Conditioning Sparse Variational Gaussian Processes for Online Decision-making

Anonymous Author(s)
Affiliation
Address
email

Abstract

With a principled representation of uncertainty and closed form posterior updates, Gaussian processes (GPs) are a natural choice for online decision making. However, Gaussian processes typically require at least $\mathcal{O}(n^2)$ computations for n training points, limiting their general applicability. Stochastic variational Gaussian processes (SVGPs) can provide scalable inference for a dataset of fixed size, but are difficult to efficiently condition on new data. We propose online variational conditioning (OVC), a procedure for efficiently conditioning SVGPs in an online setting that does not require re-training through the evidence lower bound with the addition of new data. OVC enables the pairing of SVGPs with advanced look-ahead acquisition functions for black-box optimization, even with non-Gaussian likelihoods. We show OVC provides compelling performance in a range of applications including active learning of malaria incidence, and reinforcement learning on MuJoCo simulated robotic control tasks.

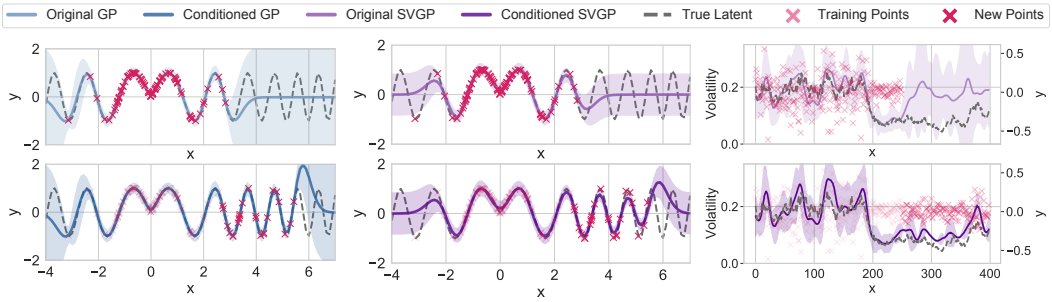
1 Introduction

Intelligent systems should be able to quickly and efficiently adapt to new data, adjusting their prior beliefs in response to the most recent events. These characteristics are desirable whether the system in question is controlling the actuators of a robot, tuning the power output of a laser, or monitoring the changing preferences of users on an online platform. What these applications share in common is a constant stream of new information. In this paper, we are interested in efficient *conditioning*, meaning that we wish to efficiently update a posterior distribution after receiving new data.

The ability of Gaussian process (GP) regression models to condition on new data in closed form has made them a popular choice for Bayesian optimization (BO), active learning, and control [23]. All of these settings share similar characteristics: there is an "outer loop", where new data is acquired from the real world (e.g. an expensive simulator), interleaved with an "inner loop", which chooses where to collect data. In BO, for example, the "inner loop" is the optimization of an acquisition function evaluated using a surrogate model of the true objective. Simple acquisition functions, e.g. expected improvement (EI), consider only the current state of the surrogate, while more sophisticated acquisition functions "look ahead" to consider the effect of hypothetical observations on future surrogate states. One such acquisition function, batch knowledge gradient (qKG), defines the one-step Bayes-optimal data batch as the batch that maximizes the expected surrogate maximum *after* the batch has been acquired [2, 78]. Advanced acquisition functions like qKG require the surrogate to have both efficient posterior sampling and efficient conditioning on new data.

GP regression has two major limitations that have prevented its large scale deployment for online decision-making. First, the compute and memory consumption of exact GPs grows at least quadratically with the amount of data [24, 60], generally limiting them to BO problems with fewer than $1,000$ function evaluations [23, 2, 73]. Second, they are limited to applications that are modeled well by a

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) Exact GP (b) SVGP + OVC (c) SVGP + OVC volatility model

Figure 1: An exact GP updates its predictive distribution in response to new data points (that is, condition on new observations) **(a,** moving from top row to bottom row). With OVC, we can condition SVGPs on both Gaussian responses **(b)** and non-Gaussian models **(c)** such as the Gaussian copula volatility model [76]. Efficient, closed form condtioning for SVGPs was previously impractical.

Gaussian likelihood. Stochastic variational Gaussian processes (SVGPs) [32] have constant compute and memory footprints and are applicable to non-Gaussian likelihoods, but they sacrifice closed form expressions for conditional posteriors. The SVGP posterior is *optimized* through the evidence lower bound (ELBO). In the online setting, training with the ELBO has two primary difficulties: the need to specify a fixed number of observations to properly scale the ELBO gradient [8] and the need to adjust the inducing points without looking at past data [3, 9]. Thus, we are presented with a choice between the simplicity and analytic tractability of exact GPs and the scalability and flexibility of SVGPs.

In this work, we develop **O**nline **V**ariational **C**onditioning (OVC) to allow SVGPs to be conditioned on-the-fly, as shown in Figure 1. In the top row of each subplot, we fit the data points shown in red, shifting to another batch of data points in the bottom row. We use an exact GP in Figure 1a, with exact conditioning shown in the bottom panel. The SVGP emulates the exact GP very well before seeing the new data and again after conditioning on the new data by using OVC (Figure 1b). In Figure 1c, we consider a non-Gaussian data model (a Gaussian copula volatility model [76]), where we cannot use exact GPs; the SVGP is still able to update its posterior over the latent volatility in response to new data without "forgetting" old observations. OVC is inspired by a new, simple rederivation of streaming sparse GPs (O-SGPR), originally proposed by Bui et al. [9]. OVC makes SVGPs truly compelling models for online decision-making, augmenting their existing strengths with efficient, closed-form conditioning on new data points. In short, our contributions are:

- The development of OVC, a novel method to condition SVGPs on new data without re-optimizing the variational posterior through an evidence lower bound.
- An improved conceptual understanding of O-SGPR [9] as SGPR trained on pseudo-data under a block-diagonal Gaussian likelihood, yielding significant practical improvements, including principled inducing location updates, an extension to non-Gaussian likelihoods, and improved numerical stability.
- Enabling the effective application of SVGPs, through OVC, in look-ahead acquisitions in BO for black-box optimization, controlling dynamical systems, and active learning.

Please see Appendix A for discussion of the limitations and broader impacts of our work.

2 Preliminaries and Related Work

In this section, we first review exact inference with Gaussian processes before reviewing variational sparse Gaussian processes, introducing sparse Gaussian process regression (SGPR), sparse variational Gaussian processes (SVGP), and streaming sparse GPs (O-SGPR). Please see Appendix B for further background and Appendix B.1 specifically for further related work.

