2nd Symposium on Advances in Approximate Bayesian Inference, 2019 1–9

Distributional Bayesian optimisation for variational inference on black-box simulators

Rafael Oliveira RAFAEL.OLIVEIRA@SYDNEY.EDU.AU
Centre for Translational Data Science, the University of Sydney, Australia

Lionel Ott LIONEL.OTT@SYDNEY.EDU.AU
School of Computer Science, the University of Sydney, Australia

Fabio Ramos FABIO.RAMOS@SYDNEY.EDU.AU
The University of Sydney, Australia & NVIDIA, USA

Abstract

Inverse problems are ubiquitous in natural sciences and refer to the challenging task of inferring complex and potentially multi-modal posterior distributions over hidden parameters given a set of observations. Typically, a model of the physical process in the form of differential equations is available but leads to intractable inference over its parameters. While the forward propagation of parameters through the model simulates the evolution of the system, the inverse problem of finding the parameters given the sequence of states is not unique. In this work, we propose a generalisation of the Bayesian optimisation framework to approximate inference. The resulting method learns approximations to the posterior distribution by applying Stein variational gradient descent on top of estimates from a Gaussian process model. Preliminary results demonstrate the method's performance on likelihood-free inference for reinforcement learning environments.

1. Introduction

We consider the problem of estimating parameters $\boldsymbol{\theta}$ of a physical system according to observed data $\mathbf{y}$. The forward model of the system is approximated by a computational model that generates data $\hat{\mathbf{y}}_{\boldsymbol{\theta}}$ based on the given parameter settings $\boldsymbol{\theta}$. In many cases, the corresponding likelihood function $p(\hat{\mathbf{y}}_{\boldsymbol{\theta}}|\boldsymbol{\theta})$ is not available, and one resorts to likelihood-free methods, such as approximate Bayesian computation (ABC) (Robert, 2016), conditional density estimation (Papamakarios and Murray, 2016), etc. For certain applications in robotics and reinforcement learning, however, the number of simulations might be limited by resource constraints, imposing challenges to current approaches.

Recent methods address the problem of efficiency in the use of simulations by either constructing conditional density estimators from joint data $\{\boldsymbol{\theta}_i, \hat{\mathbf{y}}_i\}_{i=1}^N$, using, for example, mixture density networks (Papamakarios and Murray, 2016; Ramos et al., 2019), or by sequentially learning approximations to the likelihood function (Gutmann and Corander, 2016; Papamakarios et al., 2019) and then running Markov chain Monte Carlo (MCMC). In particular, Gutmann and Corander (2016) derive an active learning approach using Bayesian optimisation (BO) (Shahriari et al., 2016) to propose parameters for simulations. Their approach reduces the number of simulator runs from the typical thousands to a few hundreds.

This paper investigates an approach to combine the flexible representative power of variational inference methods (Liu and Wang, 2016) with the data efficiency of Bayesian optimisation. We present a Thompson sampling strategy (Russo and Van Roy, 2016) to sequentially refine variational approximations to a black-box posterior. Parameters for new simulations are proposed by running Stein variational gradient descent (SVGD) (Liu and Wang, 2016) over samples from a Gaussian process (GP) (Rasmussen and Williams, 2006). The approach is also equipped with a method to optimally subsample the variational approximations for batch evaluations of the simulator models at each round. In the following, we present the derivation of our approach and preliminary experimental results.

2. Distributional Bayesian optimisation

Our goal is to estimate a distribution q that approximates a posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ over simulator parameters $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^d$ given observations $\mathbf{y}$ from a target system. We assume no access to a likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$, but only to a discrepancy measure [1] between simulator outputs and observations $\Delta_{\boldsymbol{\theta}}$, as in Gutmann and Corander (2016).

We take a Bayesian optimisation approach to find the optimal q^* by minimising a discrepancy between q and the target p:

$$q^* \in \operatorname*{argmin}_{q \in \mathcal{Q}} \mathbb{S}(q, p) \ . \tag{1}$$

where $\mathbb{S}$ represents the kernelised Stein discrepancy (KSD) (Liu et al., 2016).[2] We solve Equation 1 via a black-box approach which does not require gradients of the target distribution p nor its availability in closed form. The resulting BO algorithm is composed of a GP model to form an approximate likelihood, a Thompson sampling acquisition function to select candidate distributions and a kernel herding procedure to optimally select samples of simulator parameters.

