High-dimensional Bayesian Optimization for Learning in Generative Models

Afonso Eduardo

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh

2018

Abstract

The learning process can be cast as an inference problem. Inference is typically difficult and standard inference methods rely on the ability to explicitly evaluate the likelihood function. However, many models such as those in natural sciences are based on stochastic simulators and, consequently, the likelihood function is not explicitly defined. Likelihood-free inference methods are able to solve this problem, but can be inefficient. As an alternative, Bayesian optimization for likelihood-free inference (BOLFI) has emerged. Thus far, only low-dimensional problems have been explored. In this work, we focus on relatively high-dimensional likelihood-free inference, where the number of parameters to be inferred is large. The contribution is twofold. First, by exploiting recent advances in high-dimensional Bayesian optimization, we show that it is possible to improve the efficiency of BOLFI. In particular, the methods rely on an additive structure that must be learned. We demonstrate that it is possible to learn the structure and that such process plays an important role in improving the efficiency of Bayesian optimization and likelihood-free inference. Second, there is a general lack of tools to assess high-dimensional likelihood-free inference methods. We explore and design performance measures and models that can be applied in this setting. The methods are tested on these models and evaluated according to the proposed performance measures. We discuss and show their usefulness.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Michael Gutmann. The long discussions and his feedback have been invaluable for the development of this work. I would also like to thank my friends and colleagues. Finally, a special word of appreciation to my family for their continuous support and encouragement.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Afonso Eduardo*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

In real-world applications, uncertainty is pervasive, and observed data are often assumed to have been generated according to unknown random phenomena. In turn, models are hypothesized descriptions of reality, allowing researchers and practitioners to better understand the underlying characteristics of such phenomena. In this setting, unobserved causes are typically modeled as latent random variables. Notably, there are two types of models: prescribed and implicit[1]. The former explicitly defines a function that measures the agreement between the latent variables and the observed data (likelihood function), whereas the latter is specified by a parametrized stochastic program, i.e. a stochastic simulator whose behavior is controlled by a set of input parameters. Simulator-based models are typically used in natural sciences. For instance, they have been shown to be useful in the study of infectious diseases [94], ecological systems [101] and population genetics [73]. The usefulness of these models lies in their ability to capture theories of real-world systems, unlike prescribed models whose structure is often too rigid.

Given a model, the question is then how to draw conclusions from observed data. Inferring conclusions is typically difficult in prescribed models, and even more so in simulator-based models where the likelihood function is defined implicitly by the simulator. In order to keep the problem tractable, approximate inference methods are therefore required.

[1] This motivation is to some extent based on my proposal.

Likelihood-free inference methods have emerged in the context of simulator-based models. However, conventional methods can be inefficient, having the need to query the simulator an inordinate amount of times. This in turn can be especially problematic if running the simulator is computationally expensive. In order to address this problem, Bayesian optimization for likelihood-free inference (BOLFI) has been recently proposed. A fundamental idea is that it is possible to cast the problem of inference as one of Bayesian optimization, where a probabilistic regression model is learned and then exploited so as to determine which queries are the most informative, leading to a procedure that is more efficient. Thus far, BOLFI has only been applied to relatively low-dimensional simulator-based models, i.e. simulators with a small number of input parameters. In this work, we tackle high-dimensional likelihood-free inference.

1.2 Contributions

In this work, we attempt to address some of the current problems in high-dimensional likelihood-free inference. One and arguably the most important problem is directly related to the inference task. In high-dimensional BOLFI, not only there is the need to estimate an accurate high-dimensional regression model, but also to be able optimize it effectively. Based on recent advances in high-dimensional Bayesian optimization, we introduce additional assumptions into the regression model. In particular, we consider an additive structure that must be learned. For the type of models we consider, we show that it is possible to learn such structure and that this process plays an important role in improving the efficiency of Bayesian optimization and likelihood-free inference. However, this is only one aspect of this work. High-dimensional likelihood-free inference is a relatively recent problem and *de facto* standards in terms of benchmarking have not yet been established. We explore performance measures that can scale to high-dimensional problems. The methods are tested on two scalable models that we have designed and evaluated according to the proposed performance measures. An ample discussion regarding their usefulness is provided.

1.3 Outline

In Chapter 2, we provide a broad view of the Bayesian approach to probabilistic inference, followed by a discussion of popular approximate inference methods in prescribed models. We then introduce the problem posed by simulator-based models and discuss how this relates to the previous case. Bayesian optimization and BOLFI are introduced at the end. In Chapter 3, we begin by providing a summary of previous work, and then restate the research questions, scope and objectives. In Chapter 4, we present and discuss the technical details of the methods and performance measures we propose. In Chapter 5, we discuss some of the practical challenges that have been encountered. We then motivate and present the models, followed by an ample discussion of the results. Finally, in Chapter 6, we summarize our work and make our concluding remarks, providing ideas for future work.

Chapter 2

Background

2.1 Preliminaries

Before proceeding further, it is important to note that our aim in this section is not to introduce the many formal definitions that arise in probability theory (and the more general measure theory), but only to provide enough context so that the reader can appreciate the development and contributions of this work. An extensive body of literature has been written about these subjects, and for more detailed expositions we refer the reader to some of these past publications [62, 100].

Probability theory and probabilistic modeling are central to the development of this work. In real-world applications, uncertainty is pervasive, and observed data are often assumed to have been generated according to unknown random phenomena. In turn, statistical models allow researchers and practitioners to better understand the underlying characteristics of such phenomena. These type of models can be fully specified by a set of random quantities or, equivalently, random variables.

Formally, a random variable x is defined as a function $x : \Omega \to \Omega_x$, where Ω denotes the sample space (set of all possible outcomes) and Ω_x is the measurable space. In certain cases, it suffices to assume that x is real valued, hence $\Omega_x = \mathbb{R}$. It is then typical to consider discrete random variables, whose range has countably many elements, or continuous random variables, in which case the range is uncountably infinite. In order

to characterize x, its cumulative distribution function (cdf) F_x should be specified:

$$F_x(a) = \Pr(\{\omega \in \Omega : x(\omega) \leq a\}) \tag{2.1}$$

$$= \Pr(x \leq a), \tag{2.2}$$

where Pr is the associated probability measure, mapping possible events (subsets of Ω_x) to a result in the interval $[0,1]$. In addition, if x is discrete, then the corresponding probability mass function (pmf) is as follows:

$$\pi_x(a) = \Pr(x = a). \tag{2.3}$$

Alternatively, if x is continuous, the probability density function (pdf) is defined such that:

$$F_x(a) = \int_{-\infty}^{a} \pi_x(r)dr, \tag{2.4}$$

which means that if the cdf F_x is differentiable, the pdf π_x can be obtained by differentiation. Therefore, the behavior of the random variable x can be fully characterized by direct specification of its pdf or pmf, π_x. In practice, it is also common to drop the subscript x when the function can be inferred from context. In particular, the notation $\pi(x)$ has become widespread.

At this point, it should be noted that random variables are not limited to being scalar valued. It is also useful to consider the measurable space to be a real vector space, allowing the definition of random vectors ($\Omega_x = \mathbb{R}^n$), matrices ($\Omega_x = \mathbb{R}^{m \times n}$) or, more generally, tensors. The usefulness of these representations lies in the fact that scalar-valued random variables can be jointly distributed according to a multivariate distribution that does not fully factorize and is consequently more expressive, as opposed to a set of scalar-valued random variables that are independent from one another. Since the differences can be easily identified in a given context, we refer to all these variables as simply random variables. For instance, we use standard vector notation $\boldsymbol{x} = (x_1, ..., x_n)$ to refer to a n-dimensional random variable.

2.2 Probabilistic Framework

Probability can have different interpretations [62]. One may adopt a frequentist view, where probability is defined as the relative frequency of occurrence of a certain event

or, alternatively, a Bayesian view, where it is seen as a degree of belief. The former is useful in the analysis of repeated trials, whereas the latter is arguably better at quantifying uncertainty. It is natural to discuss the plausibility of events that do not have long-run frequencies or are simply unrepeatable. Football fans may, for instance, argue about the probability of a given country winning the next World Cup, and indeed, reasonable statements can be made when viewing the probability as a degree of belief. More generally, one may formulate a model about a particular phenomenon and, since uncertainty is pervasive, a logical design decision is to consider that not only the observed data can be noisy, but also that the model parameters that might have generated such data are unknown random variables, as opposed to unknown fixed quantities. This is essentially the paradigm that Bayesian reasoning offers.

In this probabilistic framework, problems can be solved according to an iterative process where models are hypothesized, tested and revised at regular intervals [8]. There are thus three main stages:

1. *Modeling*: The model encodes the relationship between all observable and unobservable (latent) random variables. It can be prescribed by a joint probability distribution or defined implicitly by a parametrized stochastic program (simulator). This constitutes an hypothesis of the true, but unknown data generating process. Models should also contain all available prior information.

2. *Inference*: After observing data (evidence), conclusions need to be drawn. This is equivalent to inferring the posterior distribution, i.e. the conditional distribution of the variables of interest given the observed data. This step involves a computational task that in many cases is intractable and, consequently, approximations need to be made.

3. *Model criticism*: Models tend to be simplified descriptions of reality and are often wrong. It is important to assess to which degree they can explain the observed data and revise them accordingly. Explanatory models can be assessed by checking whether the generated data resembles the observed data using a specified set of test statistics (predictive checks). The performance of predictive models, whose aim is to make predictions, is instead evaluated on out-of-sample data (forecasting).

In this work, the focus is on the second step of this process, namely the computational task of inferring the posterior distribution. The difficulty of this step lies in the fact

that only in special circumstances this task can be performed exactly. In particular, simple models may yield expressions such that it is possible to determine the posterior distribution analytically. However, due to their simplicity, they may not be able to provide an accurate description of real phenomena. It is therefore necessary to consider more realistic models.

Let us define this problem more precisely. Consider a generic (Bayesian) model $\mathcal{M}$ that is given by the prior distribution $\pi(\boldsymbol{\theta})$ and the data generating process $f(\boldsymbol{y} \mid \boldsymbol{\theta})$. The former should contain all the information that is known about the variables of interest $\boldsymbol{\theta}$ before observing any data $\mathcal{D} = \boldsymbol{y}_o$, whereas the latter is the part of the model that is responsible for providing a plausible description of the underlying mechanism that has generated the observed data. The quantity $\mathcal{L}(\boldsymbol{\theta}) = f(\boldsymbol{y}_o \mid \boldsymbol{\theta})$ is known as the likelihood function and it measures the agreement between a particular value of $\boldsymbol{\theta}$ and the observed data. Since this is the quantity that actually takes into account the observed data, it plays a special role in probabilistic inference. Moreover, in prescribed models, the likelihood function can be evaluated directly, whereas in simulator-based models the likelihood function is only defined implicitly, requiring further approximations to be made. For ease of exposition, we assume here that $\mathcal{M}$ is a prescribed model. In Section 2.4, this assumption is relaxed. According to Bayes' theorem, the posterior distribution can then be written as:

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}_o) = \frac{\pi(\boldsymbol{y}_o, \boldsymbol{\theta})}{f(\boldsymbol{y}_o)} = \frac{f(\boldsymbol{y}_o \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})}{\int f(\boldsymbol{y}_o \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}}. \tag{2.5}$$

One possible interpretation of the above equation is that the prior information contained in the prior distribution is weighted according to its plausibility (likelihood). Hence, samples from the posterior distribution correspond to samples from the prior distribution that generate data that closely matches the data that has been observed. Equation (2.5) implicitly assumes the conditioning on model $\mathcal{M}$, which can alternatively be written as:

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{y}_o, \mathcal{M}) = \frac{f(\boldsymbol{y}_o \mid \boldsymbol{\theta}, \mathcal{M})\, \pi(\boldsymbol{\theta}, \mathcal{M})}{f(\boldsymbol{y}_o \mid \mathcal{M})} = \frac{f(\boldsymbol{y}_o \mid \boldsymbol{\theta}, \mathcal{M})\, \pi(\boldsymbol{\theta}, \mathcal{M})}{\int f(\boldsymbol{y}_o \mid \boldsymbol{\theta}, \mathcal{M})\, \pi(\boldsymbol{\theta}, \mathcal{M}) d\boldsymbol{\theta}}, \tag{2.6}$$

where the denominator is known as the model evidence (or marginal likelihood), a constant that measures the goodness of fit of model $\mathcal{M}$. Further application of Bayes' theorem allows one to perform model comparison:

$$\pi(\mathcal{M} \mid \boldsymbol{y}_o) = \frac{f(\boldsymbol{y}_o \mid \mathcal{M})\, \pi(\mathcal{M})}{\sum_{\mathcal{M}'} f(\boldsymbol{y}_o \mid \mathcal{M}')\, \pi(\mathcal{M}')}. \tag{2.7}$$

The posterior distribution is also important in predictive tasks, where for instance one may be interested in computing the predictive distribution of an observable:

$$\pi(y_{new} \mid \boldsymbol{y}_o, \mathcal{M}) = \int f(y_{new} \mid \boldsymbol{\theta}, \mathcal{M}) \pi(\boldsymbol{\theta} \mid \boldsymbol{y}_o, \mathcal{M}) d\boldsymbol{\theta}. \tag{2.8}$$

Finally, note that all the derived quantities that may be of interest in a particular application rely on the posterior distribution and, for this reason, it is important to estimate it accurately. In simple cases where the prior is conjugate for the likelihood function, the posterior distribution is in the same probability distribution family as the prior and its expression can be determined analytically. However, in more general settings, exact inference is unattainable and approximate inference methods must be used.

2.3 Likelihood-based Approximate Inference

In the previous section, we described the computational problem of inferring the posterior distribution. This problem stems from the fact that in order to obtain the normalization constant (model evidence), integration needs to be performed. In very low dimensions, simple numerical integration (quadrature) methods, such as the trapezoidal rule or Simpson's rule, provide reasonable approximations [71]. However, the cost to obtain an approximation scales exponentially with the number of dimensions. If the Bayesian model contains many variables of interest, then the integral is high dimensional, and the reliance on these simple integration methods is no longer viable.

Fortunately, there is a rich literature on how to evaluate high-dimensional integrals with respect to probability distributions. In fact, this field of research is so active and intricate that our aim is only to provide a brief overview of the most popular methodologies. A common theme to these methods is that there is always a trade-off between accuracy and computational cost. In what follows, we describe two different approaches. One is inherently stochastic, exploiting the idea of repeated random (Monte Carlo) sampling, whereas the other has roots in optimization and is, at least historically, deterministic. Again, it is important to stress that our description will be fairly superficial, focusing mostly on practical aspects and not necessarily on the theoretical results. For detailed reviews of this topic, we refer the reader to [9, 15, 62, 79, 80, 102].