2.1 Gaussian Processes and Exact Inference

For a full introduction to GPs, see Rasmussen and Williams [60]. Given inputs $\mathbf{x} \in \mathcal{X}$, the observed response, $y \in \mathbb{R}$, is given by $y \sim \mathcal{N}(f, \sigma^2)$ where $f \sim \mathcal{GP}(0, k_\theta(\mathbf{x}, \mathbf{x}'))$ is a Gaussian process with by the kernel function, $k_\theta(\mathbf{x}, \mathbf{x}')$, with hyperparameters θ. Given $X_{\text{train}} = [\mathbf{x}_i, \ldots, \mathbf{x}_n]^\top$ and $X_{\text{test}} = [\mathbf{x}'_1, \ldots, \mathbf{x}'_k]^\top$, we define $\mathbf{v} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$, $\mathbf{w} = [f(\mathbf{x}'_1), \ldots, f(\mathbf{x}'_k)]^\top$, so $\text{Cov}[\mathbf{v}, \mathbf{w}] = k_\theta(X_{\text{train}}, X_{\text{test}}) = K_{\mathbf{vw}}$. The posterior distribution over $\mathbf{w}$, conditioned on $\mathcal{D} := (X_{\text{train}}, \mathbf{y})$, is given in closed form by the standard expression for conditional Gaussians. That is, $p(\mathbf{w}|X_{\text{test}}, \mathcal{D}, \theta) = \mathcal{N}(\mu^*_{\mathbf{w}|\mathcal{D}}, \Sigma^*_{\mathbf{w}|\mathcal{D}})$, where $\mu^*_{\mathbf{w}|\mathcal{D}} = K_{\mathbf{wv}}(K_{\mathbf{vv}} + \sigma^2 I)^{-1}\mathbf{y}$ and $\Sigma^*_{\mathbf{w}|\mathcal{D}} = K_{\mathbf{ww}} - K_{\mathbf{wv}}(K_{\mathbf{vv}} + \Sigma_{\mathbf{y}})^{-1}K_{\mathbf{vw}}$ where $\Sigma_{\mathbf{y}} = \sigma^2 I$. Computations with $K_{\mathbf{vv}}$ cost $\mathcal{O}(n^2)$ space and $\mathcal{O}(n^3)$ computation. When repeatedly querying predictive distributions, it is more efficient to cache (these will be written in orange) the terms dependent on the training data alone, as $a := (K_{\mathbf{vv}} + \Sigma_{\mathbf{y}})^{-1}\mathbf{y}$ (the predictive mean cache) and $RR^\top := (K_{\mathbf{vv}} + \Sigma_{\mathbf{y}})^{-1}$ (the predictive covariance cache). Caching simplifies the predictive: $\mu^*_{\mathbf{w}|\mathcal{D}} = K_{\mathbf{wv}}a$ and $\Sigma^*_{\mathbf{w}|\mathcal{D}} = K_{\mathbf{ww}} - K_{\mathbf{wv}}RR^\top K_{\mathbf{vw}}$. On adding new data to the GP (i.e. conditioning), is equivalent to adding new rows and columns to $K_{\mathbf{vv}}$, enabling efficient conditioning via low rank updates to the predictive caches [54, 24, 59, 38].

2.2 Variationally Sparse Gaussian Processes

Variational sparse GPs reduce the computational burden of GP inference by using sparse approximations of the kernel matrix. For further reference, see Matthews [47] and Van der Wilk [71]. These methods define a variational distribution over the inducing point *values*, $\phi(\mathbf{u}) = \mathcal{N}(\mathbf{m}_{\mathbf{u}}, S_{\mathbf{u}})$, where $\mathbf{u} = [f(\mathbf{z}_1), \ldots, f(\mathbf{z}_p)]^\top$ with $Z = [\mathbf{z}_1, \cdots, \mathbf{z}_p] \subset \mathcal{X}$ denoting the *locations* of p inducing points, and $\mathbf{m}_{\mathbf{u}}, S_{\mathbf{u}}$ are the variational mean and covariance. They assume the latent function values $f(\mathbf{x}), f(\mathbf{x}')$ are conditionally independent given $\mathbf{u}$ and $\mathbf{x}, \mathbf{x}' \notin Z$, so as to cheaply approximate the posterior $q(f) := \int p(f|\mathbf{u})\phi(\mathbf{u})d\mathbf{u} \approx p(f|X_{\text{test}}, \mathcal{D}, \theta)$.

In seminal work, Titsias [67] proposed sparse GP regression (SGPR), by analytically optimizing an evidence lower bound (ELBO) (a lower bound of the true GP marginal log-likelihood) w.r.t. $\mathbf{m}_{\mathbf{u}}$ and $S_{\mathbf{u}}$, resulting in a new ELBO with no explicit dependence on $\mathbf{m}_{\mathbf{u}}$ and $S_{\mathbf{u}}$. The predictive distribution $q(\mathbf{w}) = \mathcal{N}(\mu^*_{\text{SGPR}}, \Sigma^*_{\text{SGPR}})$ relies on two terms dependent on the training data,

$$\mathbf{c} = K_{\mathbf{uv}}\Sigma_{\mathbf{y}}^{-1}\mathbf{y}, \qquad C = K_{\mathbf{uv}}\Sigma_{\mathbf{y}}^{-1}K_{\mathbf{vu}}. \tag{1}$$

Informally, Eq. (1) can be viewed as a compression of the real data $(X_{\text{train}}, \mathbf{y})$ into pseudo-data $(Z, \mathbf{c})$ with a pseudo-likelihood $\mathcal{N}(\mathbf{u}, C)$. The optimal $m_{\mathbf{u}}, S_{\mathbf{u}}$, and resulting $q(\mathbf{w})$ are

$$\mathbf{m}_{\mathbf{u}} = K_{\mathbf{uu}}(K_{\mathbf{uu}} + C)^{-1}\mathbf{c}, \qquad S_{\mathbf{u}} = K_{\mathbf{uu}}(K_{\mathbf{uu}} + C)^{-1}K_{\mathbf{uu}} \tag{2}$$

$$\mu^*_{\text{SGPR}} := K_{\mathbf{wu}}K_{\mathbf{uu}}^{-1}\mathbf{m}_{\mathbf{u}}, \qquad \Sigma^*_{\text{SGPR}} := K_{\mathbf{ww}} - K_{\mathbf{wu}}(K_{\mathbf{uu}}^{-1} - S_{\mathbf{u}})K_{\mathbf{uw}}, \tag{3}$$

where $\Sigma_{\mathbf{y}}$ is the covariance of the likelihood $p(\mathbf{y}|\mathbf{f})$ ($\sigma^2 I_n$ in the homoscedastic case). The predictive mean and covariance caches are given by $a = K_{\mathbf{uu}}^{-1}\mathbf{m}_{\mathbf{u}}$ and $RR^\top = K_{\mathbf{uu}}^{-1} - S_{\mathbf{u}}$, reducing the complexity of inference from $\mathcal{O}(n^3)$ to $\mathcal{O}(np^2)$, which is a significant improvement if $p \ll n$.

Hensman et al. [32, 33] developed the stochastic variational GP (SVGP) to remedy the limitations of SGPR (linear dependence on n and Gaussian likelihoods). They directly optimize $\mathbf{m}_{\mathbf{u}}$ and $S_{\mathbf{u}}$ alongside Z and θ by optimizing an "uncollapsed" ELBO which decomposes additively across the training data, allowing the use of stochastic optimizers. The time per-batch is now $\mathcal{O}(bp^2 + p^3)$, where b is the minibatch size, removing dependence on n. The uncollapsed ELBO is numerically unstable, so we can use the "whitened" parameterization, $\bar{\mathbf{m}}_{\mathbf{u}} = K_{\mathbf{uu}}^{-1/2}\mathbf{m}_{\mathbf{u}}$ and $\bar{S}_{\mathbf{u}} = K_{\mathbf{uu}}^{-1/2}S_{\mathbf{u}}K_{\mathbf{uu}}^{-1/2}$ [47].

We emphasize the distinction between constant-time minibatch gradients, and constant-time conditioning. Given an SVGP already trained on some existing data, conditioning jointly on both the old and new data requires storing all the data and making multiple gradient updates to the variational parameters. As the size of the dataset grows, so does the number of gradient steps needed. In contrast by constant-time conditioning we mean a procedure that takes a posterior conditioned on existing data and produces a new posterior conditioned jointly on the old and new data with a fixed amount of compute and memory, regardless of the number of past observations. One example of constant-time conditioning is found in Bui et al. [9], who proposed streaming sparse GPs (which we call online SGPR, or O-SGPR, to distinguish from sparse spectrum GPs [42]) for incremental

Algorithm 1 Online Variational Conditioning (OVC)

Input: Data batch $(X_t, \mathbf{y}_t)$, SVGP with inducing points locations Z_{t-1} and $\phi_{t-1}(\mathbf{u}') = \mathcal{N}(\mathbf{m}_{\mathbf{u}'}, S_{\mathbf{u}'})$.

