Aushev, Alexander; Pesonen, Henri; Heinonen, Markus; Corander, Jukka; Kaski, Samuel

Likelihood-Free Inference with Deep Gaussian Processes

Published in:
Computational Statistics & Data Analysis

DOI:
10.1016/j.csda.2022.107529

Published: 01/10/2022

Document Version
Publisher's PDF, also known as Version of record

Computational Statistics and Data Analysis 174 (2022) 107529

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

www.elsevier.com/locate/csda

Likelihood-free inference with deep Gaussian processes ☆

Alexander Aushev [a,*], Henri Pesonen [b], Markus Heinonen [a], Jukka Corander [b], Samuel Kaski [a,c]

[a] Department of Computer Science, Aalto University, Konemiehentie 2, 02150 Espoo, Finland
[b] Department of Biostatistics, University of Oslo, Oslo, Norway
[c] Department of Computer Science, University of Manchester, UK

ARTICLE INFO

Article history:
Received 23 August 2021
Received in revised form 29 April 2022
Accepted 12 May 2022
Available online 18 May 2022

Keywords:
Approximate Bayesian computation
Bayesian optimization
Gaussian processes
Deep learning

ABSTRACT

Surrogate models have been successfully used in likelihood-free inference to decrease the number of simulator evaluations. The current state-of-the-art performance for this task has been achieved by Bayesian Optimization with Gaussian Processes (GPs). While this combination works well for unimodal target distributions, it is restricting the flexibility and applicability of Bayesian Optimization for accelerating likelihood-free inference more generally. This problem is addressed by proposing a Deep Gaussian Process (DGP) surrogate model that can handle more irregularly behaved target distributions. The experiments show how DGPs can outperform GPs on objective functions with multimodal distributions and maintain a comparable performance in unimodal cases. At the same time, DGPs generally require much fewer data to achieve the same level of performance as neural density and kernel mean embedding alternatives. This confirms that DGPs as surrogate models can extend the applicability of Bayesian Optimization for likelihood-free inference (BOLFI), while only adding computational overhead that remains negligible for computationally intensive simulators.

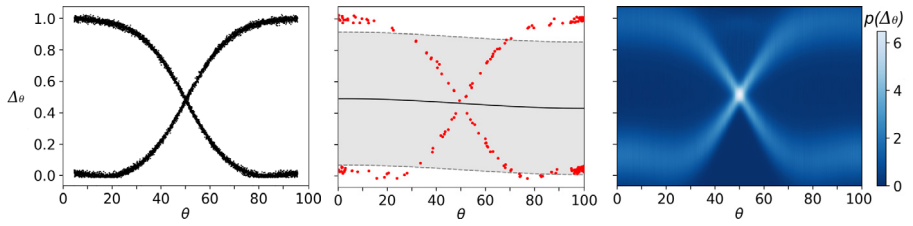
1. Introduction

Likelihood-free inference (LFI) for simulator-based models has been a topic of substantial interest during the past two decades for the computational modelling community (Hartig et al., 2011; Lintusaari et al., 2018). In LFI, we aim to infer the generative parameters $\boldsymbol{\theta}$ of an observed dataset $\mathbf{X}_{\text{obs}} = \{\mathbf{x}_{\text{obs}}\}$, whose likelihood $p(\mathbf{x}_{\text{obs}}|\boldsymbol{\theta})$ is intractable, which prevents the conventional statistical parameter estimation (Diggle and Gratton, 1984). Instead, we assume we can simulate new data $\{\mathbf{x}_\theta\} \sim p(\mathbf{x}|\boldsymbol{\theta})$ using any feasible parameter values. We relate the probability of a parameter to how similar its simulated dataset $\mathbf{X}_\theta$ is to the observed one (Hartig et al., 2011), measured via a discrepancy function. Different simulator-based LFI approaches have been proposed under the names of approximate Bayesian computation (ABC) (Beaumont et al., 2002, 2009; Csilléry et al., 2010; Sunnåker et al., 2013), indirect inference (Genton and Ronchetti, 2003; Gouriéroux et al., 2010; Heggland and Frigessi, 2004) and synthetic likelihood (Ong et al., 2018; Price et al., 2018; Wood, 2010) in domains ranging

☆ Additional experiments and implementation details of simulators can be found in the Supplement. All code is available through the link: https://github.com/AaltoPML/LFI-with-DGPs.

* Corresponding author at: Department of Computer Science, Aalto University, P.O.Box 15400, FI-00076 Aalto, Finland.
E-mail addresses: alexander.aushev@aalto.fi (A. Aushev), samuel.kaski@aalto.fi (S. Kaski).

https://doi.org/10.1016/j.csda.2022.107529



(a) True $\Delta_{\boldsymbol{\theta}}$ (b) Vanilla GP fit (c) DGP fit

Fig. 1. (a) An example of a multimodal target distribution: the discrepancy $\Delta_{\boldsymbol{\theta}}$ is bimodal for each value of the parameter $\boldsymbol{\theta}$. **(b)** Vanilla GP as a surrogate distribution is unable to fit the target (red: observed data; line and shading: GP prediction with uncertainty). **(c)** Deep GP surrogate is able to accurately model the bimodal target distribution. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

from genetics (Beaumont et al., 2002; Boitard et al., 2016; Pritchard et al., 1999) to economics (Guvenen and Smith, 2010; Monfardini, 1998) and ecology (Beaumont, 2010; van der Vaart et al., 2015; Wood, 2010).

When the simulator call-time is long, the number of simulator queries has to be limited for computational reasons. Therefore, a popular trend in LFI literature combines traditional methods with active learning (Rubens et al., 2015; Settles, 2009) to improve sample-efficiency. For instance, some neural-network-based density estimations (Alsing et al., 2019; Greenberg et al., 2019; Lueckmann et al., 2019; Papamakarios et al., 2019) and kernel mean embedding methods (Chowdhury et al., 2020; Hsu and Ramos, 2019) allow high-fidelity posterior inference with thousands of samples. A particularly suitable approach to LFI in this setting is to find a data-efficient surrogate to the discrepancy function, which can be used to derive a proxy for the unknown likelihood. Previous research by Gutmann and Corander (2016) has addressed this issue by using Gaussian Processes (GPs) as the discrepancy surrogates and applying Bayesian Optimization (BO) as an efficient search strategy. This approach drastically reduced the number of simulations required for accurate inference, to the order of only hundreds.

However, inferring simulator-based statistical models often requires approximating too complex distributions to be adequately represented by GPs, especially in the high-dimensional case. In particular, multimodal distributions (Franck and Koutsourelakis, 2017; Li et al., 2021; Shaw et al., 2007) still are a serious problem for the current LFI methods (Fig. 1). Sequential neural density estimation methods, based on Masked Autoregressive Flows (MAFs) and Mixture Density Networks (MDNs) (Papamakarios et al., 2017, 2019), use powerful deep network models to address this issue. However, to our knowledge, no current method is flexible enough to handle multimodal target distributions, unless given numerous samples (beyond hundreds) which would, however, be infeasible for computationally costly simulators. Our research hypothesis is that by adopting highly flexible Deep Gaussian Processes (DGPs) as surrogates in BO, we can simultaneously model both uni- and multimodal target distributions, and further cover also non-stationarity and heteroscedasticity.

In this paper, we propose three main contributions. Firstly, we solve the LFI problem for multimodal target distributions, with a limited number of function evaluations, which is important for computationally heavy simulators. Secondly, we propose quantile-based modifications for acquisition functions and likelihood approximation that are required for adopting Latent-Variable (LV) DGPs in BO for LFI. We provide a full computational complexity analysis for using LV-DGPs with these modifications. Thirdly, we give empirical evidence in several tasks, showing that the new surrogate model is able to handle well both uni- and multimodal targets, as well as non-stationarity and heteroscedasticity. Consequently, the new DGP-based surrogate has a greater application range than vanilla GPs for solving LFI problems. We also show that the method outperforms alternatives that are based on neural density estimation and kernel mean embedding.

In the following sections, we contextualize our work in LFI literature (Section 2), introduce elements of the proposed solution (Section 3), evaluate our method in simulated scenarios (Section 4) and finally discuss our findings (Section 5).

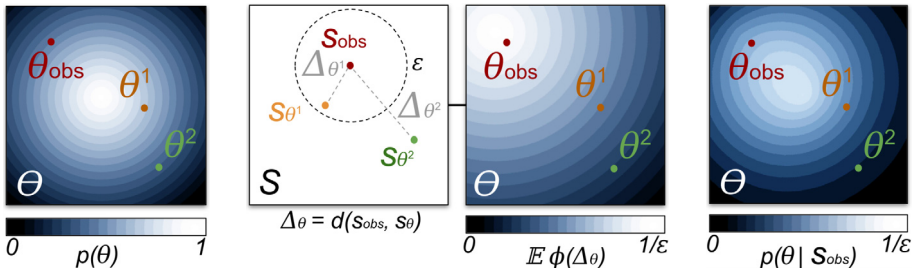
2. Likelihood-free inference

The general setting for LFI is illustrated in Fig. 2. In LFI, the target likelihood $p(\mathbf{X}_{\text{obs}}|\boldsymbol{\theta})$ of the observed data $\mathbf{X}_{\text{obs}}$ given estimated parameters $\boldsymbol{\theta}$ is implicitly modelled by a stochastic simulator, when its analytical form is unavailable. Our goal is to estimate the posterior distribution of $\boldsymbol{\theta}$ while only having the ability to draw simulated samples $\{\mathbf{x}_\theta\} \sim p(\mathbf{x}|\boldsymbol{\theta})$. This work follows the surrogate model approach (Gutmann and Corander, 2016) to LFI with BO (Shahriari et al., 2016).

2.1. Approximate Bayesian computation

Arguably the most popular approach which has been almost synonymous to LFI is ABC. In ABC, the inference of the unknown parameter value that generated $\mathbf{X}_{\text{obs}}$, is based on quantifying the discrepancy d between the summarized observed and synthetic datasets,

$$\Delta_{\boldsymbol{\theta}} := d\big[\underbrace{s(\mathbf{X}_{\text{obs}})}_{\mathbf{s}_{\text{obs}}}, \underbrace{s(\mathbf{X}_\theta)}_{\mathbf{s}_\theta}\big] \geq 0. \tag{1}$$



(a) Prior (b) Likelihood approximation through (3) (c) Posterior

Fig. 2. LFI estimates true parameters $\boldsymbol{\theta}_{\text{obs}}$ of the observed dataset $\mathbf{X}_{\text{obs}}$. (a) First, we assume a prior distribution over the parameter space. Parameter values, here $\boldsymbol{\theta}^1$ and $\boldsymbol{\theta}^2$, are selected to generate observations through a simulator that forms synthetic datasets $\mathbf{X}_{\theta^1}, \mathbf{X}_{\theta^2}$ (not in the figure). (b) Second, we replace the likelihood with an expectation over a kernel density (3). We use optional summarizing functions to transform datasets back to a single point of summary statistics $\mathbf{s}_{\theta^1}, \mathbf{s}_{\theta^2}$. The discrepancies between datasets $\Delta_{\boldsymbol{\theta}^1}$ and $\Delta_{\boldsymbol{\theta}^2}$ are measured (left) and used in a uniform kernel $\phi(\cdot)$ (4) to determine which parameters likely produced the observed dataset $\mathbf{X}_{\text{obs}}$. The expectation over uniform kernel density samples smooths the density surface, resulting in a likelihood approximation (right). (c) Third, we combine the prior and the likelihood approximation to compute the posterior $p(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}}) \approx p(\boldsymbol{\theta}|\mathbf{X}_{\text{obs}})$. In the context of computationally expensive simulators, considered in this paper, each $\boldsymbol{\theta}$ produces only one observation, and summaries serve as compact representations of observation points.