2.3.1 Sampling-based Inference

Simple quadrature methods rely on a deterministic partition of the space in order to evaluate integrals. In general, each dimension is binned into a set of M bins and, consequently, if the integral is defined over a n-dimensional space, the number of bins becomes M^n. Instead, the fundamental idea of Monte Carlo integration is that integrand points can be chosen randomly. Consider the estimation of the following possibly high-dimensional integral, an expected value with respect to a probability distribution $\pi(\boldsymbol{\theta})$:

$$\mathrm{I} = \int g(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}, \tag{2.9}$$

where we assume $g(\boldsymbol{\theta})$ to be a function such that the integral is finite, i.e. $\mathrm{I} < \infty$. Then, a reasonably good approximation can be obtained by repeated random sampling from $\pi(\boldsymbol{\theta})$:

$$\mathrm{I} = \mathbb{E}[g(\boldsymbol{\theta})] \approx \frac{1}{N}\sum_{i=1}^{N} g(\boldsymbol{\theta}^{(i)}), \qquad \boldsymbol{\theta}^{(i)} \sim \pi(\boldsymbol{\theta}). \tag{2.10}$$

In fact, by the law of large numbers, as the number of samples increases, the approximation becomes increasingly closer to the true expected value I. Importantly, the estimator $\hat{\mathrm{I}}$ is unbiased and its variance does not depend on the dimension of $\boldsymbol{\theta}$:

$$\mathbb{E}[\hat{\mathrm{I}}] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} g(\boldsymbol{\theta}^{(i)})\right] = \mathbb{E}[g(\boldsymbol{\theta})] = Z, \tag{2.11}$$

$$\mathbb{V}[\hat{\mathrm{I}}] = \frac{\mathbb{V}[g(\boldsymbol{\theta})]}{N}. \tag{2.12}$$

One may think that this idea is applicable to any distribution $\pi(\boldsymbol{\theta})$. However, in some cases, it may be difficult to sample from $\pi(\boldsymbol{\theta})$, especially so if this distribution is only known up to a normalization constant $\pi^*(\boldsymbol{\theta})/Z$, e.g. an unnormalized posterior distribution. Importance sampling allows to solve this problem by introducing a proposal distribution $p(\boldsymbol{\theta})$. Estimation of the integral in Equation (2.9) then becomes:

$$\mathrm{I} = \int g(\boldsymbol{\theta})\frac{\pi^*(\boldsymbol{\theta})/Z}{p(\boldsymbol{\theta})}p(\boldsymbol{\theta})d\boldsymbol{\theta} = \mathbb{E}_{p(\boldsymbol{\theta})}\left[g(\boldsymbol{\theta})\frac{\pi^*(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\right] \Big/ \mathbb{E}_{p(\boldsymbol{\theta})}\left[\frac{\pi^*(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\right] \tag{2.13}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} w^{(i)} g(\boldsymbol{\theta}^{(i)}), \qquad w^{(i)} = \frac{\pi^*(\boldsymbol{\theta}^{(i)})/p(\boldsymbol{\theta}^{(i)})}{\frac{1}{N}\sum_{i=1}^{N}\pi^*(\boldsymbol{\theta}^{(i)})/p(\boldsymbol{\theta}^{(i)})}, \qquad \boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}), \tag{2.14}$$

where $w^{(i)}$ are known as the normalized importance weights. Interestingly, it can be shown that the variance of this estimator crucially depends on how close the proposal distribution $p(\boldsymbol{\theta})$ resembles the target distribution $\pi(\boldsymbol{\theta})$ [79].

In practical applications, it may be difficult to choose the proposal distribution, particularly in high dimensions. Fortunately, a family of algorithms known as Sequential Monte Carlo (SMC) attempt to solve this problem by continually adapting the proposal distribution [79]. A different class of algorithms, and arguably the most prevalent, is that of Markov chain Monte Carlo (MCMC). The basic idea of MCMC algorithms is that a sequence of dependent random variables can be generated according to a stochastic process such that, given sufficient time, its history can provide a good empirical approximation to the target distribution $\pi(\boldsymbol{\theta})$. This stochastic process is a Markov chain, where the distribution of random variables at a certain time only depends on its immediate past, as opposed to the entire past of the process. An important aspect is that not every Markov chain is appropriate and, in particular, it must satisfy the ergodic theorem so that it converges asymptotically to the target distribution [79]. In practice, this is implemented by specifying a suitable transition operator or transition kernel which can alternatevely be interpreted as a conditional distribution $p(\cdot \mid \cdot)$. In this setting, a fairly general framework is offered by Metropolis-Hastings (MH) (Algorithm 1) [38], where updates are given by the proposed conditional distribution $p(\cdot \mid \boldsymbol{\theta}^{(t-1)})$ and then, similar to importance sampling, corrected by a ratio of distributions. In this case, the ratio is given by the detailed balance equation:

$$\pi(\boldsymbol{\theta}^{(t-1)})p(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(t-1)}) = \pi(\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^{(t-1)} \mid \boldsymbol{\theta}^*), \tag{2.15}$$

such that the target distribution $\pi(\boldsymbol{\theta})$ is the unique stationary distribution of the process.

Algorithm 1 Metropolis-Hastings (MH)

Require: Initial value: $\boldsymbol{\theta}^{(0)}$, target distribution (possibly unnormalized): $\pi^*(\boldsymbol{\theta})$, proposed conditional distribution: $p(\cdot \mid \cdot)$, number of iterations: N

for $t = 1$ **to** N **do**

  $\boldsymbol{\theta}^* \sim p(\cdot \mid \boldsymbol{\theta}^{(t-1)})$ ▷ Generate proposal

  $u \sim U(0,1)$ ▷ Generate standard uniform distributed variable

  if $\min\left(1, \frac{\pi^*(\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)}{\pi^*(\boldsymbol{\theta}^{(t-1)})p(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}\right) < u$ **then**

    $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^*$ ▷ Accept proposal

  else

    $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}$ ▷ Reject proposal

The performance of the Metropolis-Hastings algorithm depends on the choice of the proposed conditional distribution. If this distribution is poorly chosen, the acceptance

ratio is low and, consequently, the efficiency of the algorithm is also poor. A typical choice for the proposed conditional distribution is the multivariate normal distribution centered at the previous value $p(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\theta}^{(t-1)}) = \mathcal{N}(\boldsymbol{\theta}^{(t)}; \boldsymbol{\theta}^{(t-1)}, \boldsymbol{\Sigma})$. The covariance matrix can then be defined as a diagonal matrix $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$, where σ is the step size and $\boldsymbol{I}$ is the identity matrix. In such case, since this distribution is symmetric, only dependent on the absolute difference between the proposed value $\boldsymbol{\theta}^*$ and the previous value $\boldsymbol{\theta}^{(t-1)}$, the acceptance rule simplifies, becoming $\min(1, \pi^*(\boldsymbol{\theta}^*)/\pi^*(\boldsymbol{\theta}^{(t-1)}))$. Tuning the step size is difficult in practice. Ideally, it should be large enough so that the space can be explored efficiently, but not too large as it would lead to a low acceptance rate.

A related algorithm that avoids the step size problem is Gibbs sampling [31]. The algorithm relies on the observation that if the proposed conditional distribution is a complete conditional distribution of the target distribution $\pi(\theta_i \mid \boldsymbol{\theta}_{-i})$[1], then the proposal is always accepted. In Gibbs sampling, it is therefore assumed that sampling from the joint distribution $\pi(\boldsymbol{\theta})$ is difficult, but that we can sample from each complete conditional distribution with ease. The order of the marginal updates does not matter, but a popular version is to update each random variable θ_i sequentially (Algorithm 2).

Algorithm 2 Sequential-scan Gibbs sampling

Require: Initial value: $\boldsymbol{\theta}^{(0)}$, complete conditional distributions: $\{\pi(\theta_i \mid \boldsymbol{\theta}_{-i})\}_{i=1}^{n}$, number of iterations: N

for $t = 1$ **to** N **do**

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(t-1)}$

  for $i = 1$ **to** n **do**

    $\theta_i \sim \pi(\cdot \mid \boldsymbol{\theta}_{-i})$ ▷ Generate proposal given current values

  $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}$

Naturally, several other versions exist. For instance, to alleviate the problem of slow mixing, where the Markov chain exhibits values that are strongly correlated, block updates are possible. Another possibility is to combine Metropolis-Hastings and Gibbs sampling, where the former can be used to sample from unnormalized conditional distributions [15, 79]. In any case, a common MCMC problem is the assessment of convergence to the target distribution $\pi(\boldsymbol{\theta})$. It is necessary to discard N_{bin} samples before the Markov chain has converged, and these are referred as burn-in. For the rest, a useful convergence diagnostic is the potential scale reduction factor (PSRF), where

[1] Recall that $\boldsymbol{\theta} = (\theta_1, ..., \theta_n)$. The vector $\boldsymbol{\theta}_{-i}$ corresponds to $\boldsymbol{\theta}$ without the ith dimension.

several Markov chains are used in order to compare the variability within each chain to the variability across chains [29, 30]. It is often defined as the ratio between the width of the 95% credible interval of all chains and the average width of the 95% credible interval in each chain. Values around 1 suggest convergence. Since samples are drawn from a Markov chain, another useful diagnostic is the effective sample size (ESS) that takes into account autocorrelation. ESS can be computed for each dimension i as:

$$\mathrm{ESS}_i = \frac{N}{1 + 2\sum_{k=1}^{\infty} \rho_i(k)}, \tag{2.16}$$

where $\rho_i(k)$ is the autocorrelation in the sequence $\{\theta_i^{(t)}\}_{t=N_{bin}}^{N}$ at lag k.

At this point, it should be mentioned that, while Metropolis-Hastings and Gibbs sampling are still widely used, more advanced algorithms have since been proposed. Many of the proposed techniques focus on accelerating the convergence of MCMC [80]. For instance, a wide class of algorithms extends the parameter space from which we wish to sample with auxiliary random variables. The motivation is that by clever design of such variables, the Markov chain can explore the space more efficiently. One such example is slice sampling [63, 79], but arguably the most popular method is Hamiltonian Monte Carlo (HMC) [15, 80].

In HMC, the space is augmented with random variables known as momentum, which exploit the local geometry via gradient of the target distribution. The name Hamiltonian stems from the fact that, in order to generate the updates, Hamiltonian dynamics are simulated in the resulting augmented space. The approximate dynamics are solved by inducing random behavior and by discretizing time (leapfrog method), which in turn introduces free parameters that need to be adjusted. In this context, a fairly robust algorithm that automatically calibrates the free parameters is the No-U-turn sampler (NUTS) [44]. It is, for instance, the default inference engine in Stan [16], an established probabilistic programming language.

2.3.2 Optimization-based Inference

MCMC algorithms are widely popular in probabilistic inference. Perhaps their most defining feature is that they are asymptotically unbiased, but convergence can be slow and is generally difficult to assess. Instead of sampling, it is possible to cast the problem of inference as one of optimization which tends to be faster and more amenable to parallelization. As a result, in recent years, optimization-based inference methods have

been steadily gaining impetus [9]. However, they are not a panacea. They require access to gradient information and, due to simplifying assumptions, the approximations that these methods provide can be imprecise. Moreover, this requirement has undesirable implications in the sense that it imposes constraints on the type of models that can be used. For instance, discrete distributions may need continuous relaxations or, alternatevely, the corresponding variables may need to be integrated out. In this section, we present the Laplace approximation and provide a brief overview of variational inference (VI). For recent reviews of the latter, we refer the reader to [9, 102].

Essentially, the Laplace approximation is a normal distribution approximation to the target distribution $\pi(\boldsymbol{\theta})$ [62]. The idea is based on the fact that the mode of the normal distribution corresponds to the mean and that the matrix of second-order partial derivatives (Hessian) of its negative log corresponds to the inverse of the covariance matrix. Indeed, for the multivariate normal distribution $\mathcal{N}(\boldsymbol{\theta};\boldsymbol{\mu},\boldsymbol{\Sigma})$, we have:

$$E_{\mathcal{N}}(\boldsymbol{\theta}) = -\log \mathcal{N}(\boldsymbol{\theta};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu})^{\intercal}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}) + const. \tag{2.17}$$

$$\boldsymbol{\mu} = \arg\min_{\boldsymbol{\theta}} E_{\mathcal{N}}(\boldsymbol{\theta}) \tag{2.18}$$

$$\boldsymbol{H}_{\mathcal{N}} = \frac{\partial^2 E_{\mathcal{N}}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\intercal}} = \boldsymbol{\Sigma}^{-1}. \tag{2.19}$$

Hence, for a distribution $\pi(\boldsymbol{\theta}) = \exp(-E(\boldsymbol{\theta}))/Z$, by performing a Taylor series expansion of $E(\boldsymbol{\theta})$ around its mode $\boldsymbol{\theta}^*$, we can obtain the mean and the covariance of the normal approximation [62]. In particular, we obtain the following approximation:

$$\pi(\boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{\theta},\boldsymbol{\theta}^*,\boldsymbol{H}(\boldsymbol{\theta}^*)^{-1}), \tag{2.20}$$

where

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} E(\boldsymbol{\theta}), \qquad \boldsymbol{H}(\boldsymbol{\theta}^*) = \left.\frac{\partial^2 E(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\intercal}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}. \tag{2.21}$$

The main advantage of this approximation is that it is computationally efficient, leading to fairly good approximations if the shape of the target distribution is approximately normal. On the other hand, approximating non-normal distributions possibly with several modes (multimodal) is likely to lead to a poor approximation. Nonetheless, a particularly successful extension of this idea for Bayesian hierarchical models is the Integrated Nested Laplace Approximation (INLA) [85], and more recent work explores INLA within MCMC [33].

Before proceeding further, a note on notation. Thus far, we have considered the target distribution to be a fairly general distribution, twice differentiable with respect to the

parameters but possibly unnormalized. For ease of exposition, in what follows, let us consider the posterior distribution, as defined in Equation (2.5), to be the target distribution.

Laplace approximation exploits a Taylor series expansion around the mode, ignoring the overall geometry of the target distribution. Variational inference (VI), on the other hand, addresses this problem by defining a global statistical measure that is used to match the approximation to the target distribution. In particular, variational inference first defines a family of distributions Q, and then tries to find a parametrization for the proposed variational distribution $q(\boldsymbol{\theta};\boldsymbol{\nu}) \in Q$ that resembles the target, where $\boldsymbol{\nu}$ are known as the variational parameters. Traditionally, the measure is the Kullback-Leibler (KL) divergence, being zero when the proposal $q(\boldsymbol{\theta};\boldsymbol{\nu})$ and the target $\pi(\boldsymbol{\theta} \mid \boldsymbol{y}_o)$ are identical and strictly positive otherwise. The optimization problem is thus defined as [9]:

$$q(\boldsymbol{\theta};\boldsymbol{\nu}^*) = \arg\min_{\boldsymbol{\nu}} \mathrm{KL}\left(q(\boldsymbol{\theta};\boldsymbol{\nu}) \,||\, \pi(\boldsymbol{\theta} \mid \boldsymbol{y}_o)\right) \tag{2.22}$$

$$= \arg\min_{\boldsymbol{\nu}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\nu})}\left[\log \frac{q(\boldsymbol{\theta};\boldsymbol{\nu})}{\pi(\boldsymbol{\theta} \mid \boldsymbol{y}_o)}\right]. \tag{2.23}$$

Due to the unknown normalization constant (marginal likelihood), this objective cannot however be solved directly. For this reason, traditional variational inference maximizes a lower bound, known as the evidence lower bound (ELBO) [9]:

$$\mathrm{ELBO} = \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\nu})}[\log \pi(\boldsymbol{y}_o, \boldsymbol{\theta})] - \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\nu})}[\log q(\boldsymbol{\theta};\boldsymbol{\nu})], \tag{2.24}$$

$$= \log f(\boldsymbol{y}_o) - \mathrm{KL}\left(q(\boldsymbol{\theta};\boldsymbol{\nu}) \,||\, \pi(\boldsymbol{\theta}|\boldsymbol{y}_o)\right). \tag{2.25}$$

Interestingly, notice that, since the KL divergence is non-negative and $\log f(\boldsymbol{y}_o)$ is constant, maximization of the lower bound is equivalent to the minimization of KL between the variational and target distributions. At this point, it should also be noted that the KL divergence is not symmetric. This particular formulation is known as the reverse KL and has the property of being zero forcing [62], i.e. regions where the target distribution is zero force the approximation to be zero, which in turn leads to the well-known problem of variance underestimation [9]. Naturally, this has encouraged the use of alternative divergences [39, 54, 75].

Recently, variational inference has perhaps been the approximate inference method that has received the most support, at least from the machine learning community. Indeed, a significant amount of extensions have been proposed. For instance, some

innovations consider the use of unbiased Monte Carlo estimators and stochastic optimization to approximate and maximize the ELBO, allowing variational inference to scale to massive data and to cope with generic non-conjugate models [42, 51, 74, 82]. In order to further accelerate approximate inference and to improve its accuracy, others also exploit parametrizations and transformations given by deep neural networks [49, 78]. Finally, there is a rich literature on likelihood-based approximate inference methods, but the field as a whole has yet to mature. A potential direction is to investigate deeper the possible connections between sampling- and optimization-based methods [58, 86, 95].