1. Compute $\tilde{\mathbf{y}}_{\mathbf{u}'} = D_{\mathbf{u}'} S_{\mathbf{u}'}^{-1} \mathbf{m}_{\mathbf{u}'}$ and $D_{\mathbf{u}'} = (S_{\mathbf{u}'}^{-1} - K_{\mathbf{u}'\mathbf{u}'}'^{-1})^{-1}$ (Eq. 6).
2. Construct GP with $\mathcal{D} = (\hat{X}, \hat{\mathbf{y}}) = ([X_t \; Z_{t-1}], [\mathbf{y}_t \; \tilde{\mathbf{y}}_{\mathbf{u}'}])$ and $\Sigma_{\hat{\mathbf{y}}} = \text{blkdiag}(\sigma^2 I, D_{\mathbf{u}'})$.
3. Compute predictive mean and covariance caches, a and $RR^\top$ as in Section 2.1.
4. Use caches to compute conditioned GP posterior on test points, X_{test}.

learning. We extend their work, providing an alternative, simpler derivation of their model that highlights the connection with SGPR [67]. Furthermore, our perspective enables us to construct a principled approach to updating inducing point locations as new data arrives, that prevents the "forgetting" of old data induced by the resampling heuristic used by Bui et al. [9].

2.3 Bayesian Optimization and Monte Carlo Acquisitions

Bayesian optimization (BO) obtains $\mathbf{x}^* = \text{argmin}_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ by constructing a probabilistic *surrogate model* of f, which in turn is used to evaluate an acquisition function. GPs are favored for BO due to their sample-efficiency and efficient posterior sampling that enables cheap, gradient based optimization of the acquisition function to acquire new points [23]. Many interesting acquisition functions look ahead into the future to see how the model will likely change if we decide to query a specific point, a procedure known as "fantasization" [31, 78, 38]. Fantasization is done by drawing samples from the current surrogate posterior at some set of points and conditioning the surrogate on those samples. For example, the batch knowledge gradient [qKG, 78, 2] is given by

$$a(x, \mathcal{D}) := \mathop{\mathbb{E}}_{f(x) \sim p(\cdot|\mathcal{D})} \left(\max_{x' \in \mathcal{X}} \mathop{\mathbb{E}}_{f(x') \sim p(\cdot|\mathcal{D}_{+x})} f(x') \right) - \max_{x' \in \mathcal{X}} \mathop{\mathbb{E}}_{f(x') \sim p(\cdot|\mathcal{D})} f(x'), \tag{4}$$

where $\mathcal{D}_{+x} := \mathcal{D} \cup \{(x, f(x)\}$. The inner expectation in the first term requires conditioning the surrogate model on posterior samples at x, before optimizing through predictions of the conditioned surrogate model. The goal is to simulate the effect on the model if we had observed the batch of data.

Use of sparse GPs in BO: sparse GPs have not seen wide adoption in the BO community, with only several preliminary studies that have mostly used basic acquisitions. Nickson et al. [52] and Krityakierne and Ginsbourger [41] used expected improvement (EI) with SGPR on several test problems, while McIntire et al. [48] proposed a sparse GP method using EI to tune free electron lasers [19]. Stanton et al. [64] proposed WISKI, an online implementation of a scalable kernel approach called SKI [77], for low-dimensional BO problems using batch upper confidence bound (qUCB) [2].

3 Methodology

The core idea of online variational conditioning (OVC) is to transform the inducing point locations and variational distribution into pseudo-data, which is then combined with the incoming data. We then perform inference using the combined data using a modified likelihood. As OVC relies on the incoming observations and the pseudo-data being jointly Gaussian, we extend OVC to non-Gaussian likelihoods by Laplace-approximating the likelihood of the incoming data. OVC is summarized in Algorithm 1. OVC will often be applied to compute updated posterior distributions, e.g. $p(f|\mathcal{D}_{+x})$ in Eq. 4, and quantities of the posterior, during gradient-based acquisition function optimization in BO. See Appendix C.5 for a description of OVC in practice.

OVC resolves two critical shortcomings of SVGPs: First, an SVGP can only be conditioned on a new observation by re-training $\mathbf{m}_{\mathbf{u}}$ and $\mathbf{S}_{\mathbf{u}}$ through the ELBO due to the conditional independence of f given $\mathbf{u}$ [32]. Gradient-based acquisition function optimization then requires expensive higher order gradients. Second, the inducing points must be updated as well in response to the new data.

3.1 A Simpler Derivation of Streaming Sparse GPs

We now outline a new derivation of streaming sparse GPs [9] that better connects the method to SGPR, motivating the name O-SGPR and enabling us to propose a new conditioning method for

SVGPs. We now have batches of n_t observations $(X_t, \mathbf{y}_t)$, replacing the "train" and "test" subscripts with a time index t. We will denote the inducing function values $\mathbf{u}' := [f(\mathbf{z}_1), \ldots, f(\mathbf{z}_p)]^\top$ and $\mathbf{u} := [f(\mathbf{z}'_1), \ldots, f(\mathbf{z}'_p)]^\top$ for $\mathbf{z}_i \in Z_{t-1}$ and $\mathbf{z}'_i \in Z_t$ respectively. We use blue symbols to distinguish terms that can be computed before time t from those that cannot (e.g. $\mathbf{m}_{\mathbf{u}'}$ v.s. $\mathbf{m}_{\mathbf{u}}$).

Like Titsias [67], Bui et al. [9] chose ϕ_t to be the minimizer of $\mathrm{KL}(q_t(\cdot)||\hat{p}(\cdot|\mathbf{y}_t))$, derived in closed form by means of variational calculus, where $\hat{p}(\cdot|\mathbf{y}_t)) \propto p(\mathbf{y}_t|\cdot)p_t(\cdot)q_{t-1}(\cdot)/p_{t-1}(\cdot)$, an approximate predictive posterior with the previous iteration's posterior as the prior. Specifically, ϕ_t becomes

$$\phi_t(\mathbf{u}) \propto p(\mathbf{u}|\theta_t)\mathcal{N}(\hat{\mathbf{y}}; K_{\hat{\mathbf{v}}\mathbf{u}}K_{\mathbf{uu}}\mathbf{u}, \Sigma_{\hat{\mathbf{y}}}), \tag{5}$$

$$\hat{\mathbf{y}} := \begin{bmatrix} \mathbf{y}_t \\ \tilde{\mathbf{y}}_{\mathbf{u}'} \end{bmatrix}, \quad \Sigma_{\hat{\mathbf{y}}} := \begin{bmatrix} \Sigma_{\mathbf{y}_t} & 0 \\ 0 & D_{\mathbf{u}'} \end{bmatrix}, \quad D_{\mathbf{u}'} := (S_{\mathbf{u}'}^{-1} - K_{\mathbf{u}'\mathbf{u}'}'^{-1})^{-1}, \tag{6}$$

where $\tilde{\mathbf{y}}_{\mathbf{u}'} = D_{\mathbf{u}'}S_{\mathbf{u}'}^{-1}\mathbf{m}_{\mathbf{u}'}$, $K_{\hat{\mathbf{v}}\mathbf{u}} = [k_{\theta_t}(X_t, Z_t)\ k_{\theta_t}(Z_{t-1}, Z_t)]^\top$, and $K'_{\mathbf{u}'\mathbf{u}'} = k_{\theta_{t-1}}(Z_{t-1}, Z_{t-1})$ (i.e. the kernel with old-hyperparameters on the old inducing points). Bui et al. [9] then derive both an ELBO for hyper-parameter inference and a variational posterior, $q_t(f)$, for prediction.