2.1. Modelling

A standard BO approach would place a GP to model the map from q's parameters to the corresponding KSD. However, such parameter space holds a weak connection with the original Θ and is possibly higher-dimensional. We choose to bypass this step by learning q directly via Stein variational gradient descent (SVGD) (Liu and Wang, 2016).

Applying SVGD directly to Equation 1 would require gradients of the target $\log p$. In our case, we have that:

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathbf{y}) = \nabla_{\boldsymbol{\theta}} \log p(\mathbf{y}|\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) \ .$$

As $p(\mathbf{y}|\boldsymbol{\theta})$ is unavailable, we use a GP to model $g : \boldsymbol{\theta} \mapsto -\Delta_{\boldsymbol{\theta}}$, which defines a synthetic likelihood function (Gutmann and Corander, 2016), i.e.:

$$p(\mathbf{y}|\boldsymbol{\theta}) \approx \hat{p}(\mathbf{y}|\boldsymbol{\theta}) \propto e^{-\Delta_{\boldsymbol{\theta}}} \ . \tag{2}$$

1. The ABC literature offers a plenitude of choices for $\Delta_{\boldsymbol{\theta}}$. For a review, we refer the reader to Gutmann and Corander (2016) and Robert (2016). Our choice for experiments is given in Section 3.
2. Background details on the KSD are presented in the appendix (Section A.1).

The simulations-observations discrepancy $\Delta_{\boldsymbol{\theta}}$ is possibly expensive to evaluate and not differentiable, due to the need of running a black-box simulator. The GP then provides an approximation which is cheap to evaluate and whose sample functions are differentiable for smooth kernels, allowing us to apply SVGD in the BO loop.

2.2. A posterior sampling approach

We propose selecting candidate distributions $q_n \in \mathcal{Q}$ based on a GP posterior sampling approach known as Thompson sampling (Russo and Van Roy, 2016), which has been successfully applied to BO problems in the case of selecting point candidates $\boldsymbol{\theta} \in \Theta$ (Chowdhury and Gopalan, 2017; Kandasamy et al., 2018; Mutný and Krause, 2018). Thompson sampling accounts for uncertainty in the model by sampling functions from the GP posterior. For models based on finite feature maps, such as sparse spectrum Gaussian processes (SSGPs) (Lázaro-Gredilla et al., 2010), the Thompson sampling approach resumes to sampling weights $\mathbf{w}_n$ from a multivariate Gaussian (Appendix A.2), so that:

$$g_n(\boldsymbol{\theta}) = \mu_0(\boldsymbol{\theta}) + \mathbf{w}_n^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{\theta}) \ , \quad \boldsymbol{\theta} \in \Theta \ , \tag{3}$$

constitutes a sample from the posterior of a SSGP with mean function μ_0 and feature map $\boldsymbol{\phi}$. Recalling the objective in Equation 1, we can now define the acquisition function as:

$$h(q|\mathcal{D}_n) = -\mathbb{S}(q, \hat{p}_n) \ , \tag{4}$$

where $\hat{p}_n(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}) e^{g_n(\boldsymbol{\theta})}$ corresponds to an approximation to the target posterior $p(\boldsymbol{\theta}|\mathbf{y})$ based on g_n.

SVGD represents the variational distribution q as a set of particles $\{\boldsymbol{\theta}_i\}_{i=1}^M$ forming an empirical distribution. The particles are initialised as i.i.d. samples from the prior $p(\boldsymbol{\theta})$ and optimised via a sequence of smooth perturbations:

$$\boldsymbol{\theta}_{i,t+1} = \boldsymbol{\theta}_{i,t} + \eta_t \boldsymbol{\zeta}(\boldsymbol{\theta}_{i,t}) \ , \quad \boldsymbol{\zeta}(\boldsymbol{\theta}) = \frac{1}{M} \sum_{j=1}^{M} k(\boldsymbol{\theta}_{j,t}, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}_{j,t}} \log \hat{p}_n(\boldsymbol{\theta}_{j,t}) + \nabla_{\boldsymbol{\theta}_{j,t}} k(\boldsymbol{\theta}_{j,t}, \boldsymbol{\theta}) \ , \tag{5}$$

where $k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \boldsymbol{\phi}(\boldsymbol{\theta})^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{\theta}')$ corresponds to the SSGP kernel, and η_t is a small step size. Intuitively, the first term in the definition of $\boldsymbol{\zeta}$ guides the particles to the local maxima of $\log \hat{p}_n$, i.e. the modes of $\hat{p}_n$, while the second term encourages diversification by repelling nearby particles.