2.4 Approximate Bayesian Computation

In Section 2.3, we have assumed that the likelihood function is explicitly available and can be evaluated[2]. We now consider the more challenging scenario where the likelihood function is only defined implicitly. In particular, the focus is on parametrized stochastic programs, also known as simulator-based models or implicit generative models [55, 61]. The simulator allows the generation of data conditioned on parameters $\boldsymbol{\theta}$, i.e. $\boldsymbol{y}_{\boldsymbol{\theta}} \sim f(\cdot|\boldsymbol{\theta})$. Moreover, recalling the probabilistic framework described in Section 2.2, these parameters $\boldsymbol{\theta}$ are to be treated as random variables, whose prior information is summarized by $\pi(\boldsymbol{\theta})$. Given observed data $\boldsymbol{y}_o$, the goal is then to find an approximation to the true posterior $\pi(\boldsymbol{\theta} \mid \boldsymbol{y}_o)$, whose "normalization constant" may possibly be dependent on $\boldsymbol{\theta}$ (partition function):

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{y}_o) = \frac{f(\boldsymbol{y}_o \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{Z_{\boldsymbol{\theta}}}. \tag{2.26}$$

Likelihood-free inference is an umbrella term that encompasses all approximate inference methods that can be applied in this setting. As argued in [89], the term is perhaps a misnomer. Probabilistic inference crucially relies on the information provided by the likelihood function. However, in this case, querying the simulator only provides

[2]Direct evaluation of the likelihood function may still be undesirable. Consider for instance a massive dataset. In such case, to avoid the computational burden of evaluating the likelihood function explicitly, unbiased estimators can often be designed to approximate intractable factors. Nevertheless, the analytical formula for the likelihood function is available. In the context of VI, this is known as stochastic VI [42]. Closely related is stochastic gradient VI [50, 74]. Alternatively, the likelihood function may be unavailable, but unbiased estimators may be known. In the context of MCMC, such methods are known as pseudo-marginal MCMC [4]. In either case, the common assumption is the availability of unbiased estimators.

partial information. In this section, we focus on a particular subset of such methods, collectively known as approximate Bayesian computation (ABC).

Although not exclusive to ABC, the basic idea is that an approximation to the likelihood function can be obtained by exploiting its very definition. The likelihood function measures the plausibility that a given parameter configuration $\boldsymbol{\theta}$ has generated the observed dataset $\boldsymbol{y}_o \in \mathcal{Y}$. Since sampling is available, then the estimator can rely on repeated simulation of datasets $\boldsymbol{y}_{\boldsymbol{\theta}} \in \mathcal{Y}$ and compare them to the observed data in order to provide a reasonable approximation. Arguably, the ABC inference mechanism can best be described by introduction of a simple algorithm known as rejection ABC (Algorithm 3) [73]. The algorithm begins by generating candidate values from the prior distribution, followed by conditional generation of a synthetic dataset. Since the datasets may be high dimensional, it is often necessary to first project them to a lower-dimensional space, which is achieved by an appropriate choice of summary statistics $S : \mathcal{Y} \to \mathcal{S}$. A discrepancy $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ and a threshold ε dictate whether the proposed value is accepted.

Algorithm 3 Rejection ABC

Require: prior: $\pi(\boldsymbol{\theta})$, simulator: $f(\boldsymbol{y} \mid \boldsymbol{\theta})$, discrepancy: $d(\cdot)$, summary statistics: $S(\cdot)$, observed data: $\boldsymbol{y}_o$, threshold: ε, number of iterations: N

for $i = 1$ **to** N **do**

repeat

$\boldsymbol{\theta} \sim \pi(\cdot)$

$\boldsymbol{y}_{\boldsymbol{\theta}} \sim f(\cdot \mid \boldsymbol{\theta})$

until $d(S(\boldsymbol{y}_{\boldsymbol{\theta}}), S(\boldsymbol{y}_o)) \leq \varepsilon$

$\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}$

In rejection ABC, the approximate posterior distribution is obtained by the empirical distribution formed by the accepted samples. Hence, the exact form of the posterior approximation can be written as [89]:

$$\pi_{ABC}(\boldsymbol{\theta} \mid \boldsymbol{s}_o) = \int \pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s} \mid \boldsymbol{s}_o) d\boldsymbol{s} \propto \int \mathbb{1}\left(d(\boldsymbol{s}, \boldsymbol{s}_o) \leq \varepsilon\right) f(\boldsymbol{s} \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{s}, \tag{2.27}$$

where $\mathbb{1}$ denotes the indicator function, $\boldsymbol{s}$ and $\boldsymbol{s}_o$ are the generated and observed vectors of summary statistics. In particular, $f(\boldsymbol{s} \mid \boldsymbol{\theta}) = S(f(\boldsymbol{y} \mid \boldsymbol{\theta}))$. Interestingly, notice that if we let S be the identity function, or a function that computes sufficient statistics, and set $\varepsilon \to 0$, then in the limit we recover the true posterior:

$$\pi_{ABC}(\boldsymbol{\theta} \mid \boldsymbol{y}_o) \propto f(\boldsymbol{y}_o \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta}). \tag{2.28}$$

Furthermore, the indicator function in Equation (2.27) can be viewed as a uniform kernel of bandwidth ε. This, in turn, allows the ABC likelihood function to be interpreted as a nonparametric approximation to the true likelihood function, specifically a kernel density estimator [89]:

$$f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{\theta}) = \int K_\varepsilon \left(d(\boldsymbol{s}, \boldsymbol{s}_o)\right) f(\boldsymbol{s} \mid \boldsymbol{\theta}) d\boldsymbol{s}, \tag{2.29}$$

where K_ε is the uniform kernel, but more generally is a kernel[3] with bandwidth ε.

2.4.1 ABC Samplers

Thus far, we have presented rejection ABC. Despite being a simple algorithm, it has been applied in some contexts with relative success [55]. However, unless the prior $\pi(\boldsymbol{\theta})$ is targeting high density regions of the approximate likelihood function, the overall sampling process can be very inefficient, especially in high-dimensional problems. Naturally, this led to the development of samplers where the proposal distribution is adaptive[4].

An important ABC sampler is based on Markov chain Monte Carlo (MCMC) [25]. Instead of directly targeting the (marginal) ABC posterior $\pi_{ABC}(\boldsymbol{\theta} \mid \boldsymbol{s}_o)$, as given by Equation (2.27), the algorithm uses the joint ABC posterior distribution $\pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s} \mid \boldsymbol{s}_o)$ as target. Originally developed in [59], MCMC-ABC must thus satisfy the detailed balance equation with respect to this new target distribution. The overall structure is similar to Algorithm 1, except that after generating a candidate $\boldsymbol{\theta}^*$, it is necessary to simulate a synthetic dataset and reduce it to a vector of summary statistics $\boldsymbol{s}_{\boldsymbol{\theta}^*}$. At time t, the candidate pair $(\boldsymbol{\theta}^*, \boldsymbol{s}_{\boldsymbol{\theta}^*})$ is then accepted with probability:

$$p_\alpha = \min\left(1, \frac{K_\varepsilon \left(d(\boldsymbol{s}_{\boldsymbol{\theta}^*}, \boldsymbol{s}_o)\right) \pi(\boldsymbol{\theta}^*) p(\boldsymbol{\theta}^{(t-1)} \mid \boldsymbol{\theta}^*)}{K_\varepsilon \left(d(\boldsymbol{s}_{\boldsymbol{\theta}^{(t-1)}}, \boldsymbol{s}_o)\right) \pi(\boldsymbol{\theta}^{(t-1)}) p(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(t-1)})}\right). \tag{2.30}$$

Due to the nonparametric approximation to the likelihood function (Equation (2.29)), a common problem in ABC is that a value for the bandwidth ε needs to be selected. Values that are too large yield a high acceptance ratio, but the quality of the approximation becomes poor. On the other hand, if the value is set too small, only a few candidates are accepted and, as a result, the Monte Carlo error dominates. As described in Section 2.3.1, one possible strategy is to augment the state space of the Markov chain

[3] A generalization of Algorithm 3 consists in the modification of the acceptance rule so as to introduce a different kernel. Alternatively, d can itself be interpreted as a multivariate kernel.

[4] Recall that this problem has been described previously in Section 2.3.1 for prescribed models.

by introducing ε as an auxiliary variable, and indeed such approach can improve the mixing rate [12, 77]. However, due to the need to maintain an ABC posterior as the unique stationary distribution of the chain, MCMC samplers are limited in how they can adaptively set the bandwidths.

In the context of ABC, an arguably more popular sampler is Sequential Monte Carlo (SMC). The SMC-ABC requires the definition of a sequence of decreasing bandwidths and at each epoch, it uses the accepted samples from the previous epoch to continually adapt the proposal distribution (by convention, a mixture of normal distributions). This approach has been effective because during the first epochs the algorithm has yet to learn a proposal distribution that focuses on the high density region of the approximate ABC posterior. However, at each new epoch, the proposal distribution is able to provide an increasingly better approximation. A listing of the original algorithm can be found in [6]. More recent extensions automatically define the sequence of bandwidths, allow the specification of different kernels and also monitor the effective sample size in order to avoid the problem of sample or particle degeneracy [18]. For an extensive review of SMC-ABC and other ABC samplers, we refer the reader to [25].

2.4.2 High-dimensional ABC

Part of the motivation behind the design of better ABC samplers is high-dimensional ABC, i.e. inference in high-dimensional model-based simulators where the number of parameters to estimate is large. Indeed, if the parameter vector is high dimensional, then an informative vector of summary statistics typically also needs to be high dimensional. In such case, conventional ABC samplers struggle to generate data whose summary statistics closely match the observed summary [65]. In this setting, post-sampling correction strategies may help to improve the quality of the approximations.

Originally proposed in [7], one such strategy is regression adjustment. Qualitatively, the idea is that the discrepancy associated with each accepted posterior sample is less than ε, but not necessarily zero. In order to obtain a better approximation, it is possible to fit a linear or nonlinear regression model that takes the vector of summary statistics $\boldsymbol{s}_{\boldsymbol{\theta}}$ as covariate and the parameter vector $\boldsymbol{\theta}$ as response [10]. If the model is able to provide a reasonable explanation of this relationship, then the accepted samples can be adjusted based on the residuals so as to obtain a new empirical distribution in which the associated discrepancies are zero.

Regression adjustment is a general strategy that can be applied whether the simulator-based model is high or low dimensional. On the other hand, marginal adjustment is specifically designed to enable high-dimensional ABC [64], where due to the curse of dimensionality direct estimation of the joint posterior may yield a poor approximation. The strategy consists in first obtaining an estimate of the joint posterior by conventional ABC, possibly with regression adjustment. The marginal distributions of this estimate are then adjusted according to univariate marginals whose estimation occurs separately. The motivation is that lower-dimensional distributions can be estimated more accurately.

The marginal adjustment strategy is promising, but has limitations. In particular, it does not explicitly take into account the dependence structure of the target joint posterior. Hence, a possible improvement is then to assume that the dependence structure can be estimated with a suitable model. In this context, copula models are particularly convenient since any multivariate distribution is composed of a copula and a set of marginal distributions (Sklar's theorem) [90]. Motivated by asymptotic behavior, recent work considers the use of a Gaussian copula, resulting in an approximation that only requires the estimation of bivariate marginal distributions [53]. The results are inspiring, but open challenges remain. One is related to the possibility of not being able to recover posteriors with more complex dependence structure [65]. Another is that it crucially relies on the assumption that suitable sets of test statistics can be identified for each dimension. Previous work tries to automate the process of constructing summary statistics, but still requires human intervention [26]. This in turn poses a broader question of whether black-box ABC is feasible.

2.4.3 Related Approaches

Thus far, we have discussed the ABC principles, namely its approximation to the true posterior and how the process can alternatively be seen as proposing an approximate nonparametric likelihood function. The quality of the approximation crucially relies on several factors that require calibration: summary statistics, discrepancy, kernel and threshold / bandwidth. A good choice of summary statistics is particularly important and, as a result, it hinges on domain knowledge in most real data analyses, e.g. [66, 94]. The kernel function is often the uniform kernel and the discrepancy typically

is the Mahalanobis distance[5], with the Euclidean distance being a special case [89]. The threshold is either chosen adaptively using more advanced ABC samplers or fixed based on computational aspects. For instance, it can be set to a value such that the acceptance ratio is 1%.

Tuning the threshold to obtain a sensible approximation is admittedly difficult. Fortunately, there are alternatives to the ABC likelihood that may in certain circumstances be more attractive [20, 21]. Motivated by the fact that summary statistics are often obtained via averaging and that the central limit theorem may apply, synthetic likelihood[6] offers a parametric approximation where the vector of summary statistics is assumed to be conditionally normally distributed [101]. Consequently, the approximation can be written as:

$$\mathcal{L}(\boldsymbol{\theta}) = f(\boldsymbol{y}_o \mid \boldsymbol{\theta}) \approx f(\boldsymbol{s}_o \mid \boldsymbol{\theta}) \overset{\propto}{\sim} \mathcal{N}(\boldsymbol{s}_o; \boldsymbol{\mu}_{\boldsymbol{s}}(\boldsymbol{\theta}), \boldsymbol{\Sigma}_{\boldsymbol{s}}(\boldsymbol{\theta})), \tag{2.31}$$

where $\overset{\propto}{\sim}$ means approximately proportional. For any given $\boldsymbol{\theta}$, both the mean $\boldsymbol{\mu}_{\boldsymbol{s}}(\boldsymbol{\theta})$ and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{s}}(\boldsymbol{\theta})$ need to be estimated, which is typically done by simulating multiple datasets and computing the sample mean and sample covariance of the resulting summary statistics. However, a potential problem[7] of this approach is that the number of simulated datasets M needs to scale linearly with the dimensionality of the vector of summary statistics, otherwise the covariance matrix is singular. This can be problematic in high-dimensional settings, where the vector of summary statistics is high dimensional. On the other hand, recalling that the ABC likelihood is given by Equation (2.29), the corresponding Monte Carlo estimator is:

$$f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{\theta}) \approx \frac{1}{M} \sum_{j=1}^{M} K_\varepsilon \left(d(S(\boldsymbol{y}_{\boldsymbol{\theta}}^{(j)}), \boldsymbol{s}_o) \right), \qquad \boldsymbol{y}_{\boldsymbol{\theta}}^{(j)} \sim f(\cdot \mid \boldsymbol{\theta}), \tag{2.32}$$

which does not necessarily impose a constraint on M. In fact, most ABC samplers we discussed use $M = 1$, and this may even be optimal [11]. It can be argued that prior information can solve the ill-conditioning problem that occurs in synthetic likelihood and, indeed, recent work explores the use of shrinkage estimators [2, 68], but at the cost of new free parameters that require tuning. Furthermore, it seems to us that if the objective is to first explore the parameter space so as to find regions of non-negligible

[5] Interestingly, recent work proposes measuring the discrepancy via classification [37].

[6] Synthetic likelihood is a special case of indirect inference, where an auxiliary model is used in the estimation process [20]. The relation with the nonparametric approximation is explored in [36].

[7] In addition to the one where the estimator is biased if the summary statistics are not normally distributed.

density, then noisier estimates obtained with small M may provide a sensible solution. In any case, a good choice for M should take into account prior information on how noisy the simulations may be, which realistically can depend on the parameters.

Finally, in a previous note, we alluded to the fact that methods such as pseudo-marginal MCMC and stochastic gradient VI are applicable in the presence of unbiased estimators of the true likelihood. The latter in particular requires unbiased estimates of the gradient. In turn, the Monte Carlo estimator of the ABC likelihood is unbiased and an unbiased estimator of the synthetic likelihood can be designed [72]. Given these unbiased estimates of the approximate likelihood, recent work focuses thus on applying these methods to estimate the approximate posterior [21]. Importantly, and as discussed previously, only in special circumstances the approximate posterior corresponds to the true posterior.

2.5 Bayesian Optimization

An important theme in this work is Bayesian optimization (BO). As the name suggests, the primary goal of BO is that of finding the global minimum of an objective function $g : \Theta \to \mathbb{R}$:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\Theta} g(\boldsymbol{\theta}). \tag{2.33}$$

However, unlike standard optimization [13], it relies on learning a probabilistic model that is then exploited so as to determine candidate points that are likely to minimize the target function [91]. Compared to standard approaches, the decision process is computationally more demanding, but tends to be more informative. In particular, typical optimization methods only rely on the local geometry of the objective function, whereas BO is able to use all the acquired information in the decision process. This, in turn, allows to minimize the number of function evaluations. Hence, the method is particularly relevant in circumstances where the cost of evaluating the objective function dominates. BO has been successfully applied in settings where the function is treated as a black-box, i.e. it may be non-convex and access to gradient information is unavailable. Most notably, it is used to optimize architectures of deep neural networks, whose training time may range from hours to weeks. For a relatively recent review of this far-reaching topic, we refer the reader to [88].