Ultimately, the formulation of O-SGPR in Eqs. 5 and 6 reduces to SGPR. Suppose we have an SGPR model trained on the first batch of data, $(X_0, \mathbf{y}_0)$ with variational parameters $\mathbf{m}_{\mathbf{u}'}$ and $S_{\mathbf{u}'}$ with inducing locations Z_0 and projection parameters $\mathbf{c}_0, C_0$ (see Eq. 1). At the next timestep ($t = 1$) we use the expressions from Eq. (6) to define a new SGPR model with pseudo-data $(\hat{X} := [X_t\ Z_{t-1}]^\top, \hat{\mathbf{y}} := [\mathbf{y}_t\ \tilde{\mathbf{y}}_{\mathbf{u}'}]^\top)$, where the distribution of $\mathbf{y}$ is the pseudo-likelihood $\mathcal{N}(\hat{\mathbf{v}}, \Sigma_{\hat{\mathbf{y}}})$. The optimal SGPR variational distribution $\phi_t(\mathbf{u}) = \mathcal{N}(\mathbf{m}_{\mathbf{u}}, S_{\mathbf{u}})$ has the same form as Eq. (2), with the terms from Eq. (1) simplifying to become recursive expressions:

$$\mathbf{c}_t = K_{\mathbf{u}\hat{\mathbf{v}}}\Sigma_{\hat{\mathbf{y}}}^{-1}\hat{\mathbf{y}} = K_{\mathbf{uv}}\Sigma_{\mathbf{y}_t}^{-1}\mathbf{y}_t + K_{\mathbf{uu}'}(K_{\mathbf{u}'\mathbf{u}'}'^{-1} + K_{\mathbf{u}'\mathbf{u}'}'^{-1}C_{t-1}K_{\mathbf{u}'\mathbf{u}'}'^{-1})\mathbf{m}_{\mathbf{u}'}, \tag{7}$$

$$C_t = K_{\mathbf{u}\hat{\mathbf{v}}}\Sigma_{\hat{\mathbf{y}}}^{-1}K_{\hat{\mathbf{v}}\mathbf{u}} = K_{\mathbf{uv}}\Sigma_{\mathbf{y}_t}^{-1}K_{\mathbf{vu}} + K_{\mathbf{uu}'}(K_{\mathbf{u}'\mathbf{u}'}'^{-1}C_{t-1}K_{\mathbf{u}'\mathbf{u}'}'^{-1})K_{\mathbf{uu}'}. \tag{8}$$

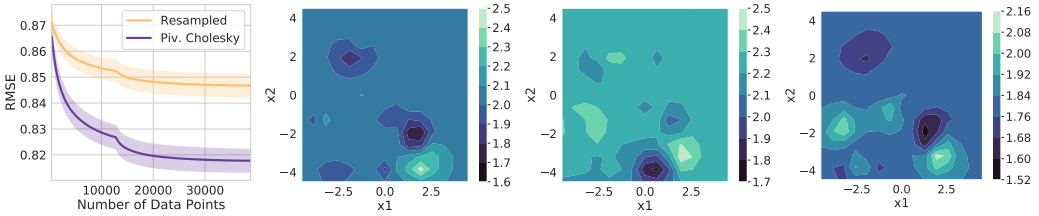
An immediate practical advantage of this perspective is that the lower block of the pseudo-likelihood covariance simplifies to $D_{\mathbf{u}'} = K'_{\mathbf{u}'\mathbf{u}'}C_{t-1}^{-1}K'_{\mathbf{u}'\mathbf{u}'}$, eliminating the matrix subtraction in Eq. (6). Similar to the caching strategy discussed in Section 2.2, O-SGPR can be efficiently implemented with low-rank cache updates (see Appendix C.2 for more details). Informally, O-SGPR can be viewed as using pseudo-data to represent the past data *and the past models* by alternately compressing data into the parameters of a variational distribution, and then projecting that distribution back into pseudo-data to be combined with a fresh batch of ground-truth data though a joint Gaussian pseudo-likelihood.

We can condition *any* variational sparse GP into an SGPR model, without touching any previous observations due to the conditional independence assumptions of variationally sparse GPs. However, the variational parameters $\mathbf{m}_{\mathbf{u}'}$ and $S_{\mathbf{u}'}$ are not guaranteed to be at the optimal solution. When the variational distribution is projected back to the pseudo-data, the projection will be to the targets and likelihood *for which the current variational parameters would be optimal*, which may not correspond well to the data that originally created the model. This potential pitfall is mitigated if $\mathbf{m}_{\mathbf{u}'}$ and $S_{\mathbf{u}'}$ are well optimized and is offset by the practical advantages of SVGPs. In Appendix C.3 we detail the necessary modifications for the whitened parameterization.

3.2 Inducing Point Selection

Here, we describe inducing point selection during the conditioning procedure to enable better variance reduction on new inputs. While heuristics including re-sampling [9] and data sufficient statistics [34] have been proposed, they either require the number of inducing points to grow or gradually forget old observations. We show in Appendix C.4 that relying exclusively on gradient-based optimization of inducing locations works very poorly in the online setting.

To update the inducing point locations during conditioning, we extend Burt et al. [10]'s batch inducing point initialization approach to heteroskedastic Gaussian likelihoods. They consider theoretical bounds on the marginal likelihood, finding that for homoscedastic Gaussian likelihoods a good strategy is to minimize the trace of the error of a rank-p Nyström approximation, e.g. $\varepsilon = tr\left(\Sigma_{\mathbf{y}}^{-1/2}(K_{\mathbf{vv}} - K_{\mathbf{vu}}K_{\mathbf{uu}}^{-1}K_{\mathbf{uv}})\Sigma_{\mathbf{y}}^{-1/2}\right) = \sigma^{-1/2}tr(K_{\mathbf{vv}} - Q_{\mathbf{vv}})$ for $Q_{\mathbf{vv}} = K_{\mathbf{vu}}K_{\mathbf{uu}}^{-1}K_{\mathbf{uv}}$. They follow a classical approach of Fine and Scheinberg [22] by a greedy minimiziation strategy: choosing as inducing points the pivots of a rank p pivoted Cholesky factorization of $K_{\mathbf{vv}}$.



(a) Incremental learning (b) Exact GP (c) O-SGPR conditioning (d) Exact conditioning

Figure 2: **(a)** Incremental learning RMSE on the UCI *protein* dataset. Pivoted cholesky initialization outperforms resampling. **(b-d)** Knowledge gradient acquisition surfaces on Branin-2. for an exact GP **(b)**, a SVGP conditioned into a SGPR model **(c)** (more like O-SGPR) and a SVGP conditioned into an exact GP **(d)**, our OVC approach. Conditioning into an exact GP faithfully replicates most of the local peaks and valleys of the exact GP's acquisition surface.

In our case the covariance of the pseudo-likelihood is no longer homoscedastic, so the slack term becomes $\varepsilon = \mathrm{tr}(\Sigma_{\hat{\mathbf{y}}}^{-1/2}(K_{\hat{\mathbf{v}}\hat{\mathbf{v}}} - Q_{\hat{\mathbf{v}}\hat{\mathbf{v}}})\Sigma_{\hat{\mathbf{y}}}^{-1/2})$ and hence the pivoted Cholesky decomposition is instead performed over $\Sigma_{\hat{\mathbf{y}}}^{-1/2} K_{\hat{\mathbf{v}}\hat{\mathbf{v}}} \Sigma_{\hat{\mathbf{y}}}^{-1/2}$. When compared to re-sampling the inducing points [9], pivoted cholesky updates perform significantly better, as shown in Figure 2a on the UCI protein dataset [18]. Experimental details are given in Appendix D.2.