In contrast to the true posterior, the gradients of $\log \hat{p}_n$ are available as:

$$\nabla_{\boldsymbol{\theta}} \log \hat{p}_n(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} g_n(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) \ . \tag{6}$$

Gradients of sample functions are always defined for SSGP models with differentiable mean functions, since the feature maps are smooth. For a uniform prior, which we use in experiments, also note that $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) = 0$ almost everywhere.

2.3. Informative sampling via kernel herding

Having selected a distribution q_n, we need to run evaluations of $\Delta_{\boldsymbol{\theta}}$ from samples $\boldsymbol{\theta} \sim q_n$ to update the GP model with. Representing q by a large number of particles M improves

Algorithm 1: DBO

Input: f, $\mathcal{Q}$, N, S
for $n \in \{1, \dots N\}$ **do**
  $q_n \in \text{argmax}_{q \in \mathcal{Q}} h(q|\mathcal{D}_{n-1})$ # Maximise acquisition function via SVGD
  $\{\boldsymbol{\theta}_{n,i}\}_{i=1}^{S} \sim \text{Herding}(q_n, \mathcal{D}_{n-1})$ # Sample simulator parameters
  for $i \in \{1, \dots, S\}$ **do**
    $z_{n,i} := -\Delta_{\boldsymbol{\theta}_{n,i}}$ # Collect observation
  end
  $\mathcal{D}_n := \mathcal{D}_{n-1} \cup \{\boldsymbol{\theta}_{n,i}, z_{n,i}\}_{i=1}^{S}$
end

exploration of the approximate posterior surface, allowing SVGD to find distant modes. However, we should not use the large number of particles directly as sample parameters to run the simulator with, since simulations are expensive. Therefore, we select $S \ll M$ query parameters $\{\boldsymbol{\theta}_{n,j}\}_{j=1}^{S} \subset \Theta$ by optimally subsampling the candidate q_n.

Kernel herding (Chen et al., 2010) constructs a set of samples which minimises the error on empirical estimates for expectations under a given distribution q. This error is bounded by the maximum mean discrepancy (MMD) between the kernel embedding of q and its subsampled version (Muandet et al., 2016). In the case of SSGPs, the kernel herding procedure resumes to the following algorithm:

$$\boldsymbol{\theta}_{j+1} = \underset{\boldsymbol{\theta} \in \Theta}{\text{argmax}}\, \boldsymbol{\alpha}_j^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{\theta}) \tag{7}$$

$$\boldsymbol{\alpha}_{j+1} = \boldsymbol{\alpha}_j + \boldsymbol{\psi}_q - \boldsymbol{\phi}(\boldsymbol{\theta}_{j+1}) \ , \tag{8}$$

for $j \in \{0, \dots, S-1\}$ and $\boldsymbol{\alpha}_0 = \boldsymbol{\psi}_q = \mathbb{E}_{\boldsymbol{\theta} \sim q}[\boldsymbol{\phi}(\boldsymbol{\theta})]$. However, instead of naively herding with the original feature map $\boldsymbol{\phi}$, we make use of the information encoded by the GP to select samples which will be the most informative for the model. Such information is encoded by the GP posterior kernel:

$$k_n(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_\epsilon^2 \boldsymbol{\phi}(\boldsymbol{\theta})^{\mathsf{T}} \mathbf{A}_n^{-1} \boldsymbol{\phi}(\boldsymbol{\theta}) \ , \tag{9}$$

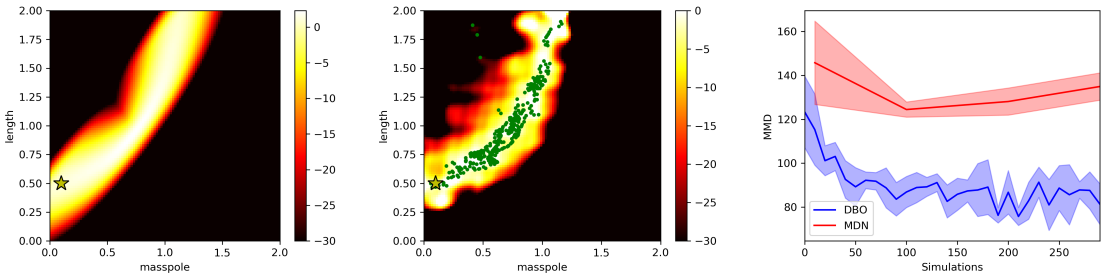
where $\sigma_\epsilon^2 \mathbf{A}_N^{-1}$ is the covariance matrix of the GP weights posterior (defined in Appendix A.2). The posterior kernel provides an embedding for q given by:

$$\boldsymbol{\psi}_q^n = \sigma_\epsilon^2 \mathbf{A}_n^{-1} \boldsymbol{\psi}_q \ , \tag{10}$$

which accounts for the previously observed locations in the GP data. Replacing $\boldsymbol{\psi}_q$ by $\boldsymbol{\psi}_q^n$ in Equation 8 yields the sampling scheme we use. The distributional Bayesian optimisation (DBO) algorithm is summarised in Algorithm 1.

3. Experiment

In this section, we present experimental results evaluating DBO in synthetic data scenarios. As a baseline we compare the method against mixture density networks (MDNs), as in Ramos et al. (2019), which were learnt from a dataset of parameters sampled from the prior $p(\boldsymbol{\theta})$ and the corresponding simulator outputs $\hat{\mathbf{y}}_{\boldsymbol{\theta}}$.



(*a*) True posterior (*b*) DBO's result (*c*) MMD

Figure 1: On the left, we show the reference Cart-Pole log-posterior density estimate obtained by ABC and interpolated by kernel density estimation. The parameters of the target system are indicated by a yellow star. The middle plot shows particles from DBO's learnt distribution on top of the final GP mean discrepancies estimate. The plot on the right shows a comparison against MDNs in terms of MMD with respect to the reference posterior as a function of the number of simulations.

The experiment evaluates the proposed method on OpenAI Gym's[3] cart-pole environment. We fix a given setting for its physics parameters $\boldsymbol{\theta}^{\text{real}}$ and generate a dataset $\mathbf{y}$ of 10 trajectories by executing randomly sampled actions. Summary statistics $\boldsymbol{\gamma}$ were the same as Ramos et al. (2019). The discrepancy was set to $\Delta_{\boldsymbol{\theta}} := \|\boldsymbol{\gamma}_{\boldsymbol{\theta}} - \boldsymbol{\gamma}^{\text{real}}\|^2/\sigma^2$. We place a uniform prior $p(\boldsymbol{\theta})$ with bounds specific for the environment. Further details on the experimental setup are described in Appendix B. An open-source implementation can be found online [4].

The results in Figure 1 show that the mehtod is able to recover the target system's curve-shaped posterior and is able to obtain better approximations to the posterior when compared to the MDN approach. We can also see that in terms of MMD, DBO is able to provide a better overall approximation than the MDN.

4. Conclusion

This paper presented a Bayesian optimisation approach to inverse problems on simulator parameters. Preliminary results demonstrated the potential of the method for reinforcement learning applications. In particular, results show that distributional Bayesian optimisation is able to provide a more sample-efficient approach than other likelihood-free inference methods when inferring parameters of a classical reinforcement learning environment. Future work includes further scalability and theoretical analysis of the method.

3. OpenAI Gym: https://gym.openai.com
4. Code available at: https://github.com/rafaol/dbo-aabi2019

References

Yutian Chen, Max Welling, and Alex Smola. Super-Samples from Kernel Herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI)*, Catalina Island, CA, 2010. AUAI Press.

Sayak Ray Chowdhury and Aditya Gopalan. On Kernelized Multi-armed Bandits. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, 2017.

Arjan Gijsberts and Giorgio Metta. Real-time model learning using Incremental Sparse Spectrum Gaussian Process Regression. *Neural Networks*, 41:59–69, 2013.

Michael U. Gutmann and Jukka Corander. Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models. *Journal of Machine Learning Research*, 17:1–47, 2016.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabas Poczos. Asynchronous Parallel Bayesian Optimisation via Thompson Sampling. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, Lanzarote, Spain, 2018.

Diederik P Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R. Figueiras-Vidal. Sparse Spectrum Gaussian Process Regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Advances in Neural Information Processing Systems*, (Nips):2378–2386, 2016.

Qiang Liu, Jason D. Lee, and Michael I. Jordan. A Kernelized Stein Discrepancy for Goodness-of-fit Tests. In *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, 2016.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel Mean Embedding of Distributions: A Review and Beyond. *arXiv*, 2016.

Mojmír Mutný and Andreas Krause. Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features. In *Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2018.

George Papamakarios and Iain Murray. Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation. In *Neural Information Processing Systems (NIPS)*, 2016.