A general BO procedure is shown in Algorithm 4. It crucially relies on the choice of the statistical model and the acquisition function.

Algorithm 4 Bayesian Optimization (BO)

Require: objective function: g, acquisition function: α, statistical model: $\mathcal{M}$, evidence set: $\mathcal{E}^{(t_0)} = \left\{ \left(\boldsymbol{\theta}^{(j)}, g^{(j)} \right) \right\}_{j=1}^{t_0}$

for $t = t_0$ **to** ... **do**

$\boldsymbol{\theta}^{(t+1)} = \arg\min_{\boldsymbol{\theta} \in \Theta} \alpha(\boldsymbol{\theta} \mid \mathcal{E}^{(t)}, \mathcal{M})$ ▷ Find minimizer of the acq. function

$g^{(t+1)} = g(\boldsymbol{\theta}^{(t+1)})$ ▷ Query objective function

$\mathcal{E}^{(t+1)} = \mathcal{E}^{(t)} \cup \left\{ \left(\boldsymbol{\theta}^{(t+1)}, g^{(t+1)} \right) \right\}$ ▷ Update evidence set

In general, there is no restriction on the type of model as long as it can reasonably represent the prior assumptions about the objective function, e.g. smoothness. Due to its flexibility and mathematical convenience, Gaussian process regression is typically used in this context. The Gaussian process is a stochastic process that defines a distribution over functions such that any finite set of function values is normally distributed [76]. A mean function $m(\boldsymbol{\theta})$ and a covariance function $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ are required in order to define such process, i.e. $\mathcal{GP}(m,k)$. The covariance function must be a positive definite kernel, while the mean function can in principle be any function [76]. A common choice for the mean function is to assume that data are centered, hence $m(\boldsymbol{\theta}) = 0$. Alternatively, previous work [36], considers the mean function to be a sum of convex quadratic polynomials, without cross terms:

$$m(\boldsymbol{\theta}) = \sum_j a_j \theta_j^2 + b_j \theta_j + c, \tag{2.34}$$

where each a_j is assumed positive since the objective involves minimization. On the other hand, a standard choice for the covariance function is the automatic relevance determination (ARD) squared exponential kernel [76]:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_f^2 \exp\left(-\sum_j \frac{1}{\lambda_j^2} (\theta_j - \theta_j')^2 \right), \tag{2.35}$$

where σ_f^2 is known as the signal variance and λ_j is the lengthscale in the jth dimension. Intuitively, the square root of the signal variance controls the typical range of values the function may take and each lengthscale controls the fluctuations of the function along dimension j. For instance, a small lengthscale indicates that the function may have high frequency components along that particular dimension. Consequently, if

there is reason to believe that the behavior of the function is roughly the same in all dimensions, then a single shared lengthscale may be more appropriate.

As in all Bayesian approaches, a central object is the posterior distribution:

$$\pi(\boldsymbol{g} \mid \Phi_t, \boldsymbol{g}_t, \mathcal{M}) \propto f(\boldsymbol{g}_t \mid \Phi_t, \boldsymbol{g}, \mathcal{M})\pi(\boldsymbol{g} \mid \mathcal{M}), \tag{2.36}$$

where $\pi(\boldsymbol{g} \mid \mathcal{M})$ is the prior distribution of function values $\boldsymbol{g}$ under a particular Gaussian process model $\mathcal{M}$, and $\boldsymbol{g}_t$, Φ_t correspond to the observed function values and respective locations (evidence set), i.e. $\boldsymbol{g}_t = (g^{(1)}, ..., g^{(t)})^\intercal$, $\boldsymbol{\Phi}_t = \{\boldsymbol{\theta}^{(j)}\}_{j=1}^t$. Assuming additive white Gaussian noise with σ_n^2, the likelihood function can be written as

$$f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \boldsymbol{g}, \mathcal{M}) = \mathcal{N}(\boldsymbol{g}_t; \boldsymbol{g}, \sigma_n^2 \boldsymbol{I}) \tag{2.37}$$

and, due to conjugacy, the posterior is itself a Gaussian process [76]:

$$\boldsymbol{g} \mid \mathcal{E}^{(t)}, \mathcal{M} \sim \mathcal{GP}(m_\pi, k_\pi), \tag{2.38}$$

where

$$m_\pi(\boldsymbol{\theta}) = m(\boldsymbol{\theta}) + \boldsymbol{k}_t(\boldsymbol{\theta})^\intercal [\boldsymbol{K}_t + \sigma_n^2 \boldsymbol{I}]^{-1} (\boldsymbol{g}_t - \boldsymbol{m}_t), \tag{2.39}$$

$$k_\pi(\boldsymbol{\theta}, \boldsymbol{\theta}') = k(\boldsymbol{\theta}, \boldsymbol{\theta}') - \boldsymbol{k}_t(\boldsymbol{\theta})^\intercal [\boldsymbol{K}_t + \sigma_n^2 \boldsymbol{I}]^{-1} \boldsymbol{k}_t(\boldsymbol{\theta}'), \tag{2.40}$$

$$\boldsymbol{k}_t(\boldsymbol{\theta}) = \left(k(\boldsymbol{\theta}, \boldsymbol{\theta}^{(1)}), ..., k(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) \right)^\intercal, \tag{2.41}$$

$$\mathbf{m}_t = \begin{pmatrix} m(\boldsymbol{\theta}^{(1)}) \\ \vdots \\ m(\boldsymbol{\theta}^{(t)}) \end{pmatrix}, \qquad \mathbf{K}_t = \begin{pmatrix} k(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(1)}) & \dots & k(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(t)}) \\ \vdots & & \vdots \\ k(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(1)}) & \dots & k(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) \end{pmatrix}. \tag{2.42}$$

In this particular context, namely BO, the Gaussian process regression model is used as a surrogate model of the objective function. Consequently, the interest lies further in the posterior predictive distribution, which can be shown to be [76]:

$$\pi(g(\boldsymbol{\theta}) \mid \boldsymbol{\theta}, \mathcal{E}^{(t)}) = \mathcal{N}(g(\boldsymbol{\theta}); \mu_t(\boldsymbol{\theta}), v_t(\boldsymbol{\theta}) + \sigma_n^2), \tag{2.43}$$

where

$$\mu_t(\boldsymbol{\theta}) = m_\pi(\boldsymbol{\theta}), \qquad v_t(\boldsymbol{\theta}) = k_\pi(\boldsymbol{\theta}, \boldsymbol{\theta}). \tag{2.44}$$

In BO, the goal is thus to propose a candidate point based on an acquisition function that encodes a trade-off between exploitation and exploration. The key observation is

that a pure greedy approach (exploitation) is likely to lead to suboptimal results, since the function may be non-convex. Minimizing the posterior predictive mean corresponds to a greedy approach, whereas choosing the point that maximizes the posterior predictive variance (uncertainty) is tied to exploration. Several acquisition functions have been proposed, and the investigation of their properties constitutes an ongoing area of research [88]. For instance, previous work explores a portfolio of acquisition functions and shows that it can perform better than a single best acquisition function [43]. Of particular relevance to this work is the lower confidence bound selection criterion (LCBSC) [14, 56, 92][8]:

$$\alpha(\boldsymbol{\theta} \mid \mathcal{E}^{(t)}, \mathcal{M}) = \mu_t(\boldsymbol{\theta}) - \sqrt{\eta_t^2 v_t(\boldsymbol{\theta})}, \tag{2.45}$$

where

$$\eta_t^2 = 2\log[t^{2n+2}\pi^2/(3\varepsilon_\eta)], \tag{2.46}$$

and $\varepsilon_\eta = 0.1$, $n = |\boldsymbol{\theta}|$. The interpretation of this rule is that it chooses points whose function values are likely to yield a lower confidence bound [92]. Interestingly, note that as the number of acquisition increases, η_t^2 becomes larger. Increasing n also leads to a larger weight.

2.6 Bayesian Optimization for Likelihood-free Inference

Thus far, we presented the Bayesian approach to probabilistic inference and the importance of the posterior distribution. We described the problem of exact inference in realistic models which in turn leads to the use of approximate methods. Several popular methods for prescribed model were discussed. The problem posed by simulator-based models was then introduced and we mentioned that approximate inference in this setting is known as likelihood-free inference. Our primary focus was on approximate Bayesian computation (ABC) methods, but we also discussed alternatives such as synthetic likelihood. In either case, the true, but implicit likelihood function is replaced by an approximation for which unbiased estimators can be designed. We concluded by observing that these estimators allow to exploit approximate inference methods whose initial purpose was to approximate intractable factors of the true likelihood. An important difference however is that in likelihood-free inference the target is an approximate

[8] As pointed in [56], the formula for η_t^2 presented in [14] may be incorrect. This is the suggested correction based on [92].

posterior distribution, since it relies on a approximation to the likelihood function. In this section, we finally present the Bayesian optimization for likelihood-free inference (BOLFI) approach [36].

In BOLFI, the authors exploit the fact that the problem of approximating the likelihood function is one of conditional density estimation. A key observation is that for convex kernel functions K_ε, a lower bound for the nonparametric approximation in Equation (2.29) can be found by application of Jensen's inequality:

$$f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{\theta}) = \mathbb{E}_{f(\boldsymbol{s}|\boldsymbol{\theta})}[K_\varepsilon(\Delta_{\boldsymbol{\theta}})] \tag{2.47}$$

$$\geq K_\varepsilon\left(\mathbb{E}_{f(\boldsymbol{s}|\boldsymbol{\theta})}[\Delta_{\boldsymbol{\theta}}]\right) \tag{2.48}$$

where $\Delta_{\boldsymbol{\theta}} = d(\boldsymbol{s}, \boldsymbol{s}_o)$ is the discrepancy. Similarly, for non-convex kernel functions, in particular uniform kernels, a lower bound can be derived according to Markov's inequality:

$$f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{\theta}) = c\,\mathbb{E}_{f(\boldsymbol{s}|\boldsymbol{\theta})}[\mathbb{1}\left(\Delta_{\boldsymbol{\theta}} \leq \varepsilon\right)] \tag{2.49}$$

$$= c\Pr\left(\Delta_{\boldsymbol{\theta}} \leq \varepsilon \mid \Delta_{\boldsymbol{\theta}} = d(\boldsymbol{s}_{\boldsymbol{\theta}}, \boldsymbol{s}_o), \boldsymbol{s}_{\boldsymbol{\theta}} \sim f(\boldsymbol{s} \mid \boldsymbol{\theta})\right) \tag{2.50}$$

$$\geq c\left[1 - \frac{1}{\varepsilon}\mathbb{E}_{f(\boldsymbol{s}|\boldsymbol{\theta})}[\Delta_{\boldsymbol{\theta}}]\right] \tag{2.51}$$

where c is a positive constant. Hence, if the probabilistic relationship between the discrepancy $\Delta_{\boldsymbol{\theta}}$ and the parameter vector $\boldsymbol{\theta}$ can be estimated, it is possible to design an approximation to the likelihood function. Importantly, unlike typical ABC methods, it is possible to include in the probabilistic regression model any prior information regarding the behavior of the discrepancy, e.g. smoothness, range, observation noise type, allowing in turn to improve the statistical efficiency of the estimation process. In addition, since high density regions of the approximate likelihood function correspond to small discrepancies, it is also possible and computationally advantageous to actively focus on such regions so as to approximate them more precisely.

Thus, in BOLFI, the relationship between the parameter $\boldsymbol{\theta}$ and discrepancy $\Delta_{\boldsymbol{\theta}}$ is modeled probabilistically. For convenience, the observed discrepancies are assumed to be subject to additive white Gaussian noise, which in turn allows to exploit the conjugacy properties of a Gaussian process prior. The active sampling process is then determined by Bayesian optimization. Interestingly, note that once enough data has been acquired, it possible to provide different approximations to the approximate nonparametric likelihood by considering different kernels and bandwidths. For instance,

assuming a uniform kernel with bandwidth ε, the model-based approximation can be evaluated as:

$$f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{\theta}) \overset{\propto}{\sim} \Pr\left(\Delta_{\boldsymbol{\theta}} \leq \varepsilon \mid \Delta_{\boldsymbol{\theta}} \sim \mathcal{N}(\mu_t(\boldsymbol{\theta}), v_t(\boldsymbol{\theta}) + \sigma_n^2)\right) \tag{2.52}$$

$$= F_{\mathcal{N}}\left(\frac{\varepsilon - \mu_t(\boldsymbol{\theta})}{\sqrt{v_t(\boldsymbol{\theta}) + \sigma_n^2}}\right), \tag{2.53}$$

where $F_{\mathcal{N}}$ denotes the standard normal cumulative distribution function. Alternatively, since the discrepancies are non-negative, one may choose to model the log discrepancies, in which case it follows that:

$$f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{\theta}) \overset{\propto}{\sim} \Pr\left(\log \Delta_{\boldsymbol{\theta}} \leq \log \varepsilon \mid \log \Delta_{\boldsymbol{\theta}} \sim \mathcal{N}(\mu_t(\boldsymbol{\theta}), v_t(\boldsymbol{\theta}) + \sigma_n^2)\right) \tag{2.54}$$

$$= F_{\mathcal{N}}\left(\frac{\log \varepsilon - \mu_t(\boldsymbol{\theta})}{\sqrt{v_t(\boldsymbol{\theta}) + \sigma_n^2}}\right). \tag{2.55}$$

Furthermore, the approximate posterior can be found by using any of the methods discussed in previous sections. In particular, since this approximation is differentiable, methods that rely on gradients can be used.

At this point, however, a natural question that arises is whether standard acquisition functions are appropriate in this setting, since the points are chosen in a deterministic fashion[9]. The authors consider a stochastic acquisition function where the minimizer, determined by LCBSC, is injected with normal distributed perturbations. The strategy seems to work well for the low-dimensional test cases that are presented. More recent work proposes rules that are specifically designed for better approximating the approximate posterior, e.g. minimization of the expected uncertainty [45, 52]. The authors provide examples where the number of parameters ranges from 2 to 10 and conclude that for higher-dimensional problems deterministic rules may perform better. This seems intuitive to us since the volume of the space increases exponentially with the number of dimensions. Hence, the major difficulty is that of finding the region of small discrepancies. Increasing the exploration by adding a stochastic rule only seems advantageous if the the region of small discrepancies was already found. In this sense, an hybrid approach that switches between rules may provide good results.

Finally, it should be noted that the BOLFI framework is not limited to approximating the nonparametric likehood approximation. The authors explore synthetic likelihood,

[9]The assumption is that it is possible to find the global optimum of the acquisition function, which may not be true, particularly if the function is itself high dimensional and non-convex.

but, in principle, BOLFI can emulate any approximation[10] to the implicit likelihood function. The key observation is that such approximate likelihood estimates are obtained via simulation, hence noisy. In turn, the negative log of these quantities can be interpreted as log discrepancies. For instance, consider the case of synthetic likelihood whose estimates are necessarily noisy because the moments are obtained via simulation. Taking the negative log of Equation (2.31) and replacing the true moments by noisy estimates $\hat{\boldsymbol{\mu}}_{\boldsymbol{s}}(\boldsymbol{\theta})$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{s}}(\boldsymbol{\theta})$, we obtain:

$$\log \Delta_{\boldsymbol{\theta}} = \frac{1}{2} \log |2\pi \hat{\boldsymbol{\Sigma}}_{\boldsymbol{s}}(\boldsymbol{\theta})| + \frac{1}{2} \left(\boldsymbol{s}_{\boldsymbol{\theta}_o} - \hat{\boldsymbol{\mu}}_{\boldsymbol{s}}(\boldsymbol{\theta}) \right)^{\intercal} \hat{\boldsymbol{\Sigma}}_{\boldsymbol{s}}^{-1}(\boldsymbol{\theta}) \left(\boldsymbol{s}_{\boldsymbol{\theta}_o} - \hat{\boldsymbol{\mu}}_{\boldsymbol{s}}(\boldsymbol{\theta}) \right). \tag{2.56}$$

As a result, the probabilistic model, namely the posterior predictive mean of a Gaussian process regression model, is able to emulate the synthetic likelihood after transformation:

$$f(\boldsymbol{s}_o \mid \boldsymbol{\theta}) \propto\!\!\!\approx \exp(-\mu_t(\boldsymbol{\theta})). \tag{2.57}$$

[10] In particular, any auxiliary model with a likelihood function.