3.3 Conditioning into an Exact GP

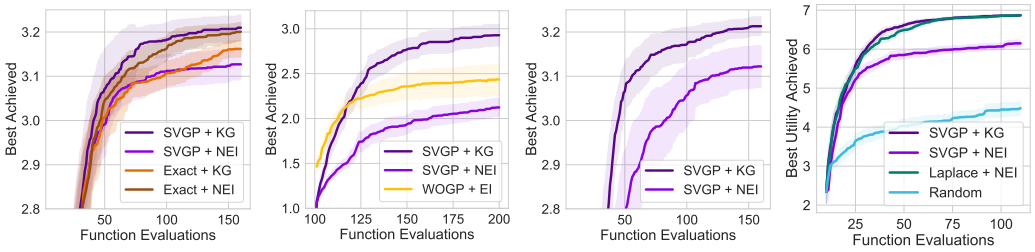
In the context of BO, we condition on *hypothetical* observations, and the conditioned surrogates are discarded after each acquisition function evaluation. Since the SVGP will not be conditioned on more than a few batches of observations, we can sidestep the issue of updating inducing locations entirely by instead conditioning into an *exact GP* trained the combined pseudo-data through the pseudo-likelihood. That is, we model the data as $\hat{\mathbf{y}} \sim \mathcal{N}(f, \Sigma_{\hat{\mathbf{y}}})$ (Gaussian with block-diagonal covariance) assuming $f \sim \mathcal{GP}(\mu_\theta, k_\theta(x, x'))$. We reach the same model by choosing $\hat{X}$ as the inducing points in our conditioned SGPR (Section 3.1). For small n_t, an exact GP is not much slower, taking only $(n_t + p)^3$ computations instead of p^3 computations, further reduced by low-rank updates to compute the predictive mean and covariance caches, a, and R (Section 2.1).

We demonstrate the superiority of conditioning into an exact GP, rather than one with inducing points in Figure 2, where we display the knowledge gradient (KG) acquisition surface for exact GPs, a SVGP conditioned into a SGPR model, and a SVGP conditioned into an exact GP on the Branin-2 test problem. The acquisition surface for KG for conditioning into the exact GP (Figure 2d) contains the same local and global optima as KG evaluated on the exact GP (Figure 2b). By comparison, the the SVGP conditioned into SGPR contains illusory optima (Figure 2c).

3.4 Local Laplace Approximations for Non-Gaussian Observations

Thus far, we have solely considered Gaussian observations. The introduction of a non-Gaussian likelihood presents a new challenge, since it implies that the current observation batch and the pseudo-data are no longer jointly Gaussian. To adapt the conditioning procedure to the non-Gaussian setting, we can simply perform a Laplace approximation of the likelihood at the new points. Specifically, this gives us an approximate likelihood, $\hat{p}(y|f) = \mathcal{N}(\tilde{y}; f, \mathcal{H}_*^{-1})$, where $\mathcal{H}_* = \nabla_f^2 \log p(y|f)|_{f^*(y)}$ and $f^*(y)$ is the maximizer of $\log p(y|f)$, computed via Newton iteration. When conditioning on new observations $\mathbf{y}_t$, we substitute $f^*(\mathbf{y}_t)$ instead. That is, our new model has data points $\hat{\mathbf{y}} = (f^*(\mathbf{y}_t), \tilde{\mathbf{y}}_{\mathbf{u}'})$ instead of $(\mathbf{y}_t, \tilde{\mathbf{y}}_{\mathbf{u}'})$, and the pseudo-likelihood remains Gaussian with covariance $\Sigma_{\hat{\mathbf{y}}} = \mathrm{blkdiag}(\mathcal{H}_*^{-1}, D_{\mathbf{u}'})$. We primarily consider natural parameterizations of exponential families, so that $\mathcal{H}_*$ is positive, diagonal and depends solely on f. Computing $\nabla \log p(y|f)$ and $\mathcal{H}_*$ is possible by hand but one can also use automatic differentiation [AD, 58].[1] In Appendix D, Figure 8 we show the effect of repeated Laplace approximations across several batches for online classification.

[1] Specifically, we use PyTorch's functional API, `https://pytorch.org/docs/stable/autograd.html#functional-higher-level-api`.



(a) Hartmann6 (b) Laser (c) Poisson-Hartmann6 (d) Preference Learning

Figure 3: **(a)** Hartmann6 test problem with one constraint. Here, SVGPs with NEI and KG match the performance of exact GPs with NEI and KG. **(b)** Free electron laser problem from McIntire et al. [48]; SVGPs with knowledge gradient outperforming weighted sparse GPs. **(c)** Constrained Hartmann6 test problem with count responses (Poisson likelihood). Only SVGPs can be used here, and qKG outperforms qNEI. **(d)** Preference learning; SVGPs with qKG slightly outperform Laplace approximations with NEI, and outperform qNEI with SVGPs.

4 Experiments

All experiments use PyTorch [57], GPyTorch [24], and BoTorch [2]. Unless otherwise specified, we run each experiment 50 times and report the mean and two standard deviations of the mean. Further experimental details and dataset descriptions are in the Appendix. Our experimental evaluation demonstrates that SVGPs using OVC can be successfully used as surrogate models with advanced acquisition functions in Bayesian optimization, even in the large batch and non-Gaussian settings. In keeping with BO literature, we will refer to the query batch size as q (not to be confused with the variational posterior $q(f)$ in previous sections). *All SVGP models that use conditioning (or fantasization) require OVC to even be practical to implement.*

4.1 Knowledge Gradient with SVGPs

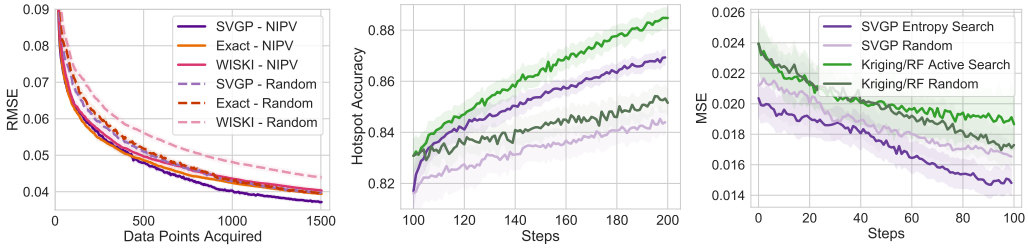
These experiments use the one-shot formulation of the batch knowledge gradient (qKG) (Eq. 4) from Balandat et al. [2], who demonstrated that qKG outperforms other acquisitions due to being able to plan two steps into the future. Using and optimizing qKG has only been available for exact GPs previously. By using OVC, we have enabled SVGPs to also efficiently and tractably optimize qKG, even for non-Gaussian observations. We compare to batch noisy expected improvement [qNEI, 43] which is myopic and does not use fantasization (e.g. conditioning).

Gaussian observations: We use the **Hartmann6** test function, with one black box constraint, maximizing $f(x) = -\sum_{i=1}^{4} \alpha_i \exp\{-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\}$ subject to the constraint that $c(x) = ||x||_1 \leq 3$ for fixed A, P, α. We use 10 initial points and a batch size of 3 optimizing for 50 iterations, comparing to SVGPs and exact GPs using qNEI. We show the results in Figure 3a where SVGPs with qKG match exact GPs with both qNEI and qKG, and outperform SVGPs using qNEI.