Here, $d(\cdot,\cdot)$ is a metric scalar distance (e.g. Euclidean distance) and $s(\cdot)$ is a summarizing function of the synthetic and observed datasets. After applying $s(\cdot)$, the resulting summary statistics are used to obtain a low-dimensional approximation of the likelihood,

$$p(\mathbf{X}_{\text{obs}}|\boldsymbol{\theta}) \approx p(\mathbf{s}_{\text{obs}}|\boldsymbol{\theta}), \tag{2}$$

which still inherits the intractability of the true likelihood. This summary likelihood is approximated by using the discrepancies (1) with a kernel density estimate $\phi(\Delta_{\boldsymbol{\theta}})$ (Fig. 2b) (Sisson et al., 2018),

$$p(\mathbf{s}_{\text{obs}}|\boldsymbol{\theta}) \approx \mathbb{E}_{\mathbf{X}_\theta \sim p(\mathbf{X}|\boldsymbol{\theta})}\left[\phi(\Delta_{\boldsymbol{\theta}})\right]. \tag{3}$$

A common choice for the kernel function $\phi(\cdot)$ is a uniform kernel (Sisson et al., 2018):

$$\phi(\Delta_{\boldsymbol{\theta}}) = \begin{cases} \frac{1}{\epsilon}, & \Delta_{\boldsymbol{\theta}} \in [0, \epsilon), \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where ϵ is the user-defined tolerance for the discrepancy. The kernel function $\phi(\cdot)$ quantifies the variability tolerance for simulated datasets, making the approximate likelihood in (3) proportional to the empirical probability of the discrepancy being below the threshold ϵ. Once the likelihood has been approximated, the Bayesian posterior over the parameter $\boldsymbol{\theta}$ can be inferred through (Fig. 2c)

$$p(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}}) \propto p(\mathbf{s}_{\text{obs}}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{5}$$

In this work, we use the approximate likelihood $p(\mathbf{s}_{\text{obs}}|\boldsymbol{\theta})$ in an importance-weighted sampling procedure to weight the posterior samples and calculate $p(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$. For a more detailed review of ABC methods and their recent advances, see (Lintusaari et al., 2017; Sisson et al., 2018).

2.2. Surrogate models in likelihood-free inference

Among the first surrogate model based solutions for the LFI problem were the synthetic likelihood approaches, where the simulator output is approximated with a Gaussian distribution. Wood (2010) generated several $\mathbf{x}_\theta$, or $\mathbf{s}_\theta$, at the parameter value $\boldsymbol{\theta}$, and then used them to estimate the mean and covariance of the Gaussian. Synthetic likelihoods can be formulated in the Bayesian framework (Price et al., 2018), which allows incorporating prior beliefs and updating them when new data are observed. GPs also lend themselves well to surrogate-modelling in LFI in multiple ways. Meeds and Welling (2014) used GPs as a surrogate for the proposal distribution in Markov Chain Monte Carlo ABC, and Gutmann and Corander (2016) modelled the discrepancy function as a function of the unknown parameters with a GP.

Sequential neural density estimators and kernel mean embedding methods are recent surrogate model approaches to the LFI problem. For instance, MAFs (Papamakarios et al., 2017) and MDNs (Papamakarios et al., 2019) use deep network models, resulting in accurate density estimations that have been suggested to require only $O(10^2) - O(10^3)$ samples for training. On the other hand, kernel mean embedding approaches tackle the problem of providing an embedding of the synthetic dataset to a reproducing kernel Hilbert space, removing the need of finding sufficient summary statistics (Nakagome et al., 2013) or automatically tuning the threshold ϵ parameter (Hsu and Ramos, 2019). These approaches are yet to be used in BO to improve sample-efficiency further, as suggested by Hsu and Ramos (2019); Nakagome et al. (2013); we provide the first

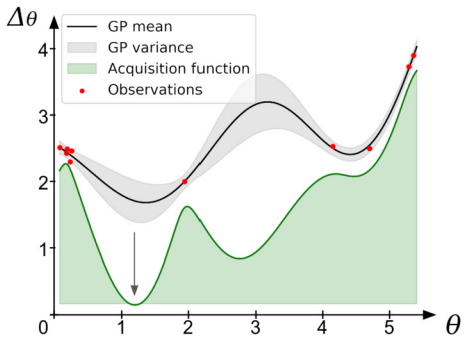


Fig. 3. GP fit with the mean (black) and variance (grey) for 12 observations (red) collected through the BO procedure. The objective function will be sampled next at the minimum of the acquisition function (green), which is marked with an arrow.

comparisons in Section 4. Our results show that both sequential neural density and kernel mean embedding approaches do not match the best methods with $O(10^2)$ sample sizes, and more research into their representativity versus training cost is still needed.

2.3. Bayesian optimization for likelihood-free inference

The task of approximating the likelihood $p(\mathbf{s}_{\text{obs}}|\boldsymbol{\theta})$ can be formulated as the problem of optimizing the expectation (3). Since the expectation (3) for the kernel (4) grows only when simulated datasets $\mathbf{X}_\theta$ produce discrepancy below the threshold ϵ, we need to find such $\boldsymbol{\theta}$, which minimizes the discrepancy. To solve this problem with as few sampled datasets as possible, we turn to BO, which has earlier been applied to LFI in a model called BOLFI (Gutmann and Corander, 2016).

BO requires a surrogate model for the objective and an acquisition function to guide the optimization process. In BOLFI, the discrepancy (1) is the objective, and it is approximated with a Gaussian process (GP) (Williams and Rasmussen, 2006) surrogate

$$\Delta_{\boldsymbol{\theta}} \sim GP(m(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{6}$$

which defines the prior mean and covariance of the discrepancy surface:

$$\mathbb{E}[\Delta_{\boldsymbol{\theta}}] = m(\boldsymbol{\theta}), \tag{7}$$

$$\text{cov}[\Delta_{\boldsymbol{\theta}}, \Delta'_{\boldsymbol{\theta}}] = k(\boldsymbol{\theta}, \boldsymbol{\theta}'). \tag{8}$$

The GP definition implies that any finite set of distances $\Delta_{\boldsymbol{\theta}}^{1:N} = \{\Delta_{\boldsymbol{\theta}}^n\}_{n=1}^N$ at alternative parameter values $\boldsymbol{\theta}^{1:N} = \{\boldsymbol{\theta}^n\}_{n=1}^N$ is jointly Gaussian:

$$p(\Delta_{\boldsymbol{\theta}}^{1:N}) = N(\Delta_{\boldsymbol{\theta}}^{1:N}|\mathbf{m}, \mathbf{K}), \tag{9}$$

where $\mathbf{m} = \{m(\boldsymbol{\theta}^n)\}_{n=1}^N \in \mathbb{R}^N$ and the kernel $\mathbf{K} \in \mathbb{R}^{N\times N}$ contains values $\mathbf{K}_{ij} = k(\boldsymbol{\theta}^i, \boldsymbol{\theta}^j)$. The commonly chosen RBF kernel induces smooth distance surfaces that allow efficient exploration.

BO chooses the point $\boldsymbol{\theta}^{t+1}$ where to next evaluate the objective function by minimizing an acquisition function $\text{A}^t(\cdot)$, such as lower confidence bound (Cox and John, 1992), at time t

$$\boldsymbol{\theta}^{t+1} = \arg\min_\theta \{\text{A}^t(\boldsymbol{\theta})\}, \tag{10}$$

$$\text{A}^t(\boldsymbol{\theta}) = m(\boldsymbol{\theta}) - \sqrt{\eta_t^2 \cdot v(\boldsymbol{\theta})}, \tag{11}$$

where η_t^2 is a user-defined tuning parameter (Srinivas et al., 2012) and $v(\boldsymbol{\theta}) := k(\boldsymbol{\theta}, \boldsymbol{\theta})$ is the GP variance. The acquisition function uses the mean and variance of the GP, and is usually chosen to make a trade-off between exploitation (minimization based on what is already known) and exploration (sampling in the regions of high uncertainty). See Fig. 3 for a demonstration.

Conventional BO works well for objectives with Gaussian uncertainties, but can be used with other surrogates as well. Some examples include: deep neural networks (Snoek et al., 2015) for objectives that require many evaluations, DGPs (Hebbal et al., 2020) for non-stationary objectives, student-t processes (Shah et al., 2013) for modelling heavy-tailed distributions, and decision trees (Jenatton et al., 2017) for modelling known dependency structures.

In this work, we bring BO to solve a so-far unsolved problem: likelihood-free inference for commonly occurring irregular distributions, in particular multimodal but also skewed distributions. This is especially difficult for computationally heavy simulators, for which we can afford only few evaluations, and hence, need to adopt surrogate functions that combine flexibility with data-efficiency.

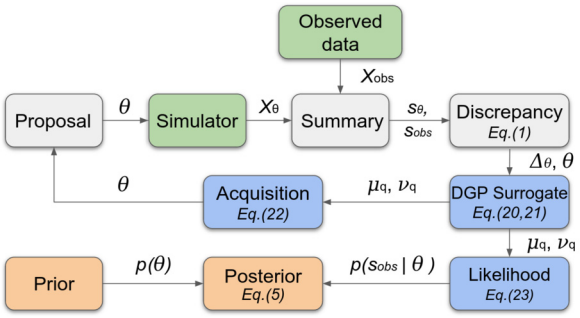


Fig. 4. Overview of the proposed LFI with DGPs approach to estimating parameter posterior $p(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$. Given an observed dataset and a simulator (green blocks), we follow the BO procedure with the introduced DGP surrogate changes (blue blocks). Each parameter proposed by the acquisition function is run through the simulator to obtain a single synthetic dataset $\mathbf{X}_\theta$. The discrepancy $\Delta_{\boldsymbol{\theta}}$ is then computed using the summaries $\mathbf{s}_{\text{obs}}$ and $\mathbf{s}_\theta$ for the observed and simulated datasets with (1), and coupled with the corresponding $\boldsymbol{\theta}$ to form the evidence for training the surrogate. Finally, the likelihood approximation $p(\mathbf{s}_{\text{obs}}|\boldsymbol{\theta})$ is extracted and used along with the prior $p(\boldsymbol{\theta})$ to infer the posterior $p(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$.

3. Bayesian optimization with deep Gaussian processes

BO uses a probabilistic surrogate to find the posterior distribution of the parameter $\boldsymbol{\theta}$. We propose to use DGP surrogates that are capable of handling multimodal and non-stationary discrepancy distributions (Section 3.1), along with quantile-based modifications for an acquisition function (Section 3.2) and likelihood approximation (Section 3.3) required for modelling such distributions in BO for LFI. In Section 3.4, we evaluate the computational overhead from the new surrogate. The general overview of the proposed approach is illustrated in Fig. 4.

3.1. Multimodal deep Gaussian processes

A DGP composes multiple GPs together for more flexible and powerful function representations (Damianou and Lawrence, 2013; Dunlop et al., 2018). These representations can have a non-Gaussian, multimodal distributional form. However, DGP posteriors do not have explicit analytical solutions as GPs, and require variational (Salimbeni and Deisenroth, 2017) or Monte Carlo (Havasi et al., 2018) approximations for inference.

The quality of the predictive posterior approximation largely depends on the chosen inference method, and most DGP models and inference methods are not able to yield multimodal marginals. Because typically in most DGPs the outputs are correlated, while multiple modes, which need to be approximated, are not. Such irregularly behaved distributions can be modelled only when DGP latent function values $\mathbf{f}$ do not correlate with each other for the same input (Salimbeni et al., 2019). In this work, we argue that it is important to use one of the possible solutions that guarantees this property. In the experiments, we have used Latent-Variable (LV) DGPs (Salimbeni et al., 2019), which augment the input vector $\mathbf{x}$ with latent variables $w \sim N(0,1)$ concatenated into $[\mathbf{x}, w] \in \mathbb{R}^{D+1}$, to be used as input for the next GP layer. By combining different LV and GP layer architectures with importance-weighted objectives, LV-DGPs provide more flexible DGP posterior approximations (Salimbeni et al., 2019).

The proposed method in this paper is not exclusive to the specific LV-DGP model, and can be used with any DGPs that are capable of approximating multimodal marginal distributions. Therefore, we refer to LV-DGPs when we discuss this specific architecture, and to DGPs, whenever the results apply to multimodal capable DGPs in general. The specific LV-DGP method we used in the experiments is as follows; here with a LV layer followed by two GPs and denoted by LV-2GP. The GP priors are

$$f(\mathbf{x}) \sim GP(m_f(\mathbf{x}), k_f(\mathbf{x}, \mathbf{x}')), \tag{12}$$

$$g(\mathbf{x}) \sim GP(m_g(\mathbf{x}), k_g(\mathbf{x}, \mathbf{x}')), \tag{13}$$

and with Gaussian likelihoods for modelling the discrepancy (1)

$$p(\Delta_{\boldsymbol{\theta}}|f, g, w) = N(\Delta_{\boldsymbol{\theta}}|f(g([\boldsymbol{\theta}, w])), \sigma^2), \tag{14}$$

$$p(\Delta_{\boldsymbol{\theta}}|f, g) = \mathbb{E}_{p(w)} N(\Delta_{\boldsymbol{\theta}}|f(g([\boldsymbol{\theta}, w])), \sigma^2). \tag{15}$$

We use the Importance-Weighted Variational Inference (IWVI) of Salimbeni et al. (2019) to minimize KL-divergence $\text{KL}[\chi(\mathbf{f}, \mathbf{g}, \mathbf{w})||p(\mathbf{f}, \mathbf{g}, \mathbf{w}|\Delta_{\boldsymbol{\theta}})]$, where $\chi(\mathbf{f}, \mathbf{g}, \mathbf{w})$ are the variational posterior approximations to be learned. The lower bound can be formulated as

$$\log p(\Delta_{\boldsymbol{\theta}}) \geq \mathbb{E}_{\chi(\mathbf{f},\mathbf{g},\mathbf{w})} \log p(\Delta_{\boldsymbol{\theta}}^{1:N}|\mathbf{f}, \mathbf{g}, \mathbf{w}) - \text{KL}\big(\chi(\mathbf{f}, \mathbf{g}, \mathbf{w})||p(\mathbf{f}, \mathbf{g}, \mathbf{w})\big), \tag{16}$$

where we assume factorized variational approximation and prior,

$$\chi(\mathbf{f},\mathbf{g},\mathbf{w}) = \chi(\mathbf{f})\chi(\mathbf{g})\chi(\mathbf{w}), \tag{17}$$

$$p(\mathbf{f},\mathbf{g},\mathbf{w}) = p(\mathbf{f})p(\mathbf{g})p(\mathbf{w}). \tag{18}$$

The variational approach of the latent variable further factorizes as

$$\chi(\mathbf{w}) = \prod_n N(w_n|a_n,b_n), \tag{19}$$

with a_n, b_n being variational parameters to be optimized. The variational approximations $\chi(\mathbf{f})$, $\chi(\mathbf{g})$ represent Gaussian process layers, for which we use the sparse inducing point approximation (Salimbeni et al., 2019). Later in the experiments, we use deeper architectures that use up to five GP layers. Similarly, as in the two-layer example, these additional layers require inference of corresponding KL terms to form a composite function in (14) and (15). Once we have a DGP predictive distribution, we can use it in BO.