Chapter 3

Research Questions

3.1 Previous work

In the previous chapter, we provided a holistic view of the Bayesian perspective to probabilistic inference and described the problem that arises in the presence of simulator-based models. We then presented the approximate Bayesian computation (ABC) approach to likelihood-free inference (LFI), pointing to the connection with inference in prescribed models (standard inference). By exploiting this relationship, we observed that likelihood-free inference corresponds to standard inference with an approximate likelihood function. This in turn allows us to take advantage of the recent advances that have been proposed for the latter. As a result, the challenges that seem to remain are that of accurately and efficiently approximating the implicit likelihood function, i.e. accurate and efficient conditional density estimation.

The first challenge is to obtain an accurate approximation given infinite computational power. Even in this idealized setting, poor summary statistics can yield a large approximation error. On the other hand, sufficient statistics yield zero approximation error. In general, the more informative are the summary statistics, the lower is the approximation error. Automatic discovery of informative summary statistics is therefore an important topic, but it falls beyond the scope of this work. In fact, in what follows, we assume that sufficient statistics are known.

The second challenge is conditional density estimation on a computational budget. At least two problems need to be solved in an efficient manner. One is that of identifying the regions of non-negligible density and the other of correctly estimating the region

of interest which roughly translates to the regions of highest density. By leveraging Bayesian optimization and stochastic acquisition rules, the Bayesian optimization for likelihood-free inference (BOLFI) framework attempts to solve these problems. Importantly, the framework relies on the following observations: the discrepancy is itself a random variable, and high density regions correspond to regions where the discrepancy tends to be small. In addition, the discrepancy is assumed to be conditionally normally distributed with a Gaussian process prior[1]. This in turn allows to cast the problem of density estimation as one of Gaussian process regression where the parameters of the simulator are the covariates and the discrepancy is the response variable. Thus far, BOLFI has only been applied to relatively low-dimensional problems.

3.2 High-dimensional Likelihood-free Inference

High-dimensional inference tends to be a difficult task[2]. In BOLFI, this task casts itself as one of high-dimensional Bayesian optimization, i.e. high-dimensional optimization and high-dimensional regression. The volume of the search space[3] increases exponentially and, as a result, identifying the regions of small discrepancies becomes exponentially more challenging. In this context, the acquisition function is high-dimensional and possibly non-convex, hence difficult to optimize. The number of acquisitions may also need to increase in order to correctly estimate the probabilistic relationship between the parameter vector and the discrepancy, which in turn can have computational repercussions. Learning the hyperparameters of the regression models becomes equally challenging. The resulting problem is thus both computational and statistical.

We attempt to solve the above problem by introducing additional assumptions into Gaussian process regression. For instance, previous work in high-dimensional Bayesian optimization assumed that the objective function has low effective dimensionality, only varying in a low-dimensional subspace [17, 19, 99]. However, this assumption is arguably too restrictive. A more flexible, but admittedly still strong assumption is to assume that the objective function has an unknown additive structure. Consequently, part of this work focuses on learning plausible decompositions, namely Gaussian process models of additive functions with non-overlapping groups.

[1] This is only assumed for mathematical convenience. In fact, it may be a strong assumption.
[2] The problem statement and proposed solution is roughly based on the proposal.
[3] In BOLFI, the search space corresponds to the parameter space.

3.3 Benchmarking

Likelihood-free inference is in general difficult, but recent methods have been able to push the field forward. Inference in implicit models is thus rapidly expanding towards high-dimensional problems [64, 65, 81]. Furthermore, this progress often relies on the empirical evaluation and comparison of different methods, and indeed several open-source software packages are able to perform likelihood-free inference [56], but it seems that there is a lack of benchmarking suites, particularly for high-dimensional problems.

Noteworthy low-dimensional models that require minimal background information are for instance the univariate g-and-k distribution [89], the Ricker model [36] and the Lotka-Volterra model [70], whose implementation can be found in [56]. On the other hand, due to the novelty factor of high-dimensional inference, standard high-dimensional models have not yet been established. Application-specific simulators exist, but require a certain amount of domain knowledge, e.g. [3]. For benchmarking purposes, the models should be able to emulate specific challenges that occur in real simulators, but otherwise be as general as possible, i.e. not necessarily tied to an application. Models based on multivariate distributions specified via marginals and a copula have been used in the context of high-dimensional inference [65], but the question whether they are able to emulate said challenges still remains. On a technical level, this challenge is important, but beyond the scope of this work.

Equally important is measuring the performance of different methods. The standard approach is to compare the approximation with the ground truth by visually inspecting a combination of univariate and bivariate marginal distributions. In low-dimensional problems it may be viable, but it does not scale well to high-dimensional settings. In particular, the amount of information that needs to be displayed increases substantially and it still may not be able to take into account the full dependence structure. Ideally, performance measures should be informative and intuitive, but not necessarily tied to a given method or model. Computational efficiency should also be considered in circumstances where performance needs to be continually monitored. Since this work focuses on extending BOLFI to high dimensions, we explore some measures that are tied to this approach, but others can in principle be applied to a wider range of methods.

Chapter 4

Methods

4.1 Additive Gaussian Processes

Gaussian process models of additive functions were first introduced in [23] and later explored in [48] for high-dimensional Bayesian optimization. In [48], the key assumption is that the objective function decomposes in an additive manner over disjoint groups. Formally, the authors assume that the n-dimensional vector of parameters $\boldsymbol{\theta}$ can be partitioned into G non-overlapping groups P_i where $|\cup_{i=1}^{G} P_i| = n$ and $A \cap B = \emptyset$, $\forall A, B \in \mathcal{M} = \{P_1, ..., P_G\} : A \neq B$. This in turn allows to write the objective function g as:

$$g(\boldsymbol{\theta}) = \sum_{i=1}^{G} g_i(\boldsymbol{\theta}_{P_i}), \tag{4.1}$$

where $\boldsymbol{\theta}_{P_i}$ is the component of $\boldsymbol{\theta}$ that belongs to partition P_i, i.e. $\boldsymbol{\theta}_{P_i} = \{\theta_j : j \in P_i\}$. If each g_i is drawn from a Gaussian process with mean function m_i and covariance function k_i, $g_i \sim \mathcal{GP}(m_i, k_i)$, then it can be shown that g is also drawn from a Gaussian process [48], $g \sim \mathcal{GP}(m, k)$, where

$$m(\boldsymbol{\theta}) = \sum_{i=1}^{G} m_i(\boldsymbol{\theta}_{P_i}), \qquad k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{i=1}^{G} k_i(\boldsymbol{\theta}_{P_i}, \boldsymbol{\theta}'_{P_i}). \tag{4.2}$$

Furthermore, the acquisition function to optimize is the lower confidence bound selection criterion (LCBSC)[1] [48], Equation (2.45). The first term involving the posterior predictive mean decomposes additively. However, due to the square root operation,

[1] Note that other acquisition functions can alternatively be considered.

the second term cannot be optimized separately. The authors then propose a simple modification to this function, which we refer to LCBSC Additive (LCBSCA):

$$\alpha(\boldsymbol{\theta} \mid \mathcal{E}^{(t)}, \mathcal{M}) = \sum_{i=1}^{G} \left(\mu_{i_t}(\boldsymbol{\theta}) - \sqrt{\eta_t^2 v_{i_t}(\boldsymbol{\theta})} \right) \tag{4.3}$$

$$= \sum_{i=1}^{G} \alpha_i(\boldsymbol{\theta}_{P_i} \mid \mathcal{E}^{(t)}, \mathcal{M}). \tag{4.4}$$

The modification to Algorithm 4 consists in finding the minimizer of each α_i separately so as to obtain $\boldsymbol{\theta}_{P_i}^{(t+1)}$. The point to query is $\boldsymbol{\theta}_i^{(t+1)} = \cup_{i=1}^{G} \boldsymbol{\theta}_{P_i}^{(t+1)}$. This is shown in Algorithm 5.

Algorithm 5 BO-LCBSCA with Additive Gaussian Process model

Require: Additive model $\mathcal{M}$, evidence set: $\mathcal{E}^{(t_0)} = \left\{ \left(\boldsymbol{\theta}^{(j)}, g^{(j)} \right) \right\}_{j=1}^{t_0}$

for $t = t_0$ **to** ... **do**

for $i = 1$ **to** G **do**

$\boldsymbol{\theta}_{P_i}^{(t+1)} = \arg\min_{\boldsymbol{\theta}_{P_i}} \alpha_i(\boldsymbol{\theta}_{P_i} \mid \mathcal{E}^{(t)}, \mathcal{M})$

$\boldsymbol{\theta}_i^{(t+1)} = \cup_{i=1}^{G} \boldsymbol{\theta}_{P_i}^{(t+1)}$

$g^{(t+1)} = g(\boldsymbol{\theta}^{(t+1)})$

$\mathcal{E}^{(t+1)} = \mathcal{E}^{(t)} \cup \left\{ \left(\boldsymbol{\theta}^{(t+1)}, g^{(t+1)} \right) \right\}$

A question that arises is whether the formula for η_t^2 should remain the same. The authors derive a formula based on bounding the regret[2], but choose not to use it, claiming it may be too conservative. Since our primary aim is not to evaluate different acquisition functions, we choose to keep it as defined in Equation (2.46)[3], based on previous work from a fellow student [24].

More importantly, Algorithm 5 assumes that the additive model $\mathcal{M}$ is known, which admittedly is not realistic. In [48], the authors adopt a bag of models approach, where at regular intervals they generate multiple random additive structures and keep the best performing model. A uniform prior distribution over the models is assumed, but the hyperparameters need to be integrated out. The formula for model comparison given

[2]Note that the regret bounds are often derived assuming a zero mean function, which does not apply in our case.

[3]In the next chapter, we show that this can lead to problems. In particular, for models where $|P_i| = 1$ and n is large. We later show that $\eta_t^2 = 0.2d\log(2t)$ [48], where $d = \max(|P_1|, ..., |P_G|)$ leads to more stable behavior.

by Equation (2.7) becomes[4]:

$$\pi(\mathcal{M}|\mathcal{E}^{(t)}) \propto \int f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \mathcal{H}, \mathcal{M})\pi(\mathcal{H} \mid \mathcal{M})\, d\mathcal{H}, \tag{4.5}$$

where $\mathcal{H} = \{\boldsymbol{\eta}_i\}_{i=1}^{G}$, i.e. the set of hyperparameters $\mathcal{H}$ includes the lengthscales and signal variances of squared exponential kernels, and the coefficients of the fully additive mean function, defined in Equation (2.34).

The expression given by Equation (4.5) cannot be computed exactly and instead of approximating the integral, the authors adopt a maximum likelihood approach by assuming a uniform prior on the hyperparameters. Hence, the hyperparameters of each randomly sampled additive model $\mathcal{M}$ are learned by maximizing the corresponding (log) marginal likelihood [76]:

$$\log f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \mathcal{M}; \mathcal{H}) = -\frac{1}{2}\boldsymbol{g}_t^{\intercal}\left(\boldsymbol{K}_t + \sigma_n^2\boldsymbol{I}\right)^{-1}\boldsymbol{g}_t - \frac{1}{2}\log\left|\boldsymbol{K}_t + \sigma_n^2\boldsymbol{I}\right| - \frac{t}{2}\log(2\pi), \tag{4.6}$$

$$\mathcal{H}_{\mathcal{M}}^{*} = \arg\max_{\mathcal{H}} \log f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \mathcal{M}; \mathcal{H}). \tag{4.7}$$

Note that more generally, Equation (4.7) includes regularization terms (penalized maximum likelihood), often given by the prior on the hyperparameters[5], in which case it is also known as maximum a posteriori, i.e.

$$\mathcal{H}_{\mathcal{M}}^{*} = \arg\max_{\mathcal{H}} \log f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \mathcal{M}; \mathcal{H}) + \log\pi(\mathcal{H} \mid \mathcal{M}). \tag{4.8}$$

After hyperparameter optimization, the best performing model is the model with the largest marginal likelihood.

In [48], the bag of additive models approach is shown to outperform other high-dimensional Bayesian optimization methods, namely those that rely on the assumption of low effective dimensionality. The motivation behind the random generation of partitions is that evaluating all possible partitions in a high-dimensional setting is intractable, since this number grows super-exponentially with the dimensionality of the parameter vector $\boldsymbol{\theta}$ [28, 84]. However, recent work shows that even this approach is sub-optimal and that learning the additive structure via Markov chain Monte Carlo (MCMC) yields better results [28, 98]. Two competing strategies exist: one based on Metropolis-Hastings (MH) and the other based on Gibbs sampling. Thus far, these strategies have not been compared.

[4] Recall the notation introduced in Section 2.5.

[5] A common assumption is that the prior fully factorizes.

4.1.1 Metropolis-Hastings for Structure Discovery

In [28], the authors sample from the posterior distribution over possible additive models by defining a transition kernel $p(\mathcal{M}'|\mathcal{M})$ with the following proposal mechanism:

1. If possible, randomly choose between *split* or *merge* with equal probability.

   (a) *split*: Choose group $P_i \in \mathcal{M}$ uniformly at random from set of non-singleton groups. Split it in two by Bernoulli sampling, i.e.
   $P_i^{(0)} = \{j : j \in P_i \wedge \mathbb{1}[z_j = 0]\}$, where $z_j \sim Bern(0.5)$,
   $P_i^{(1)} = P_i \setminus P_i^{(0)}$.
   The proposed structure is $\mathcal{M}' = (\mathcal{M} \setminus \{P_i\}) \cup \{P_i^{(0)}, P_i^{(1)}\}$.

   (b) *merge*: Randomly choose two partitions $P_i, P_j \in \mathcal{M}$ without replacement and merge them, $P^+ = P_i \cup P_j$.
   The proposed structure is $\mathcal{M}' = \big(\mathcal{M} \setminus \{P_i, P_j\}\big) \cup \{P^+\}$.

Assuming for simplicity that, for each model, the hyperparameters are optimized via penalized maximum likelihood, and that the prior distribution over additive models is uniform, the posterior $\pi(\mathcal{M} \mid \mathcal{E}^{(t)})$ defined in Equation (4.5) can be written as:

$$\pi(\mathcal{M} \mid \mathcal{E}^{(t)}) \propto f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \mathcal{M}; \mathcal{H}^*_{\mathcal{M}}). \tag{4.9}$$

In order to satisfy the detailed balance equation that targets the posterior distribution $\pi(\mathcal{M} \mid \mathcal{E}^{(t)})$ as the stationary distribution, the acceptance probability must be given by:

$$p_\alpha = \min\left(1, \frac{f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \mathcal{M}'; \mathcal{H}^*_{\mathcal{M}'}) p(\mathcal{M} \mid \mathcal{M}')}{f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \mathcal{M}; \mathcal{H}^*_{\mathcal{M}}) p(\mathcal{M}' \mid \mathcal{M})}\right). \tag{4.10}$$

Finally, it is important to note that in this case the aim is not to estimate the posterior $\pi(\mathcal{M} \mid \mathcal{E}^{(t)})$. The MH sampler is used as a randomized search algorithm that allows to generate plausible additive models. In fact, at any given time t, we only keep one model active. In [28], the authors also explore the possibility of multiple simultaneous models, but due to time and computational constraints we do not follow such approach.