George Papamakarios, David C. Sterratt, and Iain Murray. Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR: v. 89, 2019.

Ali Rahimi and Ben Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing (NIPS)*, 2007.

Fabio Ramos, Rafael Carvalhaes Possas, and Dieter Fox. BayesSim : adaptive domain randomization via probabilistic inference for robotics simulators. In *Robotics: Science and Systems (RSS)*, Freiburg im Breisgau, Germany, 2019.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006.

Christian P. Robert. Approximate Bayesian Computation: A Survey on Recent Results. In *Monte Carlo and Quasi-Monte Carlo Methods*, volume 163. Springer, Cham, 2016.

Daniel Russo and Benjamin Van Roy. An Information-Theoretic Analysis of Thompson Sampling. *Journal of Machine Learning Research (JMLR)*, 17:1–30, 2016.

Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, Cambridge, Mass, 2002.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Appendix A. Background

A.1. Kernelised Stein discrepancy

For a positive-definite kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, the kernelised Stein discrepancy is given by:

$$\mathbb{S}(q, p) = \|\mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbf{S}_p k(\cdot, \boldsymbol{\theta})]\|_{\mathcal{H}_k^d} \ , \tag{11}$$

where $\mathbf{S}_p$ represents the Stein operator for p and $\mathcal{H}_k$ denotes the reproducing kernel Hilbert space (Schölkopf and Smola, 2002) associated with k. For any $f \in \mathcal{H}_k$, we have:

$$\mathbf{S}_p f(\boldsymbol{\theta}) = f(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \ . \tag{12}$$

Similar to other measures of divergence between distributions, we have that:

$$\mathbb{S}(q, p) = 0 \iff q = p \ , \tag{13}$$

for strictly positive-definite kernels (Liu et al., 2016).

Liu and Wang (2016) present Stein variational gradient descent (SVGD) as an approach to minimise the Kullback-Leibler (KL) divergence between q and p based on the definition of the KSD. For a class $\mathcal{Q}$ of empirical distributions obtained by a sequence of smooth transforms, the vector $\boldsymbol{\zeta}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}\sim q}[\mathbf{S}_p k(\cdot, \boldsymbol{\theta})]$ corresponds to the optimal perturbation to a particle $\boldsymbol{\theta}$ in the support of $q \in \mathcal{Q}$.

A.2. Sparse-spectrum Gaussian processes

Gaussian process (GP) models (Rasmussen and Williams, 2006) provide a flexible approach to perform Bayesian inference over non-linear functions $g : \Theta \to \mathbb{R}$, $\Theta \subset \mathbb{R}^d$. One of the main issues with conventional GP models, however, is the high computational cost of $\mathcal{O}(N^3)$ in the inference process (Rasmussen and Williams, 2006), which can be a major drawback, especially in the case of online learning with Bayesian optimisation. For this reason, we make use of sparse spectrum Gaussian process (SSGP) regression (Lázaro-Gredilla et al., 2010), which allows for fast incremental updates of the GP posterior (Gijsberts and Metta, 2013).

The main idea behind sparse spectrum GP models is the decomposition of the GP covariance, or kernel, function $k : \Theta \times \Theta \to \mathbb{R}$ via Fourier series. As shown by Rahimi and Recht (2007), the Fourier transform of any shift-invariant kernel k on $\mathbb{R}^d$ yields a valid probability distribution P_k, so that k can be approximated as:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathbb{E}_{\boldsymbol{\omega} \sim P_k}[\cos(\boldsymbol{\omega}^\mathsf{T}(\boldsymbol{\theta} - \boldsymbol{\theta}'))] \approx \boldsymbol{\phi}(\boldsymbol{\theta})^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{\theta}'), \tag{14}$$

where:

$$\boldsymbol{\phi}(\boldsymbol{\theta}) := \begin{bmatrix} \boldsymbol{\phi}^c(\boldsymbol{\theta}) \\ \boldsymbol{\phi}^s(\boldsymbol{\theta}) \end{bmatrix}, \tag{15}$$