3.2. Bayesian optimization with deep Gaussian processes

BO requires a surrogate model, an acquisition function $\mathrm{A}^t(\boldsymbol{\theta})$ and the ability to evaluate the black-box objective function. Here, we minimize the discrepancy $\Delta_{\boldsymbol{\theta}}$ as the objective and use the DGP probabilistic model in the acquisition function of the discrepancy $\Delta_{\boldsymbol{\theta}}$ from (15) to choose where to sample next.

In BO for LFI, BO uses GP predictive mean and variance in the acquisition function, as shown in (10). By design, the acquisition function focuses on accurate representation of the low-valued discrepancy regions. However, when the discrepancy is multimodal, the GP mean and variance tend to overestimate the expected discrepancy value and its uncertainty. As a result, multimodal and more promising regions can be overlooked by BO in favour of unimodal regions. The solution we propose to this problem is to accurately represent the low-valued regions of discrepancy, maintaining high signal-to-noise ratio.

We introduce quantile-conditioning on DGP predictive samples to estimate more accurately the lowest values of the discrepancies. The DGP is applied to the regression problem $\boldsymbol{\theta} \mapsto \Delta_{\boldsymbol{\theta}}$, resulting in estimates of mean $\mu_q(\boldsymbol{\theta})$ and variance $\nu_q(\boldsymbol{\theta})$ in the lowest quantile through a quantile function $Q(\cdot)$

$$\mu_q(\boldsymbol{\theta}) = \mathbb{E}\{\Delta^n_{\boldsymbol{\theta}} : \Delta^n_{\boldsymbol{\theta}} \leq Q(\epsilon_q)\}_{n=1}^N, \tag{20}$$

$$\nu_q(\boldsymbol{\theta}) = \mathrm{var}\{\Delta^n_{\boldsymbol{\theta}} : \Delta^n_{\boldsymbol{\theta}} \leq Q(\epsilon_q)\}_{n=1}^N. \tag{21}$$

By only considering discrepancies below a (user-defined) small quantile-threshold ϵ_q (called *quantile-conditioning* below), the estimator is able to focus on accurately representing the important low-valued regions of the discrepancy surface, as demonstrated in Fig. 5. We use these values in the acquisition function $\mathrm{A}^t(\boldsymbol{\theta})$ to get a new proposal for simulation, resulting in a simple quantile-based modification of the lower confidence bound selection criterion (LCBSC) (Cox and John, 1992) for selecting a new parameter point $\boldsymbol{\theta}^{t+1}$ at any current time t

$$\mathrm{A}^t(\boldsymbol{\theta}) = \mu_q(\boldsymbol{\theta}) - \sqrt{\eta_t^2 \cdot \nu_q(\boldsymbol{\theta})}. \tag{22}$$

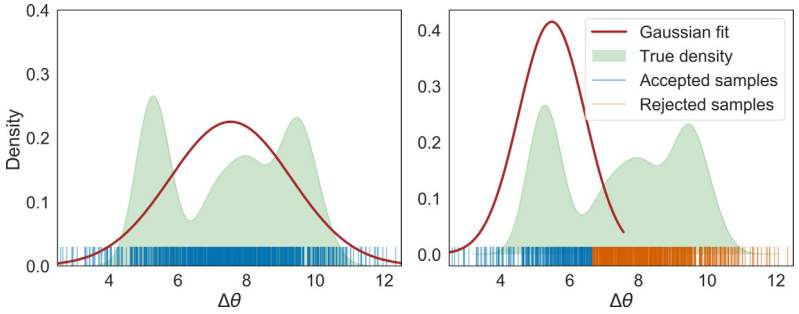
The proposed quantile-based acquisition maintains the advantages of the LCBSC, while also enabling BO with multimodal or skewed uncertainties.

3.3. Likelihood approximation

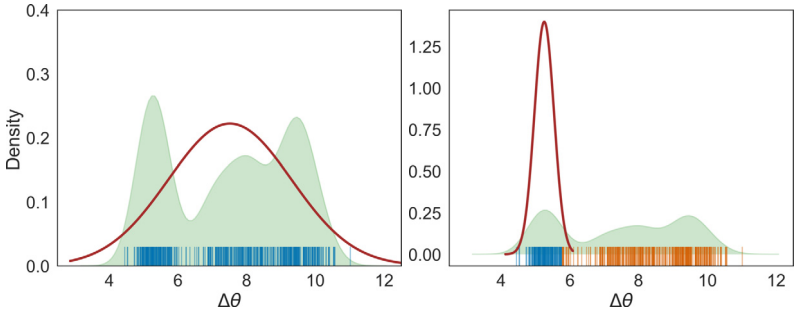
Lastly, we use the mean and the variance of DGP posterior samples below a quantile ϵ_q threshold to approximate the likelihood (3). Gutmann and Corander (2016) constructed the likelihood approximation from the GP model of the discrepancy using normal cumulative distribution function (cdf) with the discrepancy tolerance ϵ. This approach works well for unimodal distributions, where the mean and variance of the GP characterize the mode well, but for multimodal distributions individual modes are concealed when represented with the mean and variance of the whole distribution. Moreover, only the modes that correspond to low-valued discrepancy regions are likely to produce the observed dataset and, hence, should be considered in the likelihood approximation. Here, we filter out all DGP predictive posterior samples that are above quantile-threshold to focus on samples from the low-valued discrepancy regions:

$$p(\mathbf{s}_{\mathrm{obs}}|\boldsymbol{\theta}) \propto F\left(\frac{\epsilon - \mu_q(\boldsymbol{\theta})}{\sqrt{\nu_q(\boldsymbol{\theta}) + \sigma^2}}\right), \tag{23}$$

where $F(\cdot)$ is the cdf of Gaussian with mean 0 and variance 1, μ_q and ν_q are the mean and the variance of DGP posterior sample below the quantile-threshold ϵ_q, ϵ is a tolerance from (4) and σ^2 is the Gaussian likelihood noise from (15). The quantile-threshold ϵ_q depends on the signal-to-ratio noise in the simulator. For instance, $\epsilon_q = 0.3$ assumes that there is a



(a) All GP predictive samples (b) GP predictive samples after quantile-conditioning



(c) All DGP predictive samples (d) DGP predictive samples after quantile-conditioning

Fig. 5. Gaussian fit (red) before and after applying quantile-conditioning on GP and DGP predictive samples for the true density (green). GP predictive samples (blue) tend to overestimate the uncertainty of the true density (green) for the whole distribution (a) and for the lowest quantile (b) (with $\epsilon_q = 0.3$). At the same time, DGP predictive samples (blue) exhibit similar behaviour on the whole distribution (c), but with quantile-conditioning (d) result in a more accurate and narrow approximation of the low-valued discrepancy Δ_θ region, characterized as a Gaussian with the mean μ_q and variance ν_q. This can be seen by comparing how closely the Gaussian curve (red), built on top of samples below the quantile-threshold (blue), estimates the low-valued discrepancy region of the true density (green). Predictive samples that are above the quantile-threshold are marked with orange.

signal in the 0.3 quantile. Similarly, as in BOLFI, ϵ is set to the minimum value of the $\mu_q(\cdot)$, so as the number of predictive samples grows, the approximation in (23) becomes more accurate.

In summary, we have introduced a way for DGP surrogates to handle irregularly-behaved marginal distributions in the context of BO, by proposing a quantile-based likelihood approximation and acquisition rule.

3.4. Computational overhead

The computational overhead of having a more complex surrogate is negligible if the simulator is computationally expensive. DGPs, as a more flexible model, require more time for training and prediction, compared to GPs. There are three major stages of the BOLFI Algorithm 1, where the surrogate plays a role: initialization, BO updates and posterior extraction. In this section, we analyse the increase in time complexity caused by switching to multimodal capable LV-DGPs instead of traditionally-used GPs at every stage. We consider a LV-DGP architecture with l GP layers, introduced in Section 3.1, and sparse approximations of GPs in our analysis.

Initialization. At this stage of the algorithm, the simulator creates initial observations and trains the surrogate model. Sparse GPs require $O(m^2 n d_\theta)$ cost for inference, and $O(m d_\theta)$ and $O(m^2 d_\theta)$ for predicting the mean and variance respectively. Here, m is the number of inducing points, n is the number of initial data, and d_θ is the dimensionality of a parameter vector. LV-DGPs, on the other hand, are using sample average of k terms (importance-weight samples) to replace the latent variable layer, resulting in $O(lkm^2 n d_\theta)$ for training, and $O(lkm d_\theta)$ and $O(lkm^2 d_\theta)$ for the mean and variance prediction.

BO. Once the surrogate model has been trained, the BO procedure starts. It consists of minimization of the acquisition function, simulation of data, and optional surrogate model hyperparameter optimization or retraining (in our implementation, we do this last step at the final stage). Again, when the simulator is fast, the acquisition function minimization becomes the computational bottleneck of this stage. We used L-BFGS-B optimization (Zhu et al., 1997) for finding a minimum of the acquisition function (22) with the cost $O(t d_\theta A i)$, where t is the number of steps stored in memory by parameter declaration (the limited memory BFGS method does not store the full hessian but uses this many terms in an approximation to it), i is

Algorithm 1: BOLFI algorithm with DGPs.

Data: Datasets $\mathbf{x}_{\text{obs}}$, N initial simulation budget, S BO simulation budget
Result: Posterior $p(\boldsymbol{\theta}|\mathbf{x}_{\text{obs}})$
sample N times from the prior $\boldsymbol{\theta}^{1:N} \sim p(\boldsymbol{\theta})$;
simulate synthetic datasets $\mathbf{X}_{\theta}^{1:N} = g(\boldsymbol{\theta}^{1:N})$;
compute discrepancies $\Delta_{\boldsymbol{\theta}}^{1:N}$, Eq (1);
initialize DGP as $\boldsymbol{\theta}^{1:N} \mapsto \Delta_{\boldsymbol{\theta}}^{1:N}$;
train DGP with pairs $\{\boldsymbol{\theta}^{1:N}, \Delta_{\boldsymbol{\theta}}^{1:N}\}$;
starting BO procedure;
while *current simulation budget* $< S$ **do**
  acquire new $\boldsymbol{\theta}'$, Eq (22);
  simulate new datasets $\mathbf{x}'_{\theta} = g(\boldsymbol{\theta}')$;
  compute discrepancy $\Delta'_{\boldsymbol{\theta}}$, Eq (1) ;
  augment DGP data with $\{\boldsymbol{\theta}', \Delta'_{\boldsymbol{\theta}}\}$;
end
retrain DGP;
extract result (find $\mu_q(\cdot)$ minimum value);
extract posterior $p(\boldsymbol{\theta}|\mathbf{x}_{\text{obs}})$, Eq (23);

the number of initialization points and A is the cost of the acquisition function call. Further decomposition of the acquisition function complexity requires computation of DGP mean and variance, bounded by $O(lkm^2 d_\theta)$ cost, corresponding to the DGP predictive variance, or $O(p^2)$, corresponding to calculation of the quantile-conditioning based on p predictive samples. Finally, when the acquisition function minimum is found, the points in the batch are calculated by adding an acquisition noise to its value.