Second, we mimic the **laser tuning** experiment of [48, 19], demonstrating that SVGPs outperform even weighted online GPs (WOGP), which were designed for this task. Here, we use 100 initial points, with $d = 14$, and and wish to tune a laser's output energy as a function of the magnet settings that produce the beam. Like McIntire et al. [48] we treat a pretrained GP fit on experimental data as a simulator. We use a batch size of 1, finding that SVGPs + KG outperform WOGP (Figure 3b). However, exact GPs outperform the variational approaches (Appendix Fig. 12b).

Non-Gaussian likelihoods: Next, we extend the knowledge gradient to problems with non-Gaussian likelihoods. First, we take the constrained Hartmann6 test function from the previous section, and use **Poisson** responses, $y \sim \text{Poisson}(\exp\{f(x)\})$, repeating the same settings as for the Gaussian case. Now, the data is non-Gaussian and cannot be well-modelled by a Gaussian likelihood, so we compare to only SVGPs with qNEI. qNEI is outperformed by qKG, as shown in Figure 3c.

Second, in Figure 3d, we consider a **preference learning** problem. Here, the latent data is described by $f(x) = -10^{-1/2} \sum_{i=1}^{10} \sqrt{i} \cos(2\pi x_i)$ for $x \in [0, 1]^{10}$, comparing to Laplace approximations [14]. Again, we see that SVGPs with qKG outperform qNEI with both SVGPs and Laplace approximations.



(a) Malaria incidence in Nigeria (b) Hotspot prediction accuracy (c) Prevalence modelling

Figure 4: **(a)** Active learning of malaria incidence from satellite data. Using qNIPV outperforms randomly selecting points, while the SVGPs slightly outperform both exact GPs and WISKI. **(b,c)** Active learning of schistomiasis incidence in Cote d'Ivoire from Andrade-Pacheco et al. [1]. Comparison is to the random forest based approach using active sampling. While the GP based models are somewhat less accurate at predicting hotspots **(b)**, they are better as a global model of prevalence **(c)**.

4.2 Active Learning of Disease Incidence

We next present results for two active learning tasks governing the collection of disease incidence data. In both tasks the acquisition functions again require efficient conditioning on hypothetical data, and the second task has Binomial responses, so exact GPs cannot be applied. In both settings, applying OVC to SVGPs gives strong results competitive with either exact GPs or random forests.

Modelling of Malaria Incidence: We consider data from the Malaria Global Atlas [75] describing the infection rate of a parasite known to cause malaria in 2017. We wish to choose spatial locations to query malaria incidence in order to make the best possible predictions on a withheld test set, the entire country of Nigeria. Following Stanton et al. [64], we minimize the negative integrated posterior variance [NIPV, 62], defined as $a(x; \mathcal{D}) := -\int_{\mathcal{X}} \mathbb{E}(\mathbb{V}(f(x)|\mathcal{D}_{+x})|\mathcal{D})dx$, again with $\mathcal{D}_{+x} = \mathcal{D} \cup \{(x, y(x))\}$. Intuitively, the minimizer of this acquisition will be the batch of data points that when added into the model will most reduce the total posterior uncertainty across the domain, requiring efficient conditioning to do so in a tractable manner. The results are shown for a batch size of $q = 6$ across 15 trials in Figure 4a where we see that each method outperforms random baselines. Perhaps due to the optimization freedom, the SVGP outperforms both the exact GP and WISKI [64].

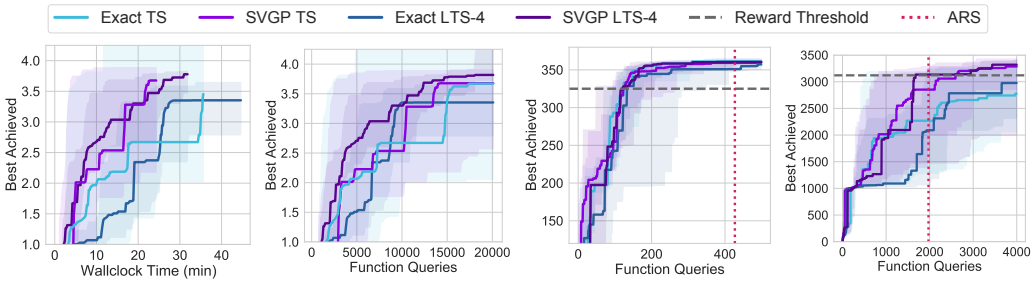
Hotspot Modelling: We follow Andrade-Pacheco et al. [1] and model the prevalence of schistomiatosis in Côte d'Ivoire using simulated responses from 1500 villages in that country, and taking into account six other demographic variables. We model the responses y (incidence) at locations x with population $n(x)$ with a Binomial likelihood $p(\hat{y}|f, x) \sim \text{Binomial}(n(x), r(f))$, where $r(f) = (1 + \exp\{-f\})^{-1}$. Letting $\tau \in (0, 1)$ be a threshold on the prevalence for a location to be considered a "hotspot" [1], we compute the entropy:

$$h_\tau(x, \mathcal{D}) := \mathbb{E}_{p(f|\mathcal{D})}(\mathbb{H}(\text{Bernoulli}(f > \text{logit}(\tau)))) \approx \frac{1}{K}\sum_{i=1}^{K} 1_{f > \text{logit}(\tau)} \mathbb{H}(\text{Bernoulli}(f)),$$

taking the acquisition value to be the reduction in the entropy of the posterior predictive distribution over the incidence under the hotspot-focused likelihood. $a_\tau(x, \mathcal{D}) := \int_{x' \in \mathcal{X}} \big(h_\tau(x', \mathcal{D}_{+x}) - h_\tau(x', \mathcal{D})\big)dx'$. A location is given a high acquisition value if observing the incidence at that location reduces the uncertainty of the model on the predicted set of hotspots. In Figure 4b, we compare to Andrade-Pacheco et al. [1] who use spatial kriging on the residuals of a random forest model. Both their random baseline and their exploration based procedure (a variant of UCB) start off with higher prediction accuracy; however our SVGP models ultimately outperform the kriging approach with random selection. The SVGP is a better predictor of true prevalence, as shown in Figure 4c. In both cases, our acquisition function significantly outperforms random selection with a SVGP surrogate.

4.3 Rollouts within Thompson Sampling for High Dimensional BO

For our final set of experiments, we solve control problems using trust region Bayesian optimization [TurBO, 21]. Inspired by multi-step look-aheads [38, 4], we propose h-step look-ahead Thompson sampling (LTS-h). In BO, Thompson sampling (TS) is often implemented by drawing samples from



(a) Wall-Clock time, rover, $d = 60$

(b) rover, $d = 60$

(c) `swimmer`, $d = 16$

(d) `hopper`, $d = 33$

Figure 5: Multi-step rollouts with OVC and SVGPs provides sample-complexity and wall-clock time improvements on high dimensional BO problems when using TurBO and LAMCTS [21, 72]. SVGP rollouts are as time efficient (to 150 iterations) as standard TS **(a)** on lunar rover, $d = 60$. **(c-d)** MuJoCo environments using LAMCTS + TurBO. Also shown is the reward threshold (dashed grey lines) and augmented random search (ARS)'s performance (dotted red lines) [46]. The median and its 95% confidence interval are shown over 24 trials for rover and 10 trials for swimmer and hopper.

the posterior at points all over the domain, then selecting the q best to form a query batch [65, 21]. LTS-h extends the idea by conditioning the surrogate independently on each posterior sample in the original TS query batch, Thompson sampling again from the updated posterior with a new set of points, and appending the best sample to its predecessor to form a *path*. The process is repeated h times. Finally, we condition the original surrogate jointly on each path, then perform TS again to choose the new query batch. Informally, each path corresponds to a distinct, coherent draw from $p(f|\mathcal{D})$, allowing the inner-loop to refine its guess of the global optimum for different f, and the final round of TS chooses the query batch based on those guesses. See Appendix C.6 for a formal description. Like other look-ahead acquisitions, LTS-h is only practical if posterior conditioning and samples are very efficient and numerically stable. LTS-h is conceptually similar to path sampling for look-ahead in Jiang et al. [38] and kriging believer [25].