4.1.2 Gibbs for Structure Discovery

An alternative to the previous strategy is Gibbs sampling. In [98], the authors consider a conjugate hierarchical model such that each dimension j is assigned to one of G groups by drawing an assignment variable z_j, such that $z_j \sim \text{Cat}(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ is in turn

a G-dimensional probability vector drawn from a Dirichlet distribution, $\boldsymbol{\lambda} \sim \text{Dir}(\boldsymbol{\beta})$, and corresponds to mixing proportions. The joint posterior distribution of additive partitions and mixing proportions can be written as:

$$\pi(\boldsymbol{z}, \boldsymbol{\lambda} \mid \mathcal{E}^{(t)}; \boldsymbol{\beta}) \propto f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \boldsymbol{z}) \pi(\boldsymbol{z} \mid \boldsymbol{\lambda}) \pi(\boldsymbol{\lambda}; \boldsymbol{\beta}). \tag{4.11}$$

The interest lies in the (marginal) posterior distribution of additive partitions, which can be obtained via marginalization[6]:

$$\pi(\boldsymbol{z} \mid \mathcal{E}^{(t)}; \boldsymbol{\beta}) \propto f(\boldsymbol{g}_t | \boldsymbol{\Phi}_t, \boldsymbol{z}) \int \pi(\boldsymbol{z} \mid \boldsymbol{\lambda}) \pi(\boldsymbol{\lambda}; \boldsymbol{\beta}) d\boldsymbol{\lambda} \tag{4.12}$$

$$\propto f(\boldsymbol{g}_t | \boldsymbol{\Phi}_t, \boldsymbol{z}) \frac{\Gamma(\sum_i \beta_i)}{\Gamma(n + \sum_i \beta_i)} \prod_{i=1}^{G} \frac{\Gamma(|P_i| + \beta_i)}{\Gamma(\beta_i)} \tag{4.13}$$

$$\propto f(\boldsymbol{g}_t | \boldsymbol{\Phi}_t, \boldsymbol{z}) \, \pi(\boldsymbol{z}; \boldsymbol{\beta}), \tag{4.14}$$

where $P_i = \{j : z_j = i\}$ and $n = |\boldsymbol{z}| = |\boldsymbol{\theta}|$. In the context of Gibbs sampling, it is necessary to derive the complete conditional distribution [98]:

$$\pi(z_j = h \mid \boldsymbol{z}_{-j}, \mathcal{E}^{(t)}; \boldsymbol{\beta}) \propto f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \boldsymbol{z}) \pi(z_j = h \mid \boldsymbol{z}_{-j}; \boldsymbol{\beta}) \tag{4.15}$$

$$\propto f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \boldsymbol{z}) \, (|P_h| + \beta_h) \tag{4.16}$$

$$\propto e^{\phi_j^{(h)}}, \tag{4.17}$$

where

$$\phi_j^{(h)} = -\frac{1}{2} \boldsymbol{g}_t^{\intercal} (\mathbf{K}_t^{(z_j = h)} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{g}_t - \frac{1}{2} \log |\mathbf{K}_t^{(z_j = h)} + \sigma_n^2 \boldsymbol{I}| + \log(|P_h| + \beta_h), \tag{4.18}$$

and $\mathbf{K}_t^{(z_j = h)}$ denotes the covariance matrix $\mathbf{K}_t$ given the proposed additive model $\mathcal{M}'$ in which the jth dimension belongs to group P_h. More precisely, letting the current additive model be $\mathcal{M}$ and $P_r \in \mathcal{M}$ the current group containing the jth dimension, the proposed additive model is:

$$\mathcal{M}' = (\mathcal{M} \setminus \{P_r, P_h\}) \cup \{P_r^-, P_h^+\}, \tag{4.19}$$

$$P_r^- = P_r \setminus \{j\} \tag{4.20}$$

$$P_h^+ = P_h \cup \{j\}. \tag{4.21}$$

Furthermore, note that in order to obtain a sample from the complete conditional distribution, the Gumbel trick can be applied. In particular, it consists in sampling G

[6]The integral can be evaluated in closed form because the Dirichlet distribution is conjugate to the categorical distribution.

independent standard Gumbel variables $\omega^{(i)} \sim Gumbel(0,1)$ and set the assignment according to $z_j = \arg\max_i \phi_j^{(i)} + \omega^{(i)}$.

The authors propose a sequential-scan Gibbs sampler (Algorithm 2). However, sampling an assignment for each dimension requires performing $G = n$ hyperparameter optimizations (one for each model)[7], where $n = |\boldsymbol{\theta}|$. In turn, each hyperparameter optimization and marginal likelihood evaluation cost $O(t^3)$ due to matrix inversion. Since each dimension is updated once, the total cost to obtain one sample from sequential-scan Gibbs is $O(t^3 n^2)$, while the MH sampler has a cost of $O(t^3)$. In high-dimensional Bayesian optimization, this Gibbs sampler has a prohibitive cost[8]. For this reason, a different sampling strategy is adopted. One possibility is to keep the sequential order, but only allow one dimension update at a time. Instead, we adopt a random-scan strategy as shown in Algorithm 6. Importantly, both samplers are controlled based on the number of evaluations to the marginal likelihood, allowing for a fairer comparison. The condition is not however explicitly enforced in Gibbs sampling, i.e. the algorithm is not stopped while sampling an assignment.

4.2 Performance Measures

In this work, the focus lies on the assessment of different additive Gaussian process models for BOLFI. In particular, one aim is to determine to what extent structure discovery via MCMC can improve the statistical efficiency of the BOLFI nonparametric estimator, as defined in Equation (2.52). It would be possible to compare these approaches with typical ABC samplers, but the comparison would not necessarily be fair. The reason is twofold: the methods rely on random sampling, as opposed to active sampling; samples from the posterior distribution are obtained without the need to learn a surrogate model of the approximate likelihood. Consequently, ABC samplers require more simulations to guarantee an accurate approximation of the posterior [36].

Thus, some of the performance measures we explore can only be applied to BOLFI,

[7] This is the limit case. In practice, it is possible to finetune the algorithm by observing that the marginal likelihood resulting from an assignment to an empty group yields the same value, regardless of the label of such partition. Therefore, it only needs to be computed once. In addition, optimizing the initial additive model before Gibbs may allow minor computational savings (not shown in Algorithm 6), e.g. when the model does not change.

[8] For increased tractability, the authors assume that G is known and is smaller than n. We do not make such assumption in this work, hence $G = n$.

Algorithm 6 Random-scan Gibbs sampling for structure discovery

Require: Initial additive model: $\mathcal{M}$, evidence set: $\mathcal{E}^{(t)}$, number of marginal likelihood evaluations: N

$c \leftarrow 0$
while $c < N$ **do**
  $j \sim U\{1,n\}$ ▷ Randomly choose a dimension
  $skipOpt \leftarrow False$ ▷ Avoid unnecessary computations
  for $i = 1$ **to** n **do**
    if $|P_i| == 0$ **then**
      if $skipOpt$ **then**
        $\phi_j^{(i)} \leftarrow \phi_j^{(q)}$
        $\mathcal{M}_i' \leftarrow \mathcal{M}_q'$
        $continue$
      else
        $q \leftarrow i$
        $skipOpt \leftarrow True$
    Set $\mathcal{M}_i'$ according to Equation (4.19)
    Determine $\mathcal{H}^*_{\mathcal{M}_i'}$ by penalized max likelihood, as given in Equation (4.8)
    Compute $\phi_j^{(i)}$ according to Equation (4.18)
    $c \leftarrow c+1$
  $\omega^{(i)} \sim Gumbel(0,1),\ i = 1,...,n$
  $z_j = \arg\max_i \phi_j^{(i)} + \omega^{(i)}$
  $\mathcal{M} \leftarrow \mathcal{M}_{z_j}'$

but others, namely those that assess the quality of the posterior approximation, are more generally applicable.

4.2.1 Bayesian Optimization and Likelihood Approximation

The first set of performance measures provide information regarding the Bayesian optimization process and the likelihood approximation. The ground truth is assumed to be known.

An important concept in Bayesian optimization is instantaneous regret [88]. The definition can vary to a slight degree depending on the application, e.g. whether the objec-

tive function is deterministic or stochastic. We define the instantaneous regret as

$$\text{Regret} = g^\star(\boldsymbol{\theta}^*) - g^\star(\boldsymbol{\theta}_o), \tag{4.22}$$

where

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mu_t(\boldsymbol{\theta}), \tag{4.23}$$

and $g^\star$ is the true objective function (deterministic), $\boldsymbol{\theta}_o$ is the global minimum of $g^\star$, $\mu_t(\boldsymbol{\theta})$ is the posterior predictive mean of a Gaussian process regression model at time t, defined according to Equation (2.44). Instantaneous regret measures the output error, but in this work we are also interested in the input error (location error):

$$\text{LocError} = ||\boldsymbol{\theta}^* - \boldsymbol{\theta}_o||_\infty, \tag{4.24}$$

where $||\cdot||_\infty$ denotes the infinity norm. Unlike the Euclidean norm, the infinity norm is insensitive to the dimensionality of the parameter vector. Another useful diagnostic[9], consists in assessing whether $\boldsymbol{\theta}^*$ is near the boundary of the search space:

$$\text{NearB} = \mathbb{1}\left[z_\infty(\boldsymbol{u} - \boldsymbol{\theta}^*) \leq \gamma\right] \vee \mathbb{1}\left[z_\infty(\boldsymbol{\theta}^* - \boldsymbol{l}) \leq \gamma\right], \tag{4.25}$$

where for a generic vector $\boldsymbol{x} = (x_1, ..., x_r)$,

$$z_\infty(\boldsymbol{x}) = \min(|x_1|, ..., |x_r|), \tag{4.26}$$

and $\boldsymbol{u}$, $\boldsymbol{l}$ denote the upper and lower bounds of the search space (hypercube). The value of the adjustable parameter γ should be small, possibly defined based on the search space. In our experiments, we set $\gamma = 0.1$.

The performance measures specified so far only provide information about a single location, namely whether the expected minimum $\boldsymbol{\theta}^*$ is close to the global minimum $\boldsymbol{\theta}_o$. In the context of LFI, it may be important to determine if μ_t is able to approximate sufficiently well the true, but generally unknown discrepancy. If we assume that $g^\star$ is the true discrepancy function[10], then a potentially informative measure is to compare the differences between μ_t and $g^\star$ at multiple test locations, specifically locations around $\boldsymbol{\theta}_o$, where the corresponding true discrepancies are small[11]. For now, assuming that

[9] In high-dimensional Bayesian optimization, a relatively unknown challenge is the boundary issue [93], where the algorithm tends to acquire data near the boundaries. This behavior is discussed in more depth in the next chapter.

[10] This assumption will become clear in the next chapter.

[11] Recall that in BOLFI the objective is to approximate the high density regions of the likelihood, which is equivalent to regions of small discrepancy.

these locations are given by $\{\boldsymbol{\theta}_{\pi}^{(r)}\}_{r=1}^{R}$, we can indirectly measure the quality of the likelihood approximation according to the discrepancy error:

$$\text{DiscError} = \frac{1}{R}\sum_{r=1}^{R} ||\mu_t(\boldsymbol{\theta}_{\pi}^{(r)}) - g^{\star}(\boldsymbol{\theta}_{\pi}^{(r)})||_1, \tag{4.27}$$

where $||\cdot||_1$ is the ℓ_1 norm.

The discrepancy error crucially relies on an informative set of test locations, which in turn implies that the sampling process should take into account the local geometry of the true likelihood function. Hence, locations can be sampled from a posterior whose prior distribution $\pi(\boldsymbol{\theta})$ is uniform. Since the resulting posterior is high-dimensional, sampling will in general be difficult. If the true likelihood function is differentiable, then it is possible to use the more efficient Hamiltonian Monte Carlo (HMC) samplers, but sampling can admittedly still be time-consuming.

Assuming differentiability, optimization-based inference methods can also be used. For instance, recalling the discussion in Section 2.3.2, the Laplace approximation can be exploited to obtain a normal distribution approximation around $\boldsymbol{\theta}_o$ for the posterior distribution, proportional to the true likelihood function. Letting $\boldsymbol{s}_o \sim f(\cdot \mid \boldsymbol{\theta}_o)$ be the observed summary statistics,

$$f(\boldsymbol{s}_o \mid \boldsymbol{\theta}) \propto \pi(\boldsymbol{\theta} \mid \boldsymbol{s}_o) \overset{\propto}{\sim} \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_o, \boldsymbol{\Sigma}_o), \tag{4.28}$$

where $\boldsymbol{\Sigma}_o = \boldsymbol{H}_o(\boldsymbol{\theta}_o)^{-1}$ and $\boldsymbol{H}_o(\boldsymbol{\theta}_o)$ is the Hessian of the negative log of the true likelihood function evaluated at $\boldsymbol{\theta}_o$. Similarly, if the BOLFI nonparametric likelihood approximation[12] is approximated by a normal distribution centered around $\boldsymbol{\theta}^*$,

$$f_{nBOLFI}(\boldsymbol{s}_o \mid \boldsymbol{\theta}) = F_{\mathcal{N}}\left(\frac{\varepsilon - \mu_t(\boldsymbol{\theta})}{\sqrt{v_t(\boldsymbol{\theta}) + \sigma_n^2}}\right) \propto \pi_{nBOLFI}(\boldsymbol{\theta} \mid \boldsymbol{s}_o) \overset{\propto}{\sim} \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}^*, \boldsymbol{\Sigma}_*), \tag{4.29}$$

where $\boldsymbol{\Sigma}_* = \boldsymbol{H}_{nBOLFI}(\boldsymbol{\theta}^*)^{-1}$, it is possible to obtain an estimate of the quality of the approximation, in closed-form, according to the Kullback-Leibler divergence for multivariate normal distributions [22]:

$$\text{KL}(\pi || \pi_{nBOLFI}) = \frac{1}{2}\left(\text{tr}(\boldsymbol{\Sigma}_*^{-1}\boldsymbol{\Sigma}_o) + (\boldsymbol{\theta}^* - \boldsymbol{\theta}_o)^{\intercal}\boldsymbol{\Sigma}_*^{-1}(\boldsymbol{\theta}^* - \boldsymbol{\theta}_o) - n + \log\frac{|\boldsymbol{\Sigma}_*|}{|\boldsymbol{\Sigma}_o|}\right), \tag{4.30}$$

where tr corresponds to the trace and $n = |\boldsymbol{\theta}^*| = |\boldsymbol{\theta}_o|$. Note that a closed-form expression is available for $\boldsymbol{H}_{nBOLFI}(\boldsymbol{\theta}^*)$, since f_{nBOLFI} is differentiable[13]. Furthermore, it

[12] Previously defined in Equation (2.53).

[13] Refer to Appendix A.

should be emphasized that while we use the Laplace approximation in this work, the approach can be extended. For instance, the normal distribution can instead be fitted using variational inference or, for increased flexibility, a mixture of normal distributions can be adopted. The exact KL divergence cannot be computed in closed-form for the latter, but computationally efficient approximations exist [40]. Other information-theoretic measures may however yield closed-form expressions, e.g. Jensen-Rényi divergence [96].

4.2.2 Additive Gaussian Processes

A particular aim of this work is to determine whether correctly learning the underlying additive structure leads to better approximations. The ground truth is again assumed to be known.

Due to possible label switching, we define the performance measures over dimension pairs. In particular, let $\mathcal{M}^a$ be the ground truth (oracle) additive model[14] and $\mathcal{M}^b$ the additive model that has been learned. We define the number of true positives (TP) and the true positive rate (TPR) as

$$\text{TP} = \sum_{i<j\leq n} \mathbb{1}[a_i = a_j \wedge b_i = b_j], \qquad \text{TPR} = \frac{TP}{\sum_{i<j\leq n} \mathbb{1}[a_i = a_j]}, \tag{4.31}$$

where n is the number of dimensions, a_i and b_i correspond to the assignment of the ith dimension to group $P^a_{a_i} \in \mathcal{M}^a$ and $P^b_{b_i} \in \mathcal{M}^b$ respectively. Similarly, the number of true negatives (TN) and the corresponding rate (TNR) are given by

$$\text{TN} = \sum_{i<j\leq n} \mathbb{1}[a_i \neq a_j \wedge b^g_i \neq b^g_j], \qquad \text{TNR} = \frac{TN}{\sum_{i<j\leq n} \mathbb{1}[a_i \neq a_j]}. \tag{4.32}$$

The accuracy over pairs is[15]

$$\text{Accuracy} = \frac{TP+TN}{\binom{n}{2}}, \tag{4.33}$$

where $\binom{n}{2}$ is the total number of pairs. Furthermore, let us define the precision as

$$\text{Precision} = \frac{TP}{\sum_{i<j\leq n} \mathbb{1}[b_i = b_j]}. \tag{4.34}$$

[14] Recall that an additive model, or partition, is defined as a set of non-overlapping groups.
[15] Also known as Rand index.