Posterior extraction. As the final stage, the posterior is extracted from the trained surrogate. This is performed by sampling the prior S times, setting the threshold ϵ (as the minimum of either $\mu(\cdot)$ or $\mu_q(\cdot)$), and then reweighing the samples by using (23). The prior sampling and importance weighted resampling are less computationally intensive than finding the threshold and calculating the weights, where DGP prediction plays an important role. The threshold is found by minimizing the DGP mean function with L-BFGS-B minimization that requires $O(td_\theta \mu i)$ cost, where μ is the cost of the DGP mean function $O(lkmd_\theta)$. And applying Equation (23) has complexity of $O(lkm^2 d_\theta S)$ or $O(p^2)$, since it requires calculating DGP predictive mean and variance for S samples conditioned on the quantile-threshold ϵ_q.

In summary, the increase in complexity from switching GPs to LV-2GPs is $O(lk)$ times for all three stages. In practice, k is a relatively small number ranging from 5 to 20 and l is from 2 to 5. Additionally, the cost of calculating the quantile-threshold should rarely exceed the cost of calculating predictive mean and variance with predictive samples, ranging from 10 to 100.

4. Experiments

We study the merits of DGP surrogates in BO, first in illustrative demonstrations and then in four case studies. Our main goal is to reduce the number of required simulator evaluations, which is important for computationally intensive simulators. Section 4.1 describes the experimental setup, simulators and comparison methods that were used. In Section 4.2, we consider the simplest deep LV-2GP architecture of the LV-DGP model and analyse its advantages against traditionally used GPs. In Section 4.3 we compare LV-DGP architectures with neural density estimation and kernel mean embedding approaches. The results of the experiments are summarized in Table 2 with details of the findings discussed in individual sections afterwards.

4.1. Experimental setup

In each simulation experiment, we select true parameter values, and use them to produce the observed data set with the simulator. Each experiment is repeated 100 times, the runs differing in the choice of random seeds that affect the observations used as initial evidence. We limit the number of total simulator calls to 200 with 100 initial evidence points drawn from the prior before the active learning procedure starts; when targeting computationally heavy simulators this is already plenty. In Section 4.2 we also study how the performance of the GP and DGP surrogates changes with fewer observations and initial evidence, where a half of all observations come from initial evidence points.

When evaluating goodness of the posterior approximations of $\boldsymbol{\theta}$, we estimate the ground-truth posterior numerically by the reference-table acceptance-rejection ABC (Cornuet et al., 2008) with 10^8 simulations, selecting 0.001% samples with the lowest discrepancy (ϵ is calculated to guarantee the exact amount of samples) to represent the posterior distribution. Closeness of the estimated posterior $p_{\text{sur}}(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$ to this ground-truth reference posterior $p_{\text{ref}}(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$ is measured with the empirical Wasserstein distance (Bernton et al., 2019; Genevay et al., 2016) that shows similarity of the estimated surrogate posterior to the ground-truth, defined as

$$W_D(y_{1:n}, z_{1:n}) = \inf_{\sigma \in \mathcal{S}_n} \frac{1}{n} \sum_{i=1}^{n} ||y_i - z_{\sigma(i)}||_2, \tag{24}$$

where $\mathcal{S}_n$ is the set of permutations of $\{1, ..., n\}$, $y_i \sim p_{\text{ref}}(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$ and $z_i \sim p_{\text{sur}}(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$. Along with the Wasserstein distance, we also report the computation time of all experiments in the Supplement.

4.1.1. Simulator descriptions

In our experiments, we used eight different simulators, four of which are toy examples (TEs) and four are case studies. The toy models were designed to demonstrate specific properties of the surrogate model: non-stationarity, multimodality and heteroscedasticity. The case studies represent more difficult problems that often occur in practice. They have multi-dimensional parameters and cover both unimodal and multimodal cases. A more detailed description of each simulator is provided below.

Demonstrations: non-stationarity, multimodality and heteroscedasticity. The discrepancy function of the first demonstrator TE1 is non-stationary with the ground-truth $\theta_{\text{obs}} = 50$. The simulator function $g_{\text{TE1}}(\theta)$ generates data from the sum of three Gaussian density functions with different means and variances,

$$g_{\text{TE1}}(\theta) = N(\theta|30, 15) + N(\theta|60, 5) + N(\theta|100, 4) + \omega, \quad \omega \sim N(0, 0.005). \tag{25}$$

The second example, TE2, has a multimodal discrepancy function with the ground-truth $\theta_{\text{obs}} = 20$. The simulator function g_{TE2} randomly 'chooses' one of two logistic functions, and generates the observation according to

$$g_{\text{TE2}}(\theta') = \omega_1 \cdot \frac{\theta'}{1+\theta'} + (1-\omega_1) \cdot \frac{1}{1+\theta'} + \omega_2, \tag{26}$$

where $\theta' = \exp(-0.1(\theta - 50))$, $\omega_1 \sim \text{Bernoulli}(0.5)$ and $\omega_2 \sim N(0, 0.01)$. The simulator function creates several modes in the observation space, that later transfer to the discrepancy function.

The discrepancy function of the third demonstrator TE3 is heteroscedastic with the ground-truth $\theta_{\text{obs}} = 20$. The output of the simulator is generated as a sum of samples from two different beta distributions that are defined through the input parameter θ. The sum of two random variables defined on the interval $[0, 1]$ were used as a simulator:

$$g_{\text{TE3}}(\theta) = \omega_1 + \omega_2, \quad \omega_1 \sim \text{Beta}(\theta + 1, 5), \quad \omega_2 \sim \text{Beta}(5, \theta + 1). \tag{27}$$

Finally, the fourth demonstrator TE4 exhibits both multimodality and non-stationarity in its discrepancy function. In this example, the posterior of parameters has three modes with different peak levels. The simulator data comes from one of two functions with added uniform noise

$$g_{\text{TE4}}(\theta) = 100 \cdot \omega_1 \cdot N(\theta \mid 0, 50) + (1-\omega_1) \cdot N(\theta \mid 60, 55) + \omega_2, \tag{28}$$

where $\omega_1 \sim \text{Bernoulli}(0.4)$, $\omega_2 \sim \text{Unif}(0, 10^{-4})$ and $\theta_{\text{obs}} = 60$.

In all TEs, a uniform prior on the interval $(0, 100)$ was used for simulator parameters, with the Euclidean distance calculated directly on observations, since they have a single dimension and summary statistics are not needed.

Birth-Death model. The Birth-Death model (BDM) describes tuberculosis transmission in the San Francisco Bay Area, as formulated by Tanaka et al. (2006). Given epidemiological parameters $\theta_{R_1}, \theta_{R_2}, \theta_\beta, \theta_{t_1}$, the model simulates tuberculosis outbreak dynamics in a population and outputs cluster indexes of observed transmission cases. Our goal is to approximate the posterior distribution $P(\theta_{R_1}, \theta_{R_2}, \theta_\beta, \theta_{t_1}|\mathbf{x}_{\text{obs}})$, where $\mathbf{x}_{\text{obs}}$ was generated with the vector of ground-truth parameters (5.88, 0.09, 192, 6.74). These ground-truth values were inferred by Lintusaari et al. (2019) from the summaries of real data (Small et al., 1994).

We used the weighted Euclidean distance as the discrepancy measure with the summaries and the corresponding distance weights shown in Table 1. The same hierarchical priors as in Lintusaari et al. (2019) were used:

$$\begin{aligned} \theta_{\text{burden}} &\sim N(200, 30), \quad \theta_{R_1} \sim \text{Unif}(1.01, 20), \\ \theta_{R_2}|\theta_{R_1} &\sim \text{Unif}(1.01, (1 - 0.05 \cdot \theta_{R_1})/0.95), \\ \theta_{t_1} &\sim \text{Unif}(0.01, 30). \end{aligned}$$

For detailed interpretation of simulator parameters and summaries, see Lintusaari et al. (2019). The implementation of the simulator model can be found in the Supplement.

Sound localization. In the sound localization (SL) model (Forbes et al., 2021), there are two different pairs of microphones, which are randomly chosen for detecting the sound source in a 2D scene. The positions of the microphones are set at locations (-0.5, 0), (0.5, 0) for the first pair, and (0, -0.5), (0, 0.5) for the second pair, while the position of the sound source

Table 1
Summaries for the BDM case and their weights in the discrepancy function (Lintusaari et al., 2019).

Summary statistics	Weight
Number of observations	1
Number of clusters	1
Relative number of singleton clusters	100/0.60
Relative number of clusters of size 2	100/0.4
Size of the largest cluster	2
Mean of the successive differences in size among the four largest clusters	10
Number of months from the first observation to the last in the largest cluster	10
The number of months in which at least one observation was made from the largest cluster	10

$\boldsymbol{\theta} = (\theta_x, \theta_y)$ is unknown. Our goal is to identify the location of the sound source by simulating the interaural time difference (ITD) (Wang and Brown, 2006) based on the randomly chosen pair of microphones $(\mathbf{m}_1, \mathbf{m}_2)$

$$\mathrm{ITD}(\boldsymbol{\theta}) = |\,||\boldsymbol{\theta} - \mathbf{m}_1||_2 - ||\boldsymbol{\theta} - \mathbf{m}_2||_2\,|, \tag{29}$$

$$f_{\boldsymbol{\theta}}(\mathbf{y}) = \mathcal{S}_d(\mathbf{y}; \mathrm{ITD}(\boldsymbol{\theta}) \cdot \mathbb{I}_d, \sigma^2 \cdot \mathrm{I}_d, \nu), \tag{30}$$

where $\mathcal{S}_d$ is the d-variate Student t-distribution with a d-dimensional location all equal to $\mathrm{ITD}(\boldsymbol{\theta})$, the scale factor for the diagonal matrix $\sigma^2 = 0.01$ and degree-of-freedom $\nu = 3$. We use Euclidean distance with the mean and the standard deviation of the $\mathbf{y} = (y_1, ..., y_d)$. The uniform priors were assumed on the interval $[-2, 2]$ for both parameters.

Cosmological inflation. The Cosmological inflation (CI) model, as proposed by Sinha and Souradeep (2006), predicts the power of a cosmological spectrum $\mathcal{P}$, based on its physical properties: the governing factor of exponential decay θ_α, the variance of the fluctuations θ_{A_s}, the infrared cut-off θ_{k_c}, the scalar spectral index θ_{n_s}, and the ratio of inflationary potential θ_{R^*}. The simulation is defined through the transfer function $T^2(y, \theta_{R^*})$, where $y = \theta_{k_c}/k_*$ and $k_* \sim \mathrm{Unif}(8 \cdot 10^{-4}, 85 \cdot 10^{-5})$,

$$T^2(y, \theta_{R^*}) = 1 - 3 \cdot (\theta_{R^*} - 1) \cdot \frac{1}{y} \cdot \left[\left(1 - \frac{1}{y^2}\right) \cdot \sin 2y + \frac{2}{y} \cdot \cos 2y\right] + \frac{9}{2} \cdot \theta_{R^*}^2 \cdot \frac{1}{y^2} \cdot \left(1 + \frac{1}{y^2}\right) \cdot \left[1 + \frac{1}{y^2} + \left(1 - \frac{1}{y^2}\right) \cdot \cos 2y - \frac{2}{y} \cdot \sin 2y\right], \tag{31}$$

$$\mathcal{P} = \theta_{A_s} \cdot (1 - e^{(0.75y)^{\theta_\alpha}}) \cdot \theta_{k_c}^{\theta_{n_s} - 1} \cdot T^2(y, \theta_{R^*}). \tag{32}$$

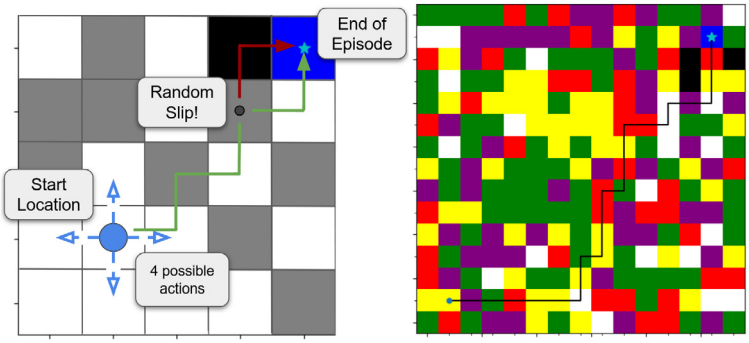
In the experiments, we used the ground-truth values (0.96, 0.0003, 0.58, 0.75, 3.35), based on the analysis of the Wilkinson Microwave Anisotropy Probe data (Bennett et al., 2003) by Sinha and Souradeep (2006). The Euclidean distance with no summary statistics was used for inference, with the same priors as in (Shaw et al., 2007):

$$\theta_\alpha \sim \mathrm{Unif}(0, 10), \quad \theta_{A_s} \sim \mathrm{Unif}(2.7, 1.3), \quad \theta_{k_c} \sim \mathrm{Unif}(10^{-7}, 10^{-3}),$$
$$\theta_{n_s} \sim \mathrm{Unif}(0.5, 1), \quad \theta_{R^*} \sim \mathrm{Unif}(0, 1).$$

Navigation World. The Navigation World (NW) model (Abel, 2019) is a simplified planning environment based on a grid world, where an agent needs to reach a target on a map (Fig. 6a). The agent moves in four directions (up, down, left and right) and receives a reward based on the colour of the tile it visits (e.g. +100 for reaching the goal, -500 for the black cell). We formulate our inference task as an inverse-reinforcement learning problem, where the goal is to approximate the multidimensional distribution over the parameters of the Q-learning agent's (Littman, 1994) reward function operating on the NW map.