For validation, we consider tuning the 60 dimensional path that a **lunar rover** takes across a field stacked with obstacles [73, 21]. We use batches of $q = 100$ with 200 initial points, and use trust region Bayesian optimization (TurBO) to split up the space effectively [21], comparing to TurBO with Thompson sampling (TS) as the acquisition [65]. We show the wall clock times per iteration in Figure 5a, where we see that the TurBO + LTS approaches compare well in wallclock time to TurBO+TS, while being more function efficient (Figure 5b).

Finally, we consider **MuJoCo** problems using the OpenAI gym [69, 7] with LTSs inside of TurBO with Monte Carlo Tree Search [72]. Following Wang et al. [72], we learn a linear policy and consider the `swimmer-v2, hopper-v2` environments over 10 trials displaying the median and its 95% confidence band due to high variance. On both problems, SVGP with LTSs tend to be the most sample efficient, with SVGP + TS performing at least as well on `swimmer-v2`. We also show the reward threshold and the performance achieved by augmented random search (ARS), which is a strong baseline reinforcement learning method that uses random search to tune linear controllers [46]. Our results suggest that LTSs are promising overall; however, more work needs to be done for high dimensional kernels on these types of stiff robotic control problems.

5 Discussion

In conclusion, we have demonstrated how to condition on new data points with stochastic variational Gaussian processes using a new formulation of the online SGPR model of Bui et al. [9]. Our conditioning approach can be modified to return an exact GP, while still being able to accommodate any likelihood through Laplace approximations. As a result, we have decoupled look-ahead BO acquisition functions from their dependence on exact GP inference through a Gaussian likelihood, increasing the range, scale, and efficiency of BO applications. In the future, we hope to extend OVC to multi-task and deep Gaussian processes for use in Bayesian optimization [30, 16].

References

[1] Andrade-Pacheco, R., Rerolle, F., Lemoine, J., Hernandez, L., Meïté, A., Juziwelo, L., Bibaut, A. F., van der Laan, M. J., Arnold, B. F., and Sturrock, H. J. (2020). Finding hotspots: development of an adaptive spatial sampling approach. *Scientific reports*, 10(1):1–12.

[2] Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. (2020). BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems*, volume 33.

[3] Bauer, M., Van Der Wilk, M., and Rasmussen, C. (2016). Understanding probabilistic sparse gaussian process approximations. In *Advances in Neural Information Processing Systems*, volume 29, pages 1533–1541.

[4] Bertsekas, D. P. (2020). *Rollout, Policy Iteration, and Distributed Reinforcement Learning*. Athena Scientific.

[5] Bijl, H., Schön, T. B., van Wingerden, J.-W., and Verhaegen, M. (2016). Online sparse gaussian process training with input noise. *arXiv preprint arXiv 1601.08068*.

[6] Boedecker, J., Springenberg, J. T., Wülfing, J., and Riedmiller, M. (2014). Approximate real-time optimal control based on sparse gaussian process models. In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 1–8. IEEE.

[7] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

[8] Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. (2013). Streaming variational bayes. In *Advances in Neural Information Processing Systems*, volume 26.

[9] Bui, T. D., Nguyen, C. V., and Turner, R. E. (2017). Streaming sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems*, volume 31, pages 3301–3309, Long Beach, California, USA. Curran Associates Inc.

[10] Burt, D., Rasmussen, C. E., and Van Der Wilk, M. (2019). Rates of convergence for sparse variational Gaussian process regression. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 862–871. PMLR.

[11] Canu, S. and Smola, A. (2006). Kernel methods and the exponential family. *Neurocomputing*, 69(7):714–720. New Issues in Neurocomputing: 13th European Symposium on Artificial Neural Networks.

[12] Cheng, C.-A. and Boots, B. (2016). Incremental variational sparse Gaussian process regression. In *Advances in Neural Information Processing Systems*, volume 30, pages 4410–4418, Barcelona, Spain. Curran Associates Inc.

[13] Chowdhary, G., Kingravi, H. A., How, J. P., and Vela, P. A. (2014). Bayesian nonparametric adaptive control using gaussian processes. *IEEE transactions on neural networks and learning systems*, 26(3):537–550.

[14] Chu, W. and Ghahramani, Z. (2005). Preference learning with gaussian processes. In *Proceedings of the 22nd International Conference on Machine learning*, pages 137–144.

[15] Csató, L. and Opper, M. (2002). Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668.

[16] Cutajar, K., Pullin, M., Damianou, A., Lawrence, N., and González, J. (2019). Deep gaussian processes for multi-fidelity modeling. *arXiv preprint arXiv:1903.07320*.

[17] Deisenroth, M. and Rasmussen, C. E. (2011). Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning*, pages 465–472.

[18] Dua, D. and Graff, C. (2017). UCI machine learning repository.

[19] Duris, J., Kennedy, D., Hanuka, A., Shtalenkova, J., Edelen, A., Baxevanis, P., Egger, A., Cope, T., McIntire, M., Ermon, S., et al. (2020). Bayesian optimization of a free-electron laser. *Physical review letters*, 124(12):124801.

[20] Eriksson, D. and Jankowiak, M. (2021). High-dimensional bayesian optimization with sparse axis-aligned subspaces. *arXiv preprint arXiv:2103.00349*.

[21] Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. (2019). Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[22] Fine, S. and Scheinberg, K. (2001). Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2(Dec):243–264.

[23] Frazier, P. I. (2018). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.

[24] Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., and Wilson, A. G. (2018). GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In *Advances in Neural Information Processing Systems*, volume 31.

[25] Ginsbourger, D., Le Riche, R., and Carraro, L. (2010). Kriging is well-suited to parallelize optimization. In *Computational intelligence in expensive optimization problems*, pages 131–162. Springer.

[26] Girard, A., Rasmussen, C. E., Candela, J. Q., and Murray-Smith, R. (2002). Gaussian process priors with uncertain inputs-application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems*, volume 15.

[27] Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276.

[28] Groot, P., Lucas, P., and Bosch, P. (2011). Multiple-step time series forecasting with sparse gaussian processes. In *Causmaecker, P. De (ed.), Proceedings of the 23rd Benelux Conference on Artificial Intelligence*, pages 1–8. [Sl: sn].

[29] Hagan, P. S., Kumar, D., Lesniewski, A. S., and Woodward, D. E. (2002). Managing smile risk. *The Best of Wilmott*, 1:249–296.

[30] Hebbal, A., Brevault, L., Balesdent, M., Talbi, E.-G., and Melab, N. (2021). Bayesian optimization using deep gaussian processes with applications to aerospace system design. *Optimization and Engineering*, 22(1):321–361.

[31] Hennig, P. and Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6).

[32] Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA. AUAI Press.

[33] Hensman, J., Matthews, A., and Ghahramani, Z. (2015). Scalable Variational Gaussian Process Classification. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 351–360, San Diego, California, USA. PMLR.

[34] Hoang, T. N., Hoang, Q. M., and Low, B. K. H. (2015). A unifying framework of anytime sparse gaussian process regression models with stochastic variational inference for big data. In *International Conference on Machine Learning*, volume 37, pages 569–578. PMLR.