$\boldsymbol{\phi}^c(\boldsymbol{\theta}) := [\sigma_k \cos(\boldsymbol{\omega}_i^\mathsf{T}\boldsymbol{\theta})]_{i=1}^M$, $\boldsymbol{\phi}^s(\boldsymbol{\theta}) := [\sigma_k \sin(\boldsymbol{\omega}_i^\mathsf{T}\boldsymbol{\theta})]_{i=1}^M$, $\boldsymbol{\omega}_i \sim P_k$, and $\sigma_k := \frac{1}{\sqrt{M}}$. Linear scaling of the kernel by some $\sigma_g > 0$ can further be achieved by setting $\sigma_k := \frac{\sigma_g}{\sqrt{M}}$, instead. With this approximation, given a set of observations $\mathcal{D}_N = \{\boldsymbol{\theta}_i, z_i\}_{i=1}^N$, we can represent any function g sampled from the SSGP posterior as $g(\boldsymbol{\theta}) = \mu_0(\boldsymbol{\theta}) + \mathbf{w}^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{\theta})$, $\boldsymbol{\theta} \in \Theta$, where:

$$\mathbf{w}|\mathcal{D}_N \sim \mathcal{N}(\mathbf{A}_N^{-1}\mathbf{b}_N, \sigma_\epsilon^2 \mathbf{A}_N^{-1}), \tag{16}$$

$$\mathbf{b}_N = \boldsymbol{\Phi}_N(\mathbf{z} - \boldsymbol{\mu}_\mathbf{0}), \tag{17}$$

$$\mathbf{A}_N = \boldsymbol{\Phi}_N \boldsymbol{\Phi}_N^\mathsf{T} + \sigma_\epsilon^2 \mathbf{I}, \tag{18}$$

with $\boldsymbol{\Phi}_N = [\boldsymbol{\phi}(\boldsymbol{\theta}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{\theta}_N)] \in \mathbb{R}^{2M \times N}$. The posterior over g is then determined by:

$$g(\boldsymbol{\theta})|\mathcal{D}_N \sim \mathcal{N}(\mu_N(\boldsymbol{\theta}), \sigma_N^2(\boldsymbol{\theta})) \tag{19}$$

$$\mu_N(\boldsymbol{\theta}) := \mu_0(\boldsymbol{\theta}) + \boldsymbol{\phi}(\boldsymbol{\theta})^\mathsf{T} \mathbf{A}_N^{-1} \mathbf{b}_N \tag{20}$$

$$\sigma_N^2(\boldsymbol{\theta}) := \sigma_\epsilon^2 \boldsymbol{\phi}(\boldsymbol{\theta})^\mathsf{T} \mathbf{A}_N^{-1} \boldsymbol{\phi}(\boldsymbol{\theta}), \tag{21}$$

where μ_N and σ_N^2 denote the GP posterior mean and variance functions, respectively.

Fast incremental updates: To reduce the time complexity in the update of the GP posterior when given a new observation pair $(\boldsymbol{\theta}_{N+1}, z_{N+1})$, Gijsberts and Metta (2013) propose using the decomposition:

$$\mathbf{b}_{N+1} = \mathbf{b}_N + \boldsymbol{\phi}(\boldsymbol{\theta}_{N+1})(z_{N+1} - \mu_0(\boldsymbol{\theta}_{N+1})), \tag{22}$$

$$\mathbf{A}_{N+1} = \mathbf{A}_N + \boldsymbol{\phi}(\boldsymbol{\theta}_{N+1})\boldsymbol{\phi}(\boldsymbol{\theta}_{N+1})^\mathsf{T}. \tag{23}$$

To avoid recomputing $\mathbf{A}_{N+1}^{-1}$, one can instead keep track of its Cholesky factors. The latter allows us to update the GP posterior with time complexity $\mathcal{O}(M^2)$ (Gijsberts and Metta, 2013), which is constant with respect to the number of data points N.

Appendix B. Experiment details

Our GP model was configured with the Matérn kernel set with smoothness parameter $\nu := \frac{3}{2}$ (Rasmussen and Williams, 2006). Other hyper-parameters include the kernel length-scales, signal variance and noise variance, and were learnt as MAP estimates using previous BO runs where the hyper-parameters had been adapted online. The GP mean function was set as the log-prior probability $\log p(\boldsymbol{\theta})$. For the experimental results presented in the paper, the hyper-parameters are fixed.

Stein variational gradient descent was run with the sparse-spectrum kernel $k(\boldsymbol{\theta}, \boldsymbol{\theta}') := \boldsymbol{\phi}(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{\phi}(\boldsymbol{\theta}')$, $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, for 1000 steps with 1000 particles using Adam (Kingma and Ba, 2015) at a learning rate of 0.01.

The mixture density networks were configured with 10 Gaussian components following the experimental setup of Ramos et al. (2019). Neural network architectures were fully connected multi-layer perceptrons with 2 hidden layers, each with 24 units.