  
(17)
where \( s_t \) denotes the state at time \( t \), \( c_t \in U \subseteq \mathbb{R}^c \) a control
input, and \( \varepsilon_t \sim \mathcal{N}(0, 1) \) a standard normal random vector.

Having trained a (sparse) GP to represent possible drift func-
tions \( f \), we simulate the system’s evolution over time by
unrolling: given a state-control pair \( (s_t, c_t) \), we sample a
transition \( \Delta s_t \) according to the GP posterior and step as
in (17). Since the resulting trajectory \( s_{1:t} \) is determined
online, standard approaches to sampling require us to itera-
tively condition on the preceding sample $f_t$ when drawing $f_{t+1} \mid f_{1:t}$. Use of caching and rank-1 downdates help limit associated costs however, the resulting algorithm’s time complexity still scales cubically in the number of steps $t$ (see Appendix C.3). By virtue of drawing functions, decoupled sampling avoids this machinery and allows us to simulate trajectories in linear time $O(t)$.

To better understand the practical ramifications of unrolling with decoupled samples, we used a sparse GP to simulate the dynamics of a well-known model of a biological neuron (FitzHugh, 1961; Nagumo et al., 1962). Results are shown in Figure 5. For both sampling schemes, simulated trajectories accurately characterizes the ways in which the system may respond to a given control signal. Their respective costs, however, vary dramatically: simulations that required 10 hours using the iterative approach, owing to cubic costs, ran in 20 seconds using decoupled sampling while achieving similar accuracy.

5. Conclusion

Decomposing Gaussian processes is a general strategy for constructing efficient approximation schemes. We have focused on a particular case, where a posterior is seen as the sum of a prior and an update, and shown how this decoupling can be exploited to efficiently draw functions from said posterior. Even within this choice of decomposition however, optimal treatment of these components will ultimately depend upon the nature of the task at hand. For example, when working with structured covariance matrices, it is sometimes possible to efficiently generate draws from the prior without introducing approximation error (Dietrich and Newsam, 1997; Wilson and Nickisch, 2015). These alternatives can then be combined with ideas discussed in previous sections to achieve the desired balance of speed versus accuracy.

Owing to the generality of our assumptions and simplicity of our proposals, decoupled sampling can be used as a plug-in extension to existing sample-based algorithms driven by (sparse) GPs. Separately representing the prior and the data with bases better suited for sampling allows us to obtain the “best of both worlds” by bringing together previous methods’ strengths. The result of this union, decoupled sampling, draws functions from GPs that may be evaluated in linear time without fear of misrepresenting their posteriors.

Acknowledgments

The authors would like to express their gratitude to Prof. Mikhail Lifshits, without whom our collaboration would never have started. This research was partially supported by “Native towns”, a social investment program of PJSC “Gazprom Neft” and by the Ministry of Science and Higher Education of the Russian Federation, agreements № 075-15-2019-1619 and № 075-15-2019-1620. The support of the EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (reference EP/L016796/1) is gratefully acknowledged.

References