The NW agent learns in a stable environment, and then has to operate in a stochastic "real world". The agent always starts at a fixed position and explores the environment with no step cost. It is first trained on the map, and after a certain number of training episodes, we ask it to sample a trajectory (e.g. green trajectory in Fig. 6a). However, this time, when we sample the trajectory, the agent can slip into an adjacent cell by accident (red trajectory in Fig. 6a). This may lead to multiple distinct trajectories with different rewards, causing multimodality in the reward space. Moreover, the agent's policy may converge to multiple optimal solutions, depending on the training initialization, which also contributes to multimodality that causes multiple trajectories for the same parameter setting. In reinforcement learning, when an optimal policy or rewards have multiple modes (Barrett and Narayanan, 2008; Kormushev and Caldwell, 2012), the solution to the inverse problem becomes particularly challenging (Franck and Koutsourelakis, 2017; Li et al., 2021).

The experiments were conducted on a more complex map, with tiles of five different colours corresponding to different rewards, shown in Fig. 6b. The simulation starts after setting the reward parameters for each colour (θ_{green}, θ_{purple}, θ_{red}, θ_{white}, θ_{yellow}), which are also simulator parameters we infer, and then proceeds to train the Q-agent for 8,000 episodes in a completely deterministic environment. Once the agent is trained, we sample 5 trajectories and learn their individual summaries: number of turns, number of steps and the reward. For example, a trajectory with summaries (9, 24, 51) is



(a) NW demonstration (b) NW experiment environment

Fig. 6. (a) In the **NW** environment the agent (blue circle) starts at a fixed location and can perform four actions: going up, down, left and right. Since the environment is stochastic, the agent may deviate from the optimal green trajectory and end up in a black cell that heavily penalizes the reward. The episode ends once the agent reaches the goal. (b) The **NW** map that we used in the experiments, with an example observed trajectory shown. Our model needs to infer the reward cell colour parameters, given the summary statistics of the trajectory: in this case 9 turns, 24 steps, and 51 reward.

illustrated in Fig. 6b. We used the Euclidean distance between the summaries of the sampled and observed trajectories to fit the surrogate model, as well as independent uniform priors on the interval $[-20, 0]$ for the simulator parameters, whereas true parameter values were (0.0, -1.0, -1.0, -5.0, -10.0). The implementation details of the simulator can be found in the Supplement.

4.1.2. Comparison methods

In this section, we specify the implementation details for the proposed LV-DGP surrogate in BOLFI and three comparison methods used in our experiments: GP-BOLFI, neural-density and kernel mean embedding. First, we describe LV-DGP surrogates, followed by GP surrogates that were traditionally used in BOLFI. Then, we introduce two variants of neural density estimators (MAF, MDN) that have performed the best in empirical LFI experiments earlier (Papamakarios et al., 2017, 2019). We conclude with the implementation details for the Kernel Embedding for LFI (KELFI) method that outperformed other kernel mean embedding alternatives in Hsu and Ramos (2019). For all three comparison methods, we used the same hyperparameters as in their original papers.

LV-DGPs. Here, we specify implementation details of the LV-DGP model, defined in Section 3.1. Throughout the paper, for referring to multiple LV-DGP architectures, we use a naming convention where the name of the architecture specifies the exact sequence of layers. For example, 'LV-3GP' refers to a DGP with a LV layer followed by three GP layers. We also used the LV-GP (Dutordoir et al., 2018; Wang and Neal, 2012) model in the experiments, as it shares the same input augmentation mechanism as the LV-DGP and following implementation details, but consists of only one GP layer.

The following LV-DGP settings performed well across all experiments presented in the paper, and we recommend them as defaults parameters for the method. In all GPs, we used the squared exponential kernel. The initial value for the lengthscale before optimization was set to the square root of the dimension, and the variance was fixed to 1, since the data was standardized. Kernel parameters and the likelihood variance (initialized with 0.01) were optimized from their initial values: the final layer using natural gradients (initial step size of 0.01) and the inner layers with the Adam optimizer (initial step size of 0.005) (Kingma and Ba Adam, 2014). Scaled conjugate gradient optimization with the maximum number of function evaluations of 50 was used for GPs. The quantile-threshold ϵ_q set to 0.3, so we get 6 posterior predictive samples after applying quantile-conditioning. The full results of the sensitivity analysis on the choice of ϵ_q can be found in the Supplement. In all LV-DGP models, we used 50 inducing points, 5 importance-weighted samples and 20 samples for predictions and gradients. The LV-2GP' model is the only exception, since it was configured to match the vanilla GP complexity; it used 10 inducing points per layer, and only 2 importance-weight samples. Lastly, we draw additional comparisons with Stochastic Gradient Hamiltonian Monte Carlo (Havasi et al., 2018) as an alternative method instead of IWVI in the Supplement. The LV-DGP model was implemented in Python with GPFlow (Matthews et al., 2017). Engine for Likelihood-Free Inference (ELFI) (Lintusaari et al., 2018) was used as the platform for the implementations, and the proposed model is available in ELFI for application and further development (elfi.ai).

GPs. The vanilla GP surrogate was initially introduced for BOLFI (Gutmann and Corander, 2016). The GP model had as hyperparameters the lengthscale of the squared-exponential kernel lengthscale, variance and added bias component. Gamma priors were used for all three of them, initialized by the expected value and variance chosen based on initial standardized data. We used LCBSC acquisition in BO. The model was implemented in Python with the GPy package (GPy, 2012).

MAF (Papamakarios et al., 2017) is an implementation of normalizing flow that uses Masked Autoencoder for Distribution Estimation (MADE) (Germain et al., 2015) as building blocks, where each conditional probability is modelled by a single Gaussian component. In the experiments, we used the architecture with 5 stacked MADEs in the flow and 2 hidden layers,

containing 50 hidden units (sequential strategy for assigning degrees to hidden nodes was used) with hyperbolic tangent as an activation function. The model was trained with Adam optimization, using a minibatch size of 100, and a learning rate of 10^{-4}. L2 regularization with coefficient 10^{-6} was added. The training was performed with 300 epochs in 5 batches, with the number of populations equal to the total number of observations divided by the number of batches. We report results for 200 and 1000 total observations.

MDN (Papamakarios et al., 2019) is a feedforward neural network that takes the observation $\mathbf{s}_\theta$ as an input and outputs the parameters of a Gaussian mixture over $\boldsymbol{\theta}$. We use an ensemble of 5 MDNs in our experiments with the same architecture: 2 hidden layers with 30 hidden units in each with the hyperbolic tangent activation function. The parameters for optimization and training procedures were the same as for the MAF. We report results for 200 and 1000 total observations.

KELFI (Hsu and Ramos, 2019) is a surrogate likelihood model that leverages smoothness properties of conditional mean embeddings. Different strategies for adjusting marginal kernel means likelihood (MKML) hyperparameters were used for the experiments. For TEs, hyperparameters $(\epsilon, \beta, \gamma)$ were chosen by gradient optimization, denoted as 'All-Opt' in the original paper. The initial values of $(0.06, 0.6, 10^{-6})$ were used for initialization. However, for the BDM and the NW cases, we were unable to train the hyperparameters using gradients due to a numerical error in the original KELFI software. Therefore, for these cases we chose the grid based optimization strategy, denoted as 'Scale-Global-Opt' and shown to be the second-best strategy in the original paper. In this strategy only (ϵ, β) we optimized using 100 uniformly distributed samples on the intervals (0.5, 1.5) and (0.05, 0.15) respectively. As for all other models, we sample only one observation per parameter point. Unlike the rest of the comparison methods, KELFI has not been adapted yet to use active learning strategies. Therefore, we report results only for the 1000 total observations.

4.2. DGPs and GPs as surrogate models

In this section, we use the LV-2GP architecture as DGPs, and compare them with vanilla GPs, as surrogates for BOLFI. The LV-2GP is the simplest model that combines benefits of multiple GPs and an inclusion of a LV layer. We expected DGPs to have advantages over GPs for multimodal cases, and hoped for them to have similar performance and data-efficiency in the rest of the cases. This is not obvious, since DGP, as a more flexible model, is expected to have a larger variance. However, the difference turned out to be negligible in practice.

We use one-dimensional TEs to demonstrate the differences between GP and DGP surrogates on four types of objective functions: non-stationary (TE1), multimodal (TE2), heteroscedastic (TE3) and non-stationary multimodal (TE4). Fig. 7 clearly shows that DGPs can handle multimodality, while vanilla GPs cannot. In the non-stationary TE1, the DGP better retains the shape of the larger density mass, while GPs have a tendency to accurately approximate only one of its peaks, completely ignoring the other. This GP problem with multimodality is clearest in the TE2 case, where GP approximates equal modes with a single uniform-like density, unlike DGP, who is able to capture the multimodal uncertainty quite well. The TE4 example was shown to be challenging for both GPs and DGPs. GP seems to approximate all posterior modes with an unimodal density, similarly as it did in the TE2 case (only now it is skewed), while DGP in general maintains the approximation of the biggest mode and struggles at separating the smallest two. It is particularly difficult to separate the smallest modes, as they have lower probability of occurrence (0.4), which in a small data setting can be problematic. Lastly, for the TE3 case, both models perform similarly, although DGP should have struggled more with overfitting the complex noise component compared to GP, as it is a more flexible surrogate. Overall, the TE results strengthen our hypothesis about DGPs being able to handle objective functions with more irregular uncertainties, which is further confirmed with Wasserstein distances summarized in Fig. 8a-8d.

In the case studies, DGPs are either better or on the same level as vanilla GPs in approximating the posterior. The clearest advantage of DGPs is observed in the NW case (Fig. 12), where the DGP samples closely follow the true marginals despite multimodality of the posterior. Some minor improvements over GPs are observed in the SL case (Fig. 10); however, both surrogates visibly struggle with replicating the complex shape of the parameter marginals and more simulations are likely needed to approximate all the posterior details (see our data-efficiency experiments below). As for the higher variance, it is most noticeable in the BDM (Fig. 9) case. The DGP is sufficiently flexible to provide better solutions than GP, but it also has a higher tendency to overfit in this case. The mean Wasserstein distance (Table 2) shows only a slight advantage of DGP over the GP model. Lastly, the CI case (Fig. 11) has multimodality only in the subspace $(\theta_\alpha, \theta_{k_c})$ (Starobinskii, 1979), and it is precisely where the DGP have improved approximations over the GP. In summary, DGPs unlike GPs can work with both multimodal and unimodal uncertainties, making them especially suitable for cases when no prior information about the form of the uncertainty is available.