[35] Huber, M. F. (2013). Recursive gaussian process regression. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3362–3366. IEEE.

[36] Huber, M. F. (2014). Recursive gaussian process: On-line regression and learning. *Pattern Recognition Letters*, 45:85–91.

[37] Jankowiak, M., Pleiss, G., and Gardner, J. (2020). Parametric gaussian process regressors. In *International Conference on Machine Learning*, volume 119, pages 4702–4712. PMLR.

[38] Jiang, S., Jiang, D., Balandat, M., Karrer, B., Gardner, J., and Garnett, R. (2020). Efficient Nonmyopic Bayesian Optimization via One-Shot Multi-Step Trees. In *Advances in Neural Information Processing Systems*, volume 33.

[39] Kapoor, S., Karaletsos, T., and Bui, T. D. (2020). Variational auto-regressive gaussian processes for continual learning. *arXiv preprint arXiv:2006.05468*.

[40] Knudde, N., van der Herten, J., Dhaene, T., and Couckuyt, I. (2017). GPflowOpt: A Bayesian Optimization Library using TensorFlow. *arXiv preprint arXiv:1711.03845*.

[41] Krityakierne, T. and Ginsbourger, D. (2015). Global optimization with sparse and local gaussian process models. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 185–196. Springer.

[42] Lázaro-Gredilla, M., Quinonero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. (2010). Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 11:1865–1881.

[43] Letham, B., Karrer, B., Ottoni, G., Bakshy, E., et al. (2019). Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519.

[44] Ling, C. K., Low, K. H., and Jaillet, P. (2016). Gaussian process planning with lipschitz continuous reward functions: Towards unifying bayesian optimization, active learning, and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

[45] MacKenzie, D. and Spears, T. (2014). 'the formula that killed wall street': The gaussian copula and modelling practices in investment banking. *Social Studies of Science*, 44(3):393–417.

[46] Mania, H., Guy, A., and Recht, B. (2018). Simple random search of static linear policies is competitive for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31, pages 1805–1814.

[47] Matthews, A. G. d. G. (2017). *Scalable Gaussian process inference using variational methods*. PhD thesis, University of Cambridge.

[48] McIntire, M., Ratner, D., and Ermon, S. (2016). Sparse gaussian processes for bayesian optimization. In *Uncertainty in Artifical Intelligence*, volume 32. Association for Uncertainty in Artifical Intelligence (AUAI).

[49] Moreno-Muñoz, P., Artés-Rodríguez, A., and Álvarez, M. A. (2019). Continual multi-task gaussian processes. *arXiv preprint arXiv:1911.00002*.

[50] Moreno-Muñoz, P., Artés-Rodríguez, A., and Álvarez, M. A. (2020). Recyclable gaussian processes. *arXiv preprint arXiv:2010.02554*.

[51] Moss, H. B., Leslie, D. S., Gonzalez, J., and Rayson, P. (2021). Gibbon: General-purpose information-based bayesian optimisation. *arXiv preprint arXiv:2102.03324*.

[52] Nickson, T., Osborne, M. A., Reece, S., and Roberts, S. (2014). Automated machine learning using stochastic algorithm tuning. In *NIPS Workshop on Bayesian Optimization*.

[53] Opper, M. and Archambeau, C. (2009). The variational gaussian approximation revisited. *Neural computation*, 21(3):786–792.

[54] Osborne, M. A. (2010). *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, Oxford University, UK.

[55] Pan, P., Swaroop, S., Immer, A., Eschenhagen, R., Turner, R. E., and Khan, M. E. (2020). Continual deep learning by functional regularisation of memorable past. *arXiv preprint arXiv:2004.14070*.

[56] Pan, Y., Yan, X., Theodorou, E. A., and Boots, B. (2017). Prediction under uncertainty in sparse spectrum gaussian processes with applications to filtering and control. In *International Conference on Machine Learning*, volume 70, pages 2760–2768. PMLR.

[57] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[58] Pearlmutter, B. A. (1994). Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160.

[59] Pleiss, G., Gardner, J. R., Weinberger, K. Q., and Wilson, A. G. (2018). Constant-Time Predictive Distributions for Gaussian Processes. In *Artificial Intelligence and Statistics*, volume 84. PMLR.

[60] Rasmussen, C. E. and Williams, C. K. I. (2008). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 3. print edition.

[61] Sæmundsson, S., Hofmann, K., and Deisenroth, M. (2018). Meta reinforcement learning with latent variable gaussian processes. In *34th Conference on Uncertainty in Artificial Intelligence*, volume 34, pages 642–652. Association for Uncertainty in Artificial Intelligence (AUAI).

[62] Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000). Gaussian process regression: Active data selection and test point rejection. In *Neural Networks, IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 3241–3241.

[63] Shi, J., Titsias, M., and Mnih, A. (2020). Sparse orthogonal variational inference for gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, volume 108, pages 1932–1942. PMLR.

[64] Stanton, S., Maddox, W. J., Delbridge, I., and Wilson, A. G. (2021). Kernel Interpolation for Scalable Online Gaussian Processes. In *Artificial Intelligence and Statistics*, volume 130. PMLR.

[65] Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

[66] Titsias, M. (2008). Variational Model Selection for Sparse Gaussian Process Regression. *www.aueb.gr/users/mtitsias/papers/sparseGPv2.pdf*, page 20.

[67] Titsias, M. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Artificial Intelligence and Statistics*, pages 567–574. PMLR. ISSN: 1938-7228.

[68] Titsias, M. K., Schwarz, J., Matthews, A. G. d. G., Pascanu, R., and Teh, Y. W. (2019). Functional regularisation for continual learning with gaussian processes. In *International Conference on Learning Representations*.

[69] Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE.

[70] Tripp, A., Daxberger, E., and Hernández-Lobato, J. M. (2020). Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33.

[71] Van der Wilk, M. (2019). *Sparse Gaussian process approximations and applications*. PhD thesis, University of Cambridge.

[72] Wang, L., Fonseca, R., and Tian, Y. (2020). Learning search space partition for black-box optimization using monte carlo tree search. In *Advances in Neural Information Processing Systems*, volume 33.

[73] Wang, Z., Gehring, C., Kohli, P., and Jegelka, S. (2018). Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, volume 84, pages 745–754. PMLR.

[74] Wang, Z. and Jegelka, S. (2017). Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning*, pages 3627–3635. PMLR.

[75] Weiss, D. J., Lucas, T. C., Nguyen, M., Nandi, A. K., Bisanzio, D., Battle, K. E., Cameron, E., Twohig, K. A., Pfeffer, D. A., Rozier, J. A., et al. (2019). Mapping the global prevalence, incidence, and mortality of plasmodium falciparum, 2000–17: a spatial and temporal modelling study. *The Lancet*, 394(10195):322–331.

[76] Wilson, A. and Ghahramani, Z. (2010). Copula Processes. In *Advances in Neural Information Processing Systems*, volume 23, page 9.

[77] Wilson, A. and Nickisch, H. (2015). Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP). In *International Conference on Machine Learning*, pages 1775–1784. ISSN: 1938-7228 Section: Machine Learning.

[78] Wu, J. and Frazier, P. I. (2016). The parallel knowledge gradient method for batch Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 30 of *NIPS'16*, pages 3134–3142, Red Hook, NY, USA. Curran Associates Inc.

[79] Xu, N., Low, K. H., Chen, J., Lim, K. K., and Ozgul, E. (2014). Gp-localize: Persistent mobile robot localization using online sparse gaussian process observation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Appendix A
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Appendix A
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Appendix D.2.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Section 4 and Appendix D.2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Appendix D.2.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] Appendix D.2.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See attached code.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]