The F-score is thus given by

$$\text{F-score} = 2\,\frac{\text{Precision}\cdot\text{TPR}}{\text{Precision}+\text{TPR}}. \tag{4.35}$$

Finally, in some cases, we also evaluate a relative goodness of fit measure given by the log (marginal) likelihood ratio[16]:

$$\text{LLR} = \log\frac{f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \mathcal{M}^a; \mathcal{H}^*_{\mathcal{M}^a})}{f(\boldsymbol{g}_t \mid \boldsymbol{\Phi}_t, \mathcal{M}^b; \mathcal{H}^*_{\mathcal{M}^b})}, \tag{4.36}$$

where $\boldsymbol{g}_t$ and $\boldsymbol{\Phi}_t$ are the observed function values and respective locations (evidence set) acquired by model $\mathcal{M}^b$. Large values indicate that the learned model $\mathcal{M}^b$ fits the acquired data poorly.

4.2.3 Posterior Approximation

The posterior distribution is of central importance in any Bayesian analysis. Hence, a careful choice of informative performance measures is necessary. In a previous chapter, it was mentioned that the assessment often relies on visual inspection of marginal distributions, but a particular problem is scalability. In this section, we define several measures that can in principle be applied to a wide range of methods. The trade-off is that these performance measures tend to be computationally expensive. In particular, all measures rely on posterior samples that have been obtained for instance from a MCMC algorithm.

Let us begin by assuming that we have two sets of posterior samples, $\{\boldsymbol{\theta}_a^{(i)}\}_{i=1}^N$ and $\{\boldsymbol{\theta}_b^{(i)}\}_{i=1}^N$. The first is drawn from the true posterior $\pi_a(\boldsymbol{\theta} \mid \boldsymbol{s}_o)$ and the other from an approximation $\pi_b(\boldsymbol{\theta} \mid \boldsymbol{s}_o)$. The first performance measure consists in performing a two-sample test to assess the similarity between the two sets. Since the samples form empirical distributions of the posteriors, for an effective sample size sufficiently large, we are able to infer whether $\pi_b(\boldsymbol{\theta} \mid \boldsymbol{s}_o)$ is able to provide a good approximation to $\pi_a(\boldsymbol{\theta} \mid \boldsymbol{s}_o)$. A popular test is based on kernel methods [35]. The Maximum Mean Discrepancy (MMD) is defined as

$$\text{MMD}[\mathcal{F}, \pi_a, \pi_b] = \sup_{f\in\mathcal{F}} \left(\mathbb{E}_{\pi_a}[f(\boldsymbol{\theta})] - \mathbb{E}_{\pi_b}[f(\boldsymbol{\theta})]\right), \tag{4.37}$$

[16] For practical reasons, the hyperparameters are optimized (not random variables): $\mathcal{H}^*_{\mathcal{M}^a}$ and $\mathcal{H}^*_{\mathcal{M}^b}$. As a result, the likelihood ratio is equivalent to Bayes factor with a uniform prior on models.

where $\mathcal{F}$ is the unit ball in a reproducing kernel Hilbert space. An unbiased empirical estimate can be shown to be [35]:

$$\begin{aligned}\mathrm{MMD}^2 =& \frac{1}{N(N-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{N} k(\boldsymbol{\theta}_a^{(i)}, \boldsymbol{\theta}_a^{(j)}) - \frac{2}{N^2} \sum_{i,j=1}^{N} k(\boldsymbol{\theta}_a^{(i)}, \boldsymbol{\theta}_b^{(j)}) \\ &+ \frac{1}{N(N-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{N} k(\boldsymbol{\theta}_b^{(i)}, \boldsymbol{\theta}_b^{(j)}).\end{aligned} \tag{4.38}$$

The kernel k is typically the squared exponential kernel with a single shared length-scale λ and $\sigma_f^2 = 1$, previously defined in Equation (2.35). In our experiments, λ is determined as the median of the Euclidean distances between all pairs in $\{\boldsymbol{\theta}_a^{(i)}\}_{i=1}^N$, as suggested in [46].

The other performance measures are inspired by posterior predictive checks [29]. In standard Bayesian analysis, posterior predictive checks are used to assess different models. The intuition is that a good model, given by a posterior distribution and a data generating process, is able to generate data that resembles the observed data[17]. Posterior predictive checks require the specification of test statistics and, in this context, it is important to choose test statistics that are different from the summary statistics used to determine the parameters of a given model [27]. The reason is that posterior predictive checks use the data twice, once for model fitting and another for model criticism. However, we argue that posterior predictive checks can equally be used for the assessment of competing inference methods. Given the same summary statistics, different inference methods yield different posterior distributions only due to the inherent approximate nature of the inference process. And in such case it no longer seems justifiable to use a set of test statistics that is different from the summary statistics, since the goal is indeed to criticize the fit, or lack thereof. Note that both model mismatch and approximate inference introduce errors during the learning process and, for that reason, it is important not to confound the source, when comparing inference methods. Ideally, inference methods should be tested in situations in which model mismatch does not occur. This in turn means that real data analyses should be avoided, since the observed

[17]Notice the similarities with rejection ABC (Algorithm 3). In fact, the latter can be interpreted as a predictive check, where the models under comparison are specified only by candidate posterior distributions, since the data generating process is known (simulator). Let $\mathcal{S}$ denote the set of all samples generated by the (fixed) prior distribution. The set of models is thus the set of empirical distributions given by the powerset of $\mathcal{S}$. A good model, or equivalently a good candidate posterior distribution, is the empirical distribution with the largest number of samples such that each sample generated a dataset whose discrepancy is less or equal than a certain threshold.

data should be generated according to a given model. In these circumstances, posterior predictive checks can indeed be used as a tool to criticize approximate inference.

In standard Bayesian analysis, the posterior predictive checks require the generation of datasets from the posterior predictive distribution:

$$\pi(\boldsymbol{y} \mid \boldsymbol{y}_o, \mathcal{A}) = \int f(\boldsymbol{y} \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta} \mid \boldsymbol{y}_o, \mathcal{A}) d\boldsymbol{\theta}, \tag{4.39}$$

where $\boldsymbol{y}_o$ is the observed dataset[18] and $\mathcal{A}$ is an arbitrary inference method that has been used to estimate the posterior distribution. Additionally, if sufficient summary statistics are known, Equation (4.39) is equivalent to:

$$\pi(\boldsymbol{y} \mid \boldsymbol{s}_o, \mathcal{A}) = \int f(\boldsymbol{y} \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta} \mid \boldsymbol{s}_o, \mathcal{A}) d\boldsymbol{\theta}, \tag{4.40}$$

where $\boldsymbol{s}_o$ are the observed statistics. The posterior predictive distribution of test statistics (same as summary statistics) can also be written as[19]:

$$\pi(\boldsymbol{s} \mid \boldsymbol{s}_o, \mathcal{A}) = \int f(\boldsymbol{s} \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta} \mid \boldsymbol{s}_o, \mathcal{A}) d\boldsymbol{\theta}. \tag{4.41}$$

Posterior predictive checks occur at this level. In high-dimensional problems, the dimensionality of the vectors of summary statistics tends to be high, and so it seems there is little to gain in terms of information reduction from this analysis. A possibility is to replace the generation process by the corresponding likelihood function, yielding a posterior predictive likelihood:

$$f(\boldsymbol{s}_o \mid \boldsymbol{s}_o, \mathcal{A}) = \int f(\boldsymbol{s}_o \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta} \mid \boldsymbol{s}_o, \mathcal{A}) d\boldsymbol{\theta}. \tag{4.42}$$

Under sufficiency, this quantity measures the plausibility of samples from the posterior distribution estimated by $\mathcal{A}$ to generate data equal to the observed or, alternatively, it can simply be interpreted as the posterior mean of the likelihood function [1]. Importantly, it can be used to compare posterior approximations of different inference methods, but a potential problem is that the maximum occurs when the posterior approximation is:

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{s}_o, \mathcal{A}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_o), \tag{4.43}$$

where δ is the Dirac delta function. Consequently, the value should be compared to that of a reference, obtained either via a ground truth posterior or an approximate

[18] The observed dataset is assumed to be generated according to $\boldsymbol{y}_o \sim f(\cdot \mid \boldsymbol{\theta}_o)$.

[19] Note that we are overloading notation.

posterior distribution that is known to be accurate (e.g. empirical distribution with a large number of samples acquired by a well-calibrated MCMC algorithm).

On the other hand, in likelihood-free inference, the data generating process is given by the simulator, but the corresponding likelihood function cannot be evaluated explicitly. If however the likelihood function is known (toy problem) or it is known that a given auxiliary model with a likelihood function can provide a reasonable approximation to the implicit likelihood, then a similar approach to the one described can be applied. For instance, assuming that sufficient statistics are known, the Monte Carlo estimator of the ABC likelihood, previously defined in Equation (2.32), converges to the true likelihood as $M \to \infty$ and $\varepsilon \to 0$. In practice, however we can simply use noisy estimates. Assuming that the posterior distribution is approximated by an empirical distribution of N samples, we can write the approximate posterior predictive likelihood (omitting $\mathcal{A}$):

$$f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{s}_o) = \int f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta} \mid \boldsymbol{s}_o)d\boldsymbol{\theta} \tag{4.44}$$

$$\approx \int f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{\theta})\frac{1}{N}\sum_{i=1}^{N}\delta(\boldsymbol{\theta}-\boldsymbol{\theta}^{(i)})d\boldsymbol{\theta}, \qquad \boldsymbol{\theta}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s}_o) \tag{4.45}$$

$$= \frac{1}{N}\sum_{i=1}^{N} f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{\theta}^{(i)}), \qquad \boldsymbol{\theta}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s}_o) \tag{4.46}$$

$$\approx \frac{1}{N}\frac{1}{M}\sum_{i=1}^{N}\sum_{j=1}^{M} K_\varepsilon\left(d(\boldsymbol{s}^{(i,j)}, \boldsymbol{s}_o)\right), \; \boldsymbol{s}^{(i,j)} \sim f(\cdot \mid \boldsymbol{\theta}^{(i)}), \boldsymbol{\theta}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s}_o). \tag{4.47}$$

If we let $M = 1$, we then obtain:

$$f_{ABC}(\boldsymbol{s}_o \mid \boldsymbol{s}_o) \approx \frac{1}{N}\sum_{i=1}^{N} K_\varepsilon\left(d(\boldsymbol{s}^{(i)}, \boldsymbol{s}_o)\right), \; \boldsymbol{s}^{(i)} \sim f(\cdot \mid \boldsymbol{\theta}^{(i)}), \boldsymbol{\theta}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s}_o). \tag{4.48}$$

This measure can potentially be used in the assessment of likelihood-free inference methods. Alternatively, since discrepancies and the likelihood are related, we can approximate the posterior predictive distribution of discrepancies:

$$\pi(\Delta \mid \boldsymbol{s}_o) = \int f(\Delta \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta} \mid \boldsymbol{s}_o)d\boldsymbol{\theta}. \tag{4.49}$$

Unlike summary statistics, the discrepancy function is by convention a scalar-valued function. Hence, the respective posterior predictive distribution is univariate, regardless of the dimensionality of $\boldsymbol{\theta}$. Importantly, performance measures based on this distribution can, to a certain extent, be interpreted without the need to compare with a ground truth. For instance, if the empirical mean of the posterior predictive is large, the approximation provided by a given method is likely to be poor.

Given a set of posterior samples $\mathcal{V} = \{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N}$, the set of discrepancies that form the corresponding empirical posterior predictive distribution can be obtained by simulating a discrepancy for each posterior sample, i.e. $\{\Delta_{\boldsymbol{\theta}} : \Delta_{\boldsymbol{\theta}} \sim f(\cdot \mid \boldsymbol{\theta}), \forall \boldsymbol{\theta} \in \mathcal{V}\}$[20]. Indeed, following a similar derivation as before, we can write:

$$\pi(\Delta \mid \boldsymbol{s}_o) = \int f(\Delta \mid \boldsymbol{\theta}) \pi(\boldsymbol{\theta} \mid \boldsymbol{s}_o) d\boldsymbol{\theta} \tag{4.50}$$

$$\approx \int f(\Delta \mid \boldsymbol{\theta}) \frac{1}{N} \sum_{i=1}^{N} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}) d\boldsymbol{\theta}, \qquad \boldsymbol{\theta}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s}_o) \tag{4.51}$$

$$= \frac{1}{N} \sum_{i=1}^{N} f(\Delta \mid \boldsymbol{\theta}^{(i)}), \qquad \boldsymbol{\theta}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s}_o) \tag{4.52}$$

$$\approx \frac{1}{N} \frac{1}{M} \sum_{i=1}^{N} \sum_{j=1}^{M} \delta(\Delta - \Delta^{(i,j)}), \qquad \Delta^{(i,j)} \sim f(\cdot \mid \boldsymbol{\theta}^{(i)}), \quad \boldsymbol{\theta}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s}_o). \tag{4.53}$$

Again, with $M = 1$, we obtain:

$$\pi(\Delta \mid \boldsymbol{s}_o) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(\Delta - \Delta^{(i)}), \qquad \Delta^{(i)} \sim f(\cdot \mid \boldsymbol{\theta}^{(i)}),\ \boldsymbol{\theta}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s}_o), \tag{4.54}$$

and the corresponding kernel density estimate is given by:

$$\pi^{kde}(\Delta \mid \boldsymbol{s}_o) = \frac{1}{N} \sum_{i=1}^{N} K_h(\Delta - \Delta^{(i)}), \qquad \Delta^{(i)} \sim f(\cdot \mid \boldsymbol{\theta}^{(i)}),\ \boldsymbol{\theta}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s}_o), \tag{4.55}$$

for a suitable kernel function K_h with bandwidth h.

In our experiments, however, we compare the approximation $\pi_b(\Delta \mid \boldsymbol{s}_o)$ with the ground truth $\pi_a(\Delta \mid \boldsymbol{s}_o)$, i.e. discrepancies obtained with the ground truth posterior. In particular, we evaluate the absolute differences between the empirical means and empirical variances[21]:

$$\text{PPDME} = |\mathbb{E}_{\pi_a}[\Delta] - \mathbb{E}_{\pi_b}[\Delta]| \approx \left|\overline{\Delta}_a - \overline{\Delta}_b\right|, \tag{4.56}$$

$$\text{PPDVE} = |\mathbb{V}_{\pi_a}[\Delta] - \mathbb{V}_{\pi_b}[\Delta]| \tag{4.57}$$

$$\approx \frac{1}{N-1} \left| \sum_{i=1}^{N} (\Delta_a^{(i)} - \overline{\Delta}_a)^2 + \sum_{i=1}^{N} (\Delta_b^{(i)} - \overline{\Delta}_b)^2 \right|, \tag{4.58}$$

[20] Recall that sampling $\Delta_{\boldsymbol{\theta}} \sim f(\cdot \mid \boldsymbol{\theta})$ is equivalent to sampling a dataset from a simulator conditioned on input parameters, $\boldsymbol{y}_{\boldsymbol{\theta}} \sim f(\cdot \mid \boldsymbol{\theta})$, and then reducing it to a vector of summary statistics $s_{\boldsymbol{\theta}} = S(y_{\boldsymbol{\theta}})$ to compute a discrepancy $\Delta_{\boldsymbol{\theta}} = d(\boldsymbol{s}_{\boldsymbol{\theta}}, \boldsymbol{s}_o)$.

[21] PPDME and PPDVE stand for posterior predictive discrepancy mean and variance errors respectively. Differences between certain quantiles are also computed, but not included in this document.

where

$$\overline{\Delta}_a = \frac{1}{N}\sum_{i=1}^{N}\Delta_a^{(i)}, \qquad \overline{\Delta}_b = \frac{1}{N}\sum_{i=1}^{N}\Delta_b^{(i)}. \tag{4.59}$$

Finally, we compute the Kullback-Leibler divergence between the kernel density estimates $\pi_a^{kde}(\Delta \mid \boldsymbol{s}_o)$, $\pi_b^{kde}(\Delta \mid \boldsymbol{s}_o)$.

$$\text{PPDKL} = \int \pi_a^{kde}(\Delta \mid \boldsymbol{s}_o)\left(\log \pi_a^{kde}(\Delta \mid \boldsymbol{s}_o) - \log \pi_b^{kde}(\Delta \mid \boldsymbol{s}_o)\right) d\Delta. \tag{4.60}$$

The integral is approximated numerically and the kernel density estimates are determined using Gaussian kernels whose bandwidths are set according to Scott's Rule [87][22]. The kernel density estimates are also used for visualization purposes.

[22] In particular, we use a function whose implementation is provided by the SciPy package [47].

Chapter 5

Evaluation

5.1 Preliminaries

Before proceeding with the description and analysis of the experiments, a note on the implementation and associated challenges is in order.