We conducted additional experiments to evaluate the performance of DGPs and GPs under different simulation budgets in the considered case studies. The results in Fig. 13 show that DGP steadily improved its median performance in all four case studies. The most noticeable improvement can be observed in the SL and NW cases, where DGP clearly outperforms GP with more than some tens of simulations in NW, and with 100 in SL. This is likely related to inability of the GP to model multimodality, which is prevalent in these case studies. As for the unimodal BDM or partially multimodal CI, DGP improves with more data, in the sense of reducing the median Wasserstein distance. There is also an abrupt increase in variance after getting over 100 simulations, which is likely due to the parameter optimization procedure converging to a more flexible configuration. More data may help the optimization procedure to converge to a more accurate model, as

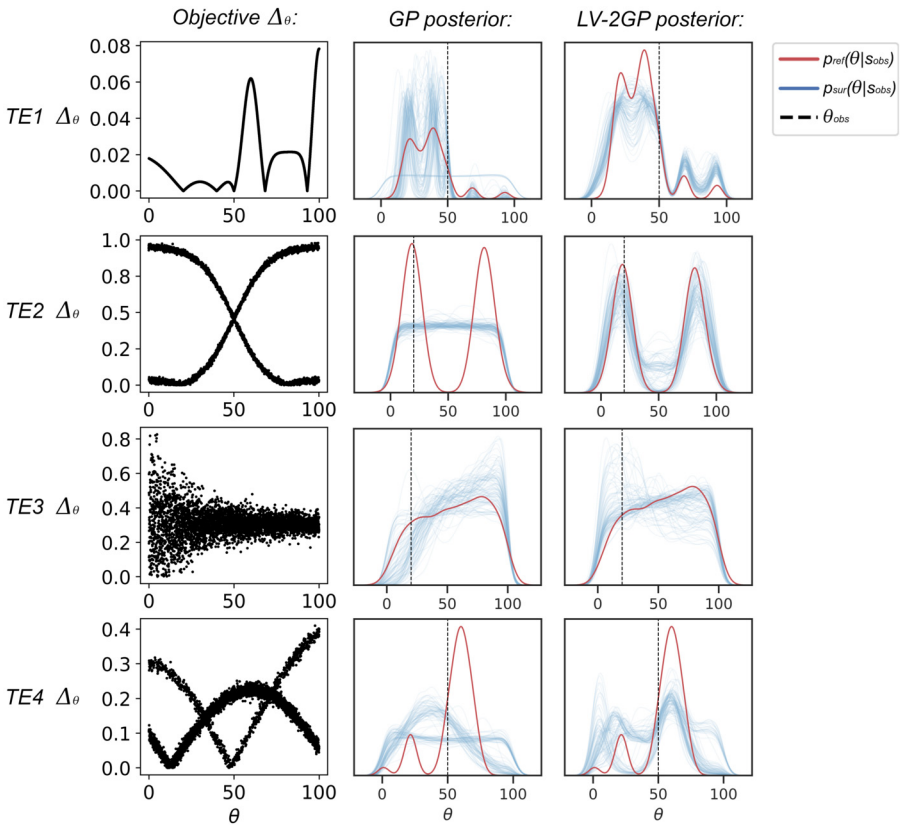


Fig. 7. Approximation quality of posteriors by vanilla GP (middle column) and DGP (right column, where LV-2GP is an instance of DGP), in four demonstration examples (rows). The figures show that DGPs maintain close approximation of the reference posterior (red lines) in **TE1** and **TE3**; DGPs significantly surpass GPs in **TE2** and show good improvement in **TE4**. Both surrogates try to model the discrepancy functions (left column), and approximate the posterior (blue lines) of $\boldsymbol{\theta}$ for $\Delta_{\boldsymbol{\theta}} \to 0$. The quality of inference can be inferred from how closely surrogate posteriors $p_{\text{sur}}(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$ with 100 different initial evidence sets follow the reference posterior $p_{\text{ref}}(\boldsymbol{\theta}|\mathbf{s}_{\text{obs}})$.

indicated by DGPs improving its median performance, but not the variance, with more than 150 simulations in BDM and CI cases. The high variance significantly hinders the DGP overall performance for these two case studies, and especially for the BDM case. Lastly, we also note that in all four case studies, GPs have much smaller posterior accuracy improvement with more data than DGPs, which means that more simulations cannot compensate for the restrictive model. The gap in flexibility between two model also explains their difference in the variance. In conclusion, DGPs being a more flexible model than GPs, can further improve the approximations with more data for both unimodal and multimodal distributions, however with a higher variance in high-dimensional unimodal cases. These results support our claim of DGPs being capable of modelling multimodal target distributions with a limited number of function evaluations.

4.3. Comparison of LFI approaches

The comparison of the proposed DGP surrogates to other LFI approaches (Table 2) show that DGPs outperform MAF, MDN and KELFI alternatives. Across all case studies, none of those methods achieve a performance comparable to DGPs, even with much more data (1000 simulations vs 200). Even though MAF and MDNs use active learning, they are trying to model the likelihood directly, in contrast to DGPs that model the discrepancy. The former is a more general and harder problem, that requires many more observations with the benefit of not having to retrain the model if the observed data is changed. On the other hand, KELFI does not use any active learning strategies, therefore, it was expected to have worse data-efficiency than DGPs. The only exception, where those alternative techniques performed better than DGPs, is the MDNs and MAFs in the TE1 case. This indicates that these neural density estimators have the advantage in non-stationary cases, although they also have much greater risk of overfitting, as shown by their poor performance in TE3. In summary, all the considered alternatives have the necessary flexibility to show good performance on the considered cases, however, they require significantly more data than DGPs, making them unsuitable for modelling irregularly behaved distributions in a small data setting. Therefore, DGP is the preferable candidate among them for doing LFI with computationally expensive simulators.

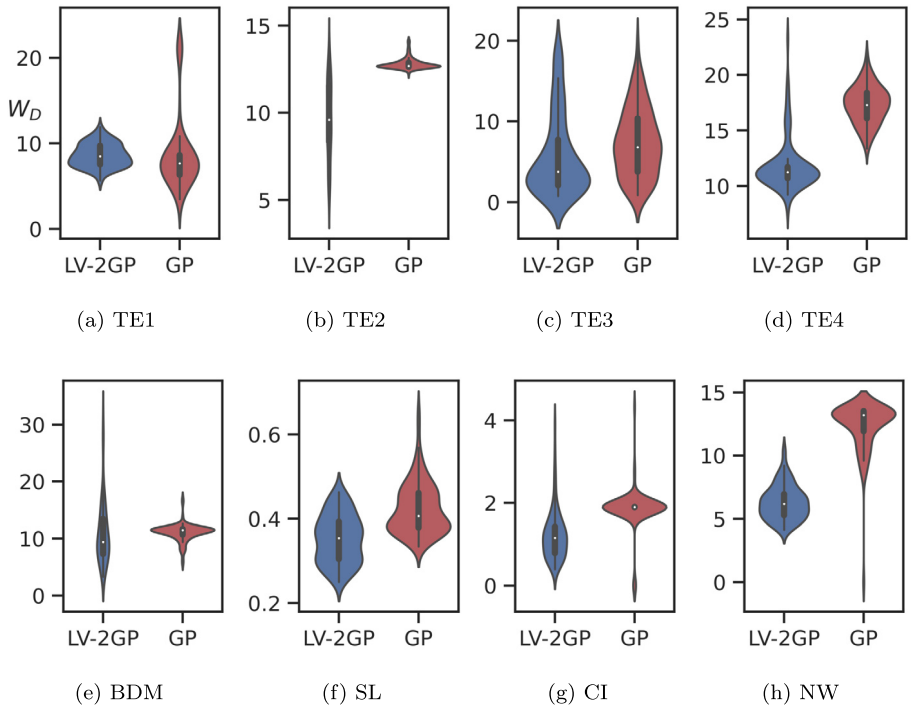


Fig. 8. Wasserstein distance between the surrogate models (GP and LV-2GP, an instance of DGP) posteriors and the true posterior of $\boldsymbol{\theta}$; the smaller the distance, the better is the quality of approximations. The DGP approximations of the true posterior are better on multimodal **TE2** (b), **TE4** (d), **SL** (f), **CI** (g) and **NW** (h) examples, maintaining comparable performance on the rest of the cases. The white dot on the violin plot is the median, the black bar is the interquartile range, and lines stretched from the bar show lower/upper adjacent values.

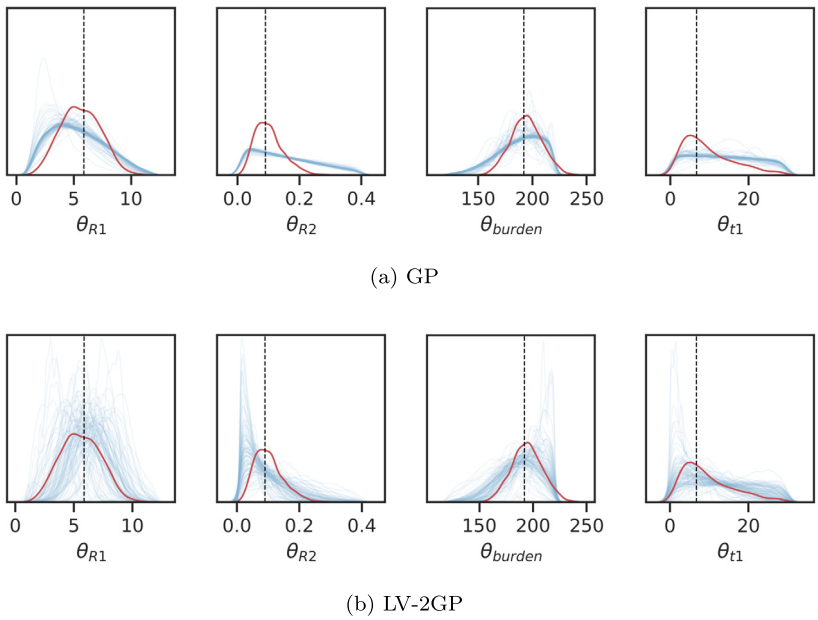
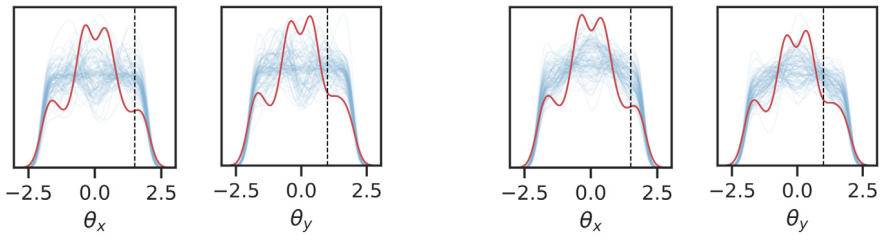
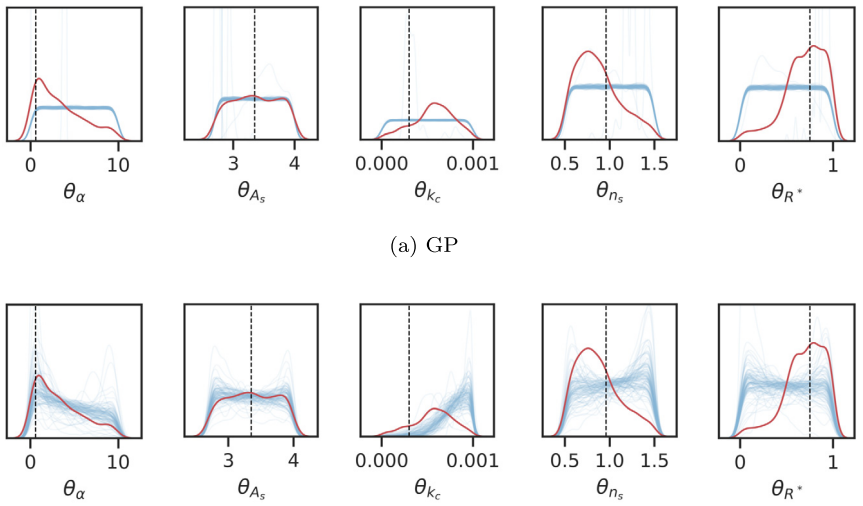


Fig. 9. Approximation quality of posterior marginals (columns) by vanilla GP (top) and DGP (bottom) in the **BDM** case. While DGP surrogate posterior samples (blue lines) do not converge towards wrong marginals, unlike GPs, they clearly have much higher variance around true posterior marginals (red lines).



(a) GP (b) LV-2GP

Fig. 10. Approximation quality of posterior marginals (columns) by vanilla GP (left) and DGP (right) in the **SL** case. Surrogate posterior samples (blue lines) of DGP provide only a slightly more accurate approximation of the posterior marginals (red lines) than GPs due to the absence of uniform-like (flat) sample instances.



(a) GP

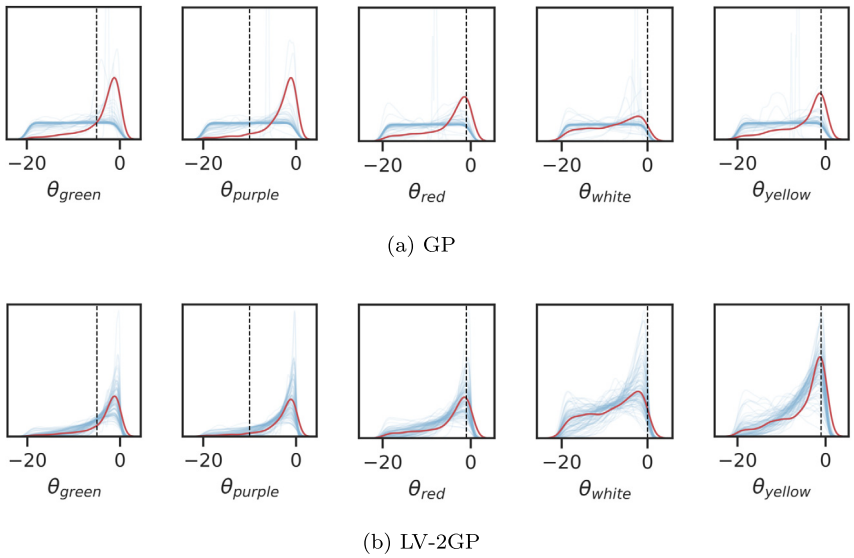
(b) LV-2GP

Fig. 11. Approximation quality of posterior marginals (columns) by vanilla GP (top) and DGP (bottom) in the **CI** case. Surrogate posterior samples (blue lines) of DGPs improve upon GPs in the joint space of parameters θ_α and θ_{k_c}, where multimodality occurs (Starobinskii, 1979), and maintain comparable poor approximations to the true $\boldsymbol{\theta}$ marginals (red lines) with higher variance for the rest of parameters.

We also compared different architectures of the LV-DGP model, assessing the influence of the amount of layers and surrogate parameters. Based on results from Table 2, the inclusion of the LV layer increased the model performance on simulators TE1 (only for LV-2GP), TE4 and NW, which required greater flexibility of the surrogate, and at the same time worsened the results for TE2 and TE3, whose discrepancy functions were simple but had complex noise components. This suggests that the LV layer should not be used with simple objectives and complex noise components, as there is a higher risk of overfitting. Moreover, the inclusion of the LV layer in the DGP architecture results only in the slight increase in the running time (see Table 2 from the Supplement), while performance often improves. At the same time, deeper architectures than 2 GPs did not show significantly better results, since they only increased the variance, which is expected, as having more layers implies more parameters to fit. The connection between high variance and the amount of parameters can be further confirmed by comparing LV-2GP and LV-2GP' models. As LV-2GP' model has much fewer parameters, it also has much lower variance than LV-2GP, proving that reducing the amount of parameters is a valid way of controlling the variance; however, it also comes at the cost of worse general performance. In summary, the architecture with one latent and two GP layers is the preferred one, since it is the simplest model that has most of the benefits from the inclusion of the LV layer. The configuration with the LV layer is more flexible with a higher risk of suffering from the high variance, while deeper than 2GP architectures unnecessarily increase the variance even further and can be considered unfit for the small-data setting implied by computationally expensive simulators.

5. Discussion

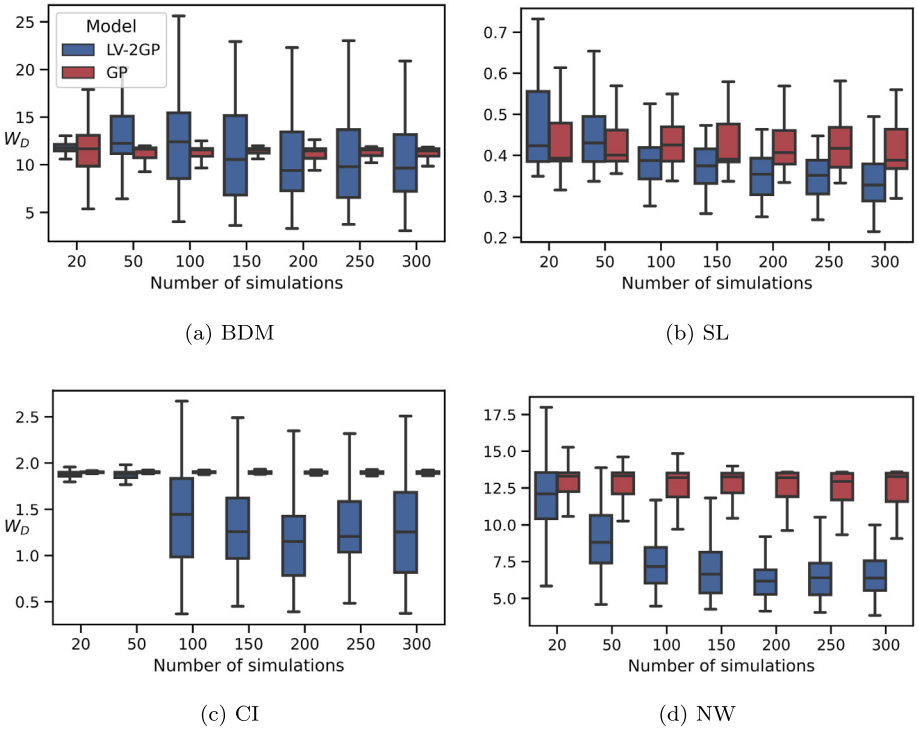
We introduced a novel method for statistical inference when the likelihood is not available, but drawing samples from a simulator is possible, although computationally intensive. The introduced method is an extension of BOLFI (Gutmann and Corander, 2016) where we have adopted DGP surrogates instead of GP surrogates to model the relationship of the param-



(a) GP

(b) LV-2GP

Fig. 12. Approximation quality of posterior marginals (columns) by vanilla GP (top) and DGP (bottom) in the **NW** case. Surrogate posterior samples (blue lines) of DGPs are flexible enough to approximate all marginals of the true parameter $\boldsymbol{\theta}$ posterior (red lines), though have higher variance, while GP samples converge to poor marginal distributions.



(a) BDM

(b) SL

(c) CI

(d) NW

Fig. 13. Wasserstein distance between surrogate posteriors and the true posterior, shown in the case study experiments, as a function of the total number of simulations. DGP approximation accuracy scales better in the **NW** and **SL**, and demonstrates lower medians with higher variance than GP in **BDM** and **CI**, as the number of simulations grows. The box plots were computed with distances across 100 simulations. The horizontal line on box plots shows the median, the bar shows upper and lower quartiles, and the whiskers indicate the rest of the quartiles.

eters and the stochastic discrepancy between observed data and simulated data. These new surrogates use quantile-based modifications for an acquisition function and likelihood approximation, making it feasible for LFI problems. The proposed extension retains the active learning property of BOLFI so that the posterior distribution is sought out with as few samples as possible. The flexibility of the DGPs improved the resulting posterior approximations in cases where flexibility was required, and otherwise the observed performance was similar in both cases. Especially good improvements were observed in

Table 2
LV-DGP models generally performed the best across all experiments (columns) and comparison models (rows). The performance was measured with the Wasserstein distance (mean ± std) between the surrogate model posterior and the true posterior of θ, across 100 runs. Models from Section 4.2 and the best results in each column are highlighted in bold. * denotes models, which used 1000 simulations instead of 200 for the sample-efficiency comparison.

Model	TE1	TE2	TE3	TE4
2GP	9.32 ± 2.33	$\mathbf{9.17 \pm 1.23}$	$\mathbf{3.36 \pm 2.76}$	16.77 ± 2.82
4GP	9.36 ± 2.38	9.48 ± 1.37	3.54 ± 2.59	16.56 ± 2.43
GP	8.66 ± 4.25	12.81 ± 0.31	7.34 ± 4.18	17.31 ± 1.68
LV-GP	8.92 ± 5.33	11.93 ± 2.49	8.97 ± 5.93	11.68 ± 1.27
LV-2GP'	10.96 ± 0.89	11.39 ± 1.48	5.06 ± 2.47	$\mathbf{10.53 \pm 0.42}$
LV-2GP	8.62 ± 1.35	9.65 ± 2.20	5.78 ± 4.98	11.83 ± 2.28
LV-5GP	9.15 ± 1.47	10.40 ± 1.59	5.14 ± 4.34	11.94 ± 2.02
KELFI*	20.78 ± 0.23	10.27 ± 0.42	3.42 ± 1.82	17.11 ± 1.57
MAF*	$\mathbf{4.89 \pm 1.82}$	9.24 ± 1.01	10.57 ± 4.72	15.61 ± 1.67
MAF	6.39 ± 1.77	10.96 ± 2.27	10.30 ± 5.02	14.83 ± 2.20
MDN*	21.15 ± 2.05	10.37 ± 4.18	10.33 ± 4.67	11.14 ± 5.72
MDN	6.56 ± 4.00	12.30 ± 5.88	12.52 ± 8.69	11.53 ± 7.25

Model	BDM	SL	CI	NW
2GP	12.38 ± 6.13	0.36 ± 0.06	1.65 ± 1.05	8.85 ± 1.47
4GP	11.50 ± 5.21	0.37 ± 0.05	1.34 ± 0.70	8.55 ± 1.83
GP	11.09 ± 1.48	0.42 ± 0.06	1.86 ± 0.47	12.38 ± 1.87
LV-GP	11.16 ± 5.32	0.39 ± 0.02	1.83 ± 0.27	8.29 ± 2.06
LV-2GP'	$\mathbf{10.54 \pm 4.13}$	0.39 ± 0.02	1.61 ± 0.47	6.56 ± 1.85
LV-2GP	10.87 ± 5.39	$\mathbf{0.35 \pm 0.06}$	1.22 ± 0.60	$\mathbf{6.30 \pm 1.34}$
LV-5GP	11.59 ± 5.41	0.37 ± 0.05	$\mathbf{1.11 \pm 0.62}$	6.84 ± 1.82
KELFI*	31.96 ± 15.40	0.48 ± 0.01	2.42 ± 1.21	14.74 ± 5.21
MAF*	26.21 ± 4.88	0.63 ± 0.15	2.89 ± 0.87	13.72 ± 1.19
MAF	31.36 ± 9.16	0.72 ± 0.19	1.93 ± 0.29	13.87 ± 0.94
MDN*	26.30 ± 4.34	0.51 ± 0.07	3.65 ± 0.34	9.31 ± 0.81
MDN	28.39 ± 4.07	0.56 ± 0.10	3.42 ± 1.23	11.63 ± 0.91

cases where the distribution of the discrepancy was multimodal, i.e. in cases where GP is known to perform poorly as an estimator.

The improvements from using DGP surrogates come with increased computational cost, which we demonstrated to be negligible for computationally heavy simulators. DGPs also had a higher variance in unimodal higher dimensional examples. Even though data-efficiency experiments indicated performance improvement with more observations, the problem with high variance persists, and is likely related to the ability to model multimodality, as comparison methods, that showed this ability as well, had similar variance in unimodal cases. Reducing the amount of DGP parameters (e.g. opting for 'shallower' configurations with fewer inducing points) or bringing more prior knowledge should help control the variance. We recommend using DGPs in cases with multimodal target distributions, where their more expressive surrogates are needed and work better than vanilla GPs. If we know a GP is sufficiently flexible, more time could be spent on additional simulations rather than a more flexible model.

A natural progression of this work is to analyse DGP uncertainty decomposition and its propagation through layers. Decomposing the uncertainty into its aleatoric and epistemic components would allow better exploration of the parameter space. This is especially important when dealing with multimodal distributions, since they often have high epistemic uncertainty that may prevent BO from exploring other parameter regions. As for uncertainty propagation, individual layers of DGPs can be used to learn intermediate transitions inside the simulator. Doing so would require opening the black-box of the simulator and incorporating these intermediate transitions as data in the training process. This additional information would lead to better usage of simulator time, since some futile simulations could be abandoned once their transition variables become available. In conclusion, better uncertainty decomposition and propagation can further improve data-efficiency of LFI, when dealing with computationally expensive simulators that have irregular noise models.

There has been a parallel work on developing surrogates for multimodal target distributions, namely GLLiM-ABC by Forbes et al. (2021), in which Gaussian mixtures are used to fit the posteriors and learn summary statistics. So far, it has been applied to only larger data setting $O(10^5)$, which is different from the one we consider in this work. Further research on the performance of GLLiM-ABC with few hundreds of simulations is needed to draw more concrete comparisons with the DGP surrogates.

Acknowledgements

This work was supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI; grants 328400, 325572, 319264, 292334). Authors HP and JC were also supported by European Research Council grant 742158 (SCARABEE, Scalable inference algorithms for Bayesian evolutionary epidemiology), and SK by the UKRI Turing AI World-Leading Researcher Fellowship EP/W002973/1. Computational resources were provided by the Aalto Science-IT Project.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csda.2022.107529.

References

Abel, D., 2019. Simple rl: reproducible reinforcement learning in Python. In: ICLR Workshop on Reproducibility in Machine Learning.

Alsing, J., Charnock, T., Feeney, S., Wandelt, B., 2019. Fast likelihood-free cosmology with neural density estimators and active learning. Mon. Not. R. Astron. Soc. 488, 4440–4458.

Barrett, L., Narayanan, S., 2008. Learning all optimal policies with multiple criteria. In: Proceedings of the 25th International Conference on Machine Learning, pp. 41–47.

Beaumont, M.A., 2010. Approximate bayesian computation in evolution and ecology. Annu. Rev. Ecol. Evol. Syst. 41, 379–406.