All models and methods in this chapter are implemented in Python. Importantly, the developed toolbox includes the standard BOLFI framework, as described in Sections 2.5 and 2.6, as well as the proposed extensions involving additive Gaussian processes (Section 4.1) and structure discovery via MCMC (Sections 4.1.1 and 4.1.2). In particular, BOLFI is a class that inherits from a BO superclass. The latter contains methods related to data acquisition, hyperparameter optimization and structure discovery. This part of the code is designed to be as modular as possible. For instance, acquisition functions and structure discovery samplers are implemented as separate classes. A Python implementation of LCBSC is available as part of the Engine for Likelihood-Free Inference (ELFI) package [56]. LCBSCA is thus an extension of this rule. Regarding the structure discovery module, the transition kernel of the MH sampler is based on a private implementation kindly provided by Jacob Gardner [28]. Minor bugs involving the computation of transition probabilities had to be fixed. The sampler itself was developed independently and can be extended with other transition kernels. In addition, the original Gibbs sampler is implemented in MATLAB [97]. However, as detailed in Section 4.1.2, our version is different, containing a number of code optimizations and a different scan order.

Despite a different interface, the BOLFI class contains similar methods to those found

in ELFI [56]. Relevant methods are related to the model-based nonparametric likelihood approximation and sampling-based inference. In particular, the latter is by default performed according to the NUTS algorithm [44]. MCMC diagnostics include PSRF and ESS (Section 2.3.1). An important addition is the Laplace approximation (Section 4.2.1 and Appendix A). Regarding the performance measures, the MMD implementation is based on [69], whereas all others measures have been developed independently. The toolbox also contains rejection ABC and SMC-ABC, among other functions.

Finally, it should be emphasized that BOLFI and BO in general crucially rely on an accurate surrogate model, which in our case is given by an (additive) Gaussian process regression model. Many Python packages are able to define and fit these models, but GPy is arguably the most mature [34], providing support to other packages such as ELFI. For this reason, we adopted GPy as part of our toolbox. Developing code based on this package revealed however to be a challenge. For instance, Gaussian processes are often specified without a mean function, but this work required the specification of one, in particular the mean function given by Equation (2.34). At first, it was not obvious how to define custom mean functions, but the problem eventually was solved. At that point, we were expecting the package to be fairly robust, but we found that it had unexpected bugs. A severe bug was related to predictive gradients. Other bugs were related to the specification of priors for the hyperparameters (penalized maximum likelihood). These problems were fixed, but caused unforeseen delays in the development of this project. Other challenges are discussed in the next section.

5.2 General Setup

For ease of reference, this section provides a description of the common methods, parameters and assumptions used across all experiments. Challenges in carrying the experiments are briefly discussed at the end.

As discussed in Section 4.2, one motivation of this work is to assess whether additive Gaussian process models are able to improve the efficiency of the BOLFI nonparametric estimator, when compared to the original method used in [36]. We mentioned that the comparison with ABC samplers would be possible, but not necessarily fair. Several reasons have already been introduced, but allow us to restate the differences from perhaps a more practical perspective. In BOLFI, it is assumed that sampling from the

simulator is expensive and, consequently, the number of queries should be minimized. This in turn means that the primary focus should lie on approximating the likelihood function via a surrogate model. Bayesian inference is then performed by taking the model-based approximation as the likelihood function. Importantly, this step does not query the simulator. On the other hand, ABC samplers rely on ad hoc approximations to the likelihood function in order to obtain posterior samples. In models where sampling is fast, ABC should be preferred, since the cost of performing Bayesian optimization dominates. BOLFI provides a better solution otherwise. As a result, both approaches seem viable, but under different circumstances. However, it is plausible that as the number of parameters of a simulator increases, the sampling process becomes increasingly difficult. BOLFI should then be preferred.

Thus, all methods we consider belong to the BOLFI framework:

1. **SSE**: Standard squared exponential kernel as defined in Equation (2.35).
2. **FAdd**: Fully additive squared exponential kernel, i.e. one kernel per dimension.
3. **Gibbs**: The initial kernel is SSE, but the partition varies according to the Gibbs sampler.
4. **MH**: The initial kernel is again SSE. A Metropolis-Hastings sampler proposes the partition.

We use a single shared lengthscale per kernel, which is a reasonable assumption for the type of models we consider. Moreover, since it is relatively costly to run structure discovery, the updates only occur at every 10 acquisitions and the number of marginal likelihood evaluations is 50. In all methods, the BO algorithm is initialized with 20 acquisitions, obtained deterministically according to a Sobol sequence, as done in [36], and 500 data points are further acquired. The search space is a hypercube $[-3,3]^n$, where n is the number of dimensions. Unlike [36], the acquisition function is optimized using the L-BFGS-B optimizer [56]. The corresponding number of maximum iterations is 1000 and the number of restarts is $\max(10,n)$. We observed that a deterministic initialization (Sobol), as opposed to random (uniform) sampling, led to better results. Despite the discussion in Section 2.6, we use a stochastic acquisition function given by LCBSCA and isotropic Gaussian noise with variance 0.1 is added to the corresponding minimizer. Some of these values have been finetuned by checking the performance over a number of pilot runs. For instance, we observed that decreasing the frequency of the updates in structure discovery led to worse results. Similarly, de-

creasing the number of likelihood evaluations also led to inferior results. The same occurred with the acquisition optimizer. Due to time and computing constraints, we were unable to determine whether the deterministic acquisition rule performs better.

Importantly, the prior distribution $\pi(\boldsymbol{\theta})$ is a uniform distribution defined in the hypercube $[-3,3]^n$. This in turn means that the shapes of the likelihood function and the posterior distribution are the equivalent. Assessing the quality of the likelihood approximation is equivalent to assessing the quality of the posterior approximation. For instance, the Laplace approximation to the BOLFI nonparametric likelihood approximation is an approximation to the posterior. Hence, the Kullback-Leibler (KL) divergence defined in Equation (4.30) can be included as a performance measure of the posterior approximation.

A particular goal of this experimental work was to continually monitor the performance of the different methods. Performance measures that are computationally efficient are performed at every 10 acquisitions, while the others are computed at every 50 acquisitions. In particular, the computationally lightweight metrics correspond to those in Sections 4.2.1 and 4.2.2. The metric DiscError (discrepancy error) is considered lightweight because the test locations only need to be sampled once from the ground truth posterior (at the start of each new run). All measures defined in Section 4.2.3 require samples from the ground truth posterior and the posterior approximation. As previously mentioned, the ground truth posterior is only sampled once, the posterior approximation needs however to be sampled repeatedly every 50 acquisitions. The total number of samples (not ESS) is 5000, acquired via NUTS (1 chain, with a target probability of 0.7) [56]. It should be mentioned that sampling from the model-based approximation can be exceedingly expensive. In fact, in many cases, the time spent sampling dominated the time it took to run BOLFI. It was particularly costly for $n \geq 12$. We were committed to obtaining a comprehensive set of results, but it had a negative impact on the number of different experiments we were able to perform. Each experiment was repeated 10 times[1] and, in the next sections, the reported values correspond to means and one standard deviations. The time it took to obtain each set ranged from days to a week. Experiments were run in parallel across 20 different machines, each with at least 40 CPU cores. Unfortunately, since the machines had different specifications and were subject to different loads, it was not possible to obtain reliable estimates of computational costs (time).

[1] Unless stated otherwise.

5.3 Models

In this work, we follow a modeling approach where the data generating process is defined directly in terms of the discrepancy. The motivation is that summary statistics are not particularly relevant to our analyses, since the focus is on the computational aspect of the approximations. We have stressed their importance in likelihood-free inference, but a correct specification of such statistics is an orthogonal problem. Furthermore, it may be argued that the models we define in this work are simple and perhaps too artificial, not reflecting the challenges of realistic simulators. While possibly true, the objective was to start from first principles. If the methods we consider do not perform well under these circumstances, then we cannot expect them to yield good results in more challenging scenarios. A similar argument applies to the performance measures.

In Section 2.6, we discussed that the discrepancy can be interpreted as a negative log likelihood of an auxiliary model. If we assume a conditionally normal distribution, as in synthetic likelihood, the log discrepancy can be written as[2]:

$$g^\star(\boldsymbol{\theta}) = \log \Delta^\star_{\boldsymbol{\theta}} = \frac{1}{2}\log|2\pi\boldsymbol{\Sigma}_{\boldsymbol{s}}(\boldsymbol{\theta})| + \frac{1}{2}\left(\boldsymbol{s}_{\boldsymbol{\theta}_o} - \boldsymbol{\mu}_{\boldsymbol{s}}(\boldsymbol{\theta})\right)^\intercal \boldsymbol{\Sigma}_{\boldsymbol{s}}^{-1}(\boldsymbol{\theta})\left(\boldsymbol{s}_{\boldsymbol{\theta}_o} - \boldsymbol{\mu}_{\boldsymbol{s}}(\boldsymbol{\theta})\right). \quad (5.1)$$

Assuming that the covariance matrix is fixed for any $\boldsymbol{\theta}$ and that the vectors of summary statistics are the parameters, we can write the log discrepancy as a quadratic form (dropping the constant term):

$$g^\star(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta}_o - \boldsymbol{\theta})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}_o - \boldsymbol{\theta}). \quad (5.2)$$

At this point, we refer to $g^\star$ simply as discrepancy. If we further assume that $\boldsymbol{\Sigma}$ is a block diagonal matrix:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Sigma}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\Sigma}_G \end{pmatrix}, \quad (5.3)$$

the discrepancy can be written as[3]:

$$g^\star(\boldsymbol{\theta}) = \sum_{i=1}^{G} \frac{1}{2}\left(\boldsymbol{\theta}_{o_{P_i}} - \boldsymbol{\theta}_{P_i}\right)^\intercal \boldsymbol{\Sigma}_i^{-1}\left(\boldsymbol{\theta}_{o_{P_i}} - \boldsymbol{\theta}_{P_i}\right) = \sum_{i=1}^{G} g_i^\star(\boldsymbol{\theta}_{P_i}). \quad (5.4)$$

[2] Previously defined in Equation (2.56).
[3] Recall the notation in Section 4.1.

The function $g^\star$ is thus additive with groups $P_1, ..., P_G$. Under the BOLFI assumption regarding the normality of the observed discrepancies, we can write:

$$g(\boldsymbol{\theta}) = \sum_{i=1}^{G} g_i^\star(\boldsymbol{\theta}_{P_i}) + \zeta, \qquad \zeta \sim \mathcal{N}(0, \sigma_n^2), \tag{5.5}$$

where σ_n^2 is the observation noise. Recalling Equation (2.52), with a uniform kernel and bandwidth ε, we can let the ground truth likelihood be the BOLFI nonparametric likelihood:

$$f(\boldsymbol{\theta}_o \mid \boldsymbol{\theta}) \propto \Pr\left(g(\boldsymbol{\theta}) \leq \varepsilon \mid g(\boldsymbol{\theta}) \sim \mathcal{N}(g^\star(\boldsymbol{\theta}), \sigma_n^2)\right) \tag{5.6}$$

$$= F_{\mathcal{N}}\left(\frac{\varepsilon - g^\star(\boldsymbol{\theta})}{\sigma_n}\right). \tag{5.7}$$

This is one of the models we test. In fact, since g^* is modeled parametrically, we refer to this model as the *parametric model*. It is not however the first model to be tested. The reason is that in order to fit an approximation, hyperparameter optimization is required. It may seem a trivial task, but it is a possible source of estimation error and the process can be computationally costly if there is the need to perform it repeatedly, e.g. structure discovery. For these reasons, we attempt to generate Equation (5.4) semiparametrically, which we refer as the *semiparametric model*. Note that in both models, the ground truth likelihood is given by Equation (5.7). In all experiments, we set $\sigma_n^2 = 0.01$, $\varepsilon = g^\star(\boldsymbol{\theta}_o)$ and $\boldsymbol{\theta}_o = (0.5, ..., 0.5)^\intercal$[4]. All matrices $\boldsymbol{\Sigma}_i$ are defined as correlation matrices, with 0.6 in all off-diagonal entries. We consider two type of problems: one where all $\boldsymbol{\Sigma}_i$ are 2×2 matrices, and the other where the corresponding shapes are 4×4. We further assume that the approximations know the value of σ_n^2.

Regarding the semiparametric generation of the quadratic form in Equation (5.4), we exploit the following result:

$$g^\star(\boldsymbol{\theta}) = \sum_{j=1}^{n} a_j \theta_j^2 + \left(\sum_{i=1}^{G} \frac{1}{2} \left(\boldsymbol{\theta}_{o_{P_i}} - \boldsymbol{\theta}_{P_i}\right)^\intercal \boldsymbol{\Sigma}_i^{-1} \left(\boldsymbol{\theta}_{o_{P_i}} - \boldsymbol{\theta}_{P_i}\right) - \sum_{j=1}^{n} a_j \theta_j^2 \right) \tag{5.8}$$

$$= \sum_{j=1}^{n} a_j \theta_j^2 + \psi(\boldsymbol{\theta}), \tag{5.9}$$

where $a_j = 1/2[\boldsymbol{\Sigma}_1^{-1}]_{(1,1)}$[5]. Intuitively, the second term, $\psi(\boldsymbol{\theta})$, contains all the interactions, and is modeled nonparametrically. The ground truth discrepancy $g^\star$ is now

[4] Importantly, $\boldsymbol{\theta}_o$ needs to be redefined in the semiparametric model, as we discuss next.

[5] Note that all matrices $\boldsymbol{\Sigma}_i$ are equivalent and have the same value in all diagonal entries.

modeled as the predictive mean of a Gaussian process regression model. In particular, we define a Gaussian process prior $\mathcal{GP}(m_{g^\star}, k_{g^\star})$, where

$$m_{g^\star}(\boldsymbol{\theta}) = \sum_{j=1}^{n} a_j \theta_j^2, \tag{5.10}$$

and $k_{g^\star}$ encodes the structure of the additive model $\mathcal{M} = \{P_1, ..., P_G\}$. Importantly, $k_{g^\star}$ is a sum of squared exponential kernels $k_{g_i^\star}$, all with the same values for the length-scales ($\lambda_i = 3$) and signal variances ($\sigma_{f_i}^2 = 10$). The choice of these values is not completely arbitrary. For instance, by setting λ_i to be $1/2$ of the difference between the upper and lower bound of the search space, we are supporting functions that vary relatively slowly, which holds for $g^\star$.

In order to generate the semiparametric model, data must first be sampled from $g^\star$. The chosen locations are $\boldsymbol{\theta}_o$ and uniformly sampled locations at radii $(0.01, 0.1, 0.5, 1)$ away from $\boldsymbol{\theta}_o$. In particular, the sampling process for one location $\boldsymbol{\theta}^\bullet$ at radius r is given by:

$$\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{5.11}$$

$$\boldsymbol{\theta}^\bullet = r \cdot \boldsymbol{z} / ||\boldsymbol{z}||_2 + \boldsymbol{\theta}_o, \tag{5.12}$$

and the corresponding discrepancy is determined according to $g^\star(\boldsymbol{\theta}^\bullet)$. The number of sampled locations at each radius is 300. As a result, the semiparametric model for the discrepancy is given by the predictive mean of the corresponding Gaussian process regression model[6]:

$$\mu_{g^\star}(\boldsymbol{\theta}) = m_{g^\star}(\boldsymbol{\theta}) + \boldsymbol{k}_{g^\star}(\boldsymbol{\theta})^\intercal \boldsymbol{K}_{g^\star}^{-1} (\boldsymbol{g}^\star - \boldsymbol{m}_{g^\star}) \tag{5.13}$$

where $\boldsymbol{g}^\star$ is the vector containing all sampled function values $\{g^\star(\boldsymbol{\theta}^\bullet)\}$, and $\boldsymbol{m}_{g^\star}$, $\boldsymbol{K}_{g^\star}$ are defined as in Equation (2.42) but for the set of sampled locations $\{\boldsymbol{\theta}^\bullet\}$, mean function $m_{g^\star}$ and kernel $k_{g^\star}$. The same reasoning applies to $\boldsymbol{k}_{g^\star}$ and Equation (2.41). At this point, we can redefine $g^\star(\boldsymbol{\theta}) = \mu_{g^\star}(\boldsymbol{