Beaumont, M.A., Zhang, W., Balding, D.J., 2002. Approximate Bayesian computation in population genetics. Genetics 162, 2025–2035. https://www.genetics.org/content/162/4/2025. https://www.genetics.org/content/162/4/2025.full.pdf.

Beaumont, M.A., Cornuet, J.-M., Marin, J.-M., Robert, C.P., 2009. Adaptive approximate Bayesian computation. Biometrika 96, 983–990.

Bennett, C., Hill, R., Hinshaw, G., Nolta, M., Odegard, N., Page, L., Spergel, D., Weiland, J., Wright, E., Halpern, M., et al., 2003. First-year wilkinson microwave anisotropy probe (wmap)* observations: foreground emission. Astrophys. J. Suppl. Ser. 148, 97.

Bernton, E., Jacob, P.E., Gerber, M., Robert, C.P., 2019. Approximate bayesian computation with the wasserstein distance. preprint. arXiv:1905.03747.

Boitard, S., Rodríguez, W., Jay, F., Mona, S., Austerlitz, F., 2016. Inferring population size history from large samples of genome-wide molecular data-an approximate Bayesian computation approach. PLoS Genet. 12, e1005877.

Chowdhury, S.R., Oliveira, R., Ramos, F., 2020. Active learning of conditional mean embeddings via bayesian optimisation. In: Conference on Uncertainty in Artificial Intelligence. PMLR, pp. 1119–1128.

Cornuet, J.-M., Santos, F., Beaumont, M.A., Robert, C.P., Marin, J.-M., Balding, D.J., Guillemaud, T., Estoup, A., 2008. Inferring population history with diy abc: a user-friendly approach to approximate bayesian computation. Bioinformatics 24, 2713–2719.

Cox, D.D., John, S., 1992. A statistical method for global optimization. In: [Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, pp. 1241–1246.

Csilléry, K., Blum, M.G., Gaggiotti, O.E., François, O., 2010. Approximate bayesian computation (abc) in practice. Trends Ecol. Evol. 25, 410–418.

Damianou, A., Lawrence, N., 2013. Deep Gaussian processes. In: Artificial Intelligence and Statistics, pp. 207–215.

Diggle, P.J., Gratton, R.J., 1984. Monte Carlo methods of inference for implicit statistical models. J. R. Stat. Soc., Ser. B, Methodol. 46, 193–227.

Dunlop, M.M., Girolami, M.A., Stuart, A.M., Teckentrup, A.L., 2018. How deep are deep Gaussian processes? J. Mach. Learn. Res. 19, 2100–2145.

Dutordoir, V., Salimbeni, H., Hensman, J., Deisenroth, M., 2018. Gaussian process conditional density estimation. Adv. Neural Inf. Process. Syst. 31, 2385–2395.

Forbes, F., Nguyen, H.D., Nguyen, T.T., Arbel, J., 2021. Approximate bayesian computation with surrogate posteriors. hal-03139256v3.

Franck, I.M., Koutsourelakis, P.-S., 2017. Multimodal, high-dimensional, model-based, bayesian inverse problems with applications in biomechanics. J. Comput. Phys. 329, 91–125.

Genevay, A., Cuturi, M., Peyré, G., Bach, F., 2016. Stochastic optimization for large-scale optimal transport. In: Advances in Neural Information Processing Systems, pp. 3440–3448.

Genton, M.G., Ronchetti, E., 2003. Robust indirect inference. J. Am. Stat. Assoc. 98, 67–76.

Germain, M., Gregor, K., Murray, I., Larochelle Made, H., 2015. Masked autoencoder for distribution estimation. In: International Conference on Machine Learning, pp. 881–889.

Gouriéroux, C., Phillips, P.C., Yu, J., 2010. Indirect inference for dynamic panel models. J. Econom. 157, 68–77.

GPy, 2012. GPy: a gaussian process framework in python. http://github.com/SheffieldML/GPy.

Greenberg, D.S., Nonnenmacher, M., Macke, J.H., 2019. Automatic posterior transformation for likelihood-free inference. preprint. arXiv:1905.07488.

Gutmann, M.U., Corander, J., 2016. Bayesian optimization for likelihood-free inference of simulator-based statistical models. J. Mach. Learn. Res. 17, 4256–4302.

Guvenen, F., Smith, A., 2010. Inferring labor income risk from economic choices: an indirect inference approach. Technical Report. National Bureau of Economic Research.

Hartig, F., Calabrese, J.M., Reineking, B., Wiegand, T., Huth, A., 2011. Statistical inference for stochastic simulation models – theory and application. Ecol. Lett. 14, 816–827. https://doi.org/10.1111/j.1461-0248.2011.01640.x.

Havasi, M., Hernández-Lobato, J.M., Murillo-Fuentes, J.J., 2018. Inference in deep Gaussian processes using stochastic gradient Hamiltonian Monte Carlo. In: Advances in Neural Information Processing Systems, pp. 7506–7516.

Hebbal, A., Brevault, L., Balesdent, M., Talbi, E.-G., Melab, N., 2020. Bayesian optimization using deep gaussian processes with applications to aerospace system design. Optim. Eng., 1–41.

Heggland, K., Frigessi, A., 2004. Estimating functions in indirect inference. J. R. Stat. Soc., Ser. B, Stat. Methodol. 66, 447–462.

Hsu, K., Ramos, F., 2019. Bayesian learning of conditional kernel mean embeddings for automatic likelihood-free inference. preprint. arXiv:1903.00863.

Jenatton, R., Archambeau, C., González, J., Seeger, M., 2017. Bayesian optimization with tree-structured dependencies. In: International Conference on Machine Learning, pp. 1655–1664.

Kingma, D.P., Ba Adam, J., 2014. A method for stochastic optimization. preprint. arXiv:1412.6980.

Kormushev, P., Caldwell, D.G., 2012. Simultaneous discovery of multiple alternative optimal policies by reinforcement learning. In: 2012 6th IEEE International Conference Intelligent Systems. IEEE, pp. 202–207.

Li, R., Shikhov, I., Arns, C.H., 2021. Solving multiphysics, multiparameter, multimodal inverse problems: an application to nmr relaxation in porous media. Phys. Rev. Appl. 15, 054003.

Lintusaari, J., Gutmann, M.U., Dutta, R., Kaski, S., Corander, J., 2017. Fundamentals and recent developments in approximate Bayesian computation. Syst. Biol. 66, e66–e82.

Lintusaari, J., Vuollekoski, H., Kangasrääsiö, A., Skytén, K., Järvenpää, M., Marttinen, P., Gutmann, M.U., Vehtari, A., Corander, J., Kaski, S., 2018. Elfi: engine for likelihood-free inference. J. Mach. Learn. Res. 19, 643–649.

Lintusaari, J., Blomstedt, P., Sivula, T., Gutmann, M.U., Kaski, S., Corander, J., 2019. Resolving outbreak dynamics using approximate Bayesian computation for stochastic birth-death models. Wellcome Open Res. 4.

Littman, M.L., 1994. Markov games as a framework for multi-agent reinforcement learning. In: Machine Learning Proceedings 1994. Elsevier, pp. 157–163.

Lueckmann, J.-M., Bassetto, G., Karaletsos, T., Macke, J.H., 2019. Likelihood-free inference with emulator networks. In: Symposium on Advances in Approximate Bayesian Inference. PMLR, pp. 32–53.

Matthews, A.G.d.G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., Hensman, J., 2017. GPflow: a Gaussian process library using TensorFlow. J. Mach. Learn. Res. 18, 1–6. http://jmlr.org/papers/v18/16-537.html.

Meeds, E., Welling, M., 2014. Gps-abc: Gaussian process surrogate approximate Bayesian computation. In: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI'14. AUAI Press, Arlington, Virginia, United States, pp. 593–602.

Monfardini, C., 1998. Estimating stochastic volatility models through indirect inference. Econom. J. 1, 113–128.
Nakagome, S., Fukumizu, K., Mano, S., 2013. Kernel approximate Bayesian computation in population genetic inferences. Stat. Appl. Genet. Mol. Biol. 12, 667–678.
Ong, V.M., Nott, D.J., Tran, M.-N., Sisson, S.A., Drovandi, C.C., 2018. Variational bayes with synthetic likelihood. Stat. Comput. 28, 971–988.
Papamakarios, G., Pavlakou, T., Murray, I., 2017. Masked autoregressive flow for density estimation. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17. Curran Associates Inc., Red Hook, NY, USA, pp. 2335–2344.
Papamakarios, G., Sterratt, D.C., Murray, I., 2019. Sequential neural likelihood: fast likelihood-free inference with autoregressive flows. In: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Vol. 89. PMLR, pp. 837–848.
Price, L.F., Drovandi, C.C., Lee, A., Nott, D.J., 2018. Bayesian synthetic likelihood. J. Comput. Graph. Stat. 27, 1–11. https://doi.org/10.1080/10618600.2017.1302882.
Pritchard, J.K., Seielstad, M.T., Perez-Lezaun, A., Feldman, M.W., 1999. Population growth of human y chromosomes: a study of y chromosome microsatellites. Mol. Biol. Evol. 16, 1791–1798.
Rubens, N., Elahi, M., Sugiyama, M., Kaplan, D., 2015. Active learning in recommender systems. In: Recommender Systems Handbook. Springer, pp. 809–846.
Salimbeni, H., Deisenroth, M., 2017. Doubly stochastic variational inference for deep Gaussian processes. In: Advances in Neural Information Processing Systems, pp. 4588–4599.
Salimbeni, H., Dutordoir, V., Hensman, J., Deisenroth, M., 2019. Deep gaussian processes with importance-weighted variational inference. In: International Conference on Machine Learning. PMLR, pp. 5589–5598.
Settles, B., 2009. Active learning literature survey. Technical Report. University of Wisconsin.
Shah, A., Wilson, A.G., Ghahramani, Z., 2013. Bayesian optimization using student-t processes. In: NIPS Workshop on Bayesian Optimisation.
Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N., 2016. Taking the human out of the loop: a review of Bayesian optimization. Proc. IEEE 104, 148–175. https://doi.org/10.1109/JPROC.2015.2494218.
Shaw, J., Bridges, M., Hobson, M., 2007. Efficient bayesian inference for multimodal problems in cosmology. Mon. Not. R. Astron. Soc. 378, 1365–1370.
Sinha, R., Souradeep, T., 2006. Post-wmap assessment of infrared cutoff in the primordial spectrum from inflation. Phys. Rev. D 74, 043518.
Sisson, S., Fan, Y., Beaumont, M., 2018. Handbook of approximate Bayesian computation. In: Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, Taylor & Francis Group. https://books.google.no/books?id=gSWFZwEACAAJ.
Small, P.M., Hopewell, P.C., Singh, S.P., Paz, A., Parsonnet, J., Ruston, D.C., Schecter, G.F., Daley, C.L., Schoolnik, G.K., 1994. The epidemiology of tuberculosis in San Francisco–a population-based study using conventional and molecular methods. N. Engl. J. Med. 330, 1703–1709.
Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., Adams, R., 2015. Scalable bayesian optimization using deep neural networks. In: International Conference on Machine Learning, pp. 2171–2180.
Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W., 2012. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. IEEE Trans. Inf. Theory 58, 3250–3265.
Starobinskii, A., 1979. Spectrum of relict gravitational radiation and the early state of the universe. JETP Lett. 30, 682–685.
Sunnåker, M., Busetto, A.G., Numminen, E., Corander, J., Foll, M., Dessimoz, C., 2013. Approximate bayesian computation. PLoS Comput. Biol. 9, e1002803.
Tanaka, M.M., Francis, A.R., Luciani, F., Sisson, S., 2006. Using approximate Bayesian computation to estimate tuberculosis transmission parameters from genotype data. Genetics 173, 1511–1520.
van der Vaart, E., Beaumont, M.A., Johnston, A.S., Sibly, R.M., 2015. Calibration and evaluation of individual-based models using approximate Bayesian computation. Ecol. Model. 312, 182–190.
Wang, D., Brown, G.J., 2006. Computational Auditory Scene Analysis: Principles, Algorithms, and Applications. Wiley-IEEE Press.
Wang, C., Neal, R.M., 2012. Gaussian process regression with heteroscedastic or non-gaussian residuals. preprint. arXiv:1212.6246.
Williams, C.K., Rasmussen, C.E., 2006. Gaussian Processes for Machine Learning, Vol. 2. MIT Press, Cambridge, MA.
Wood, S.N., 2010. Statistical inference for noisy nonlinear ecological dynamic systems. Nature 466, 1102–1104.
Zhu, C., Byrd, R.H., Lu, P., Nocedal, J., 1997. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. ACM Trans. Math. Softw. 23, 550–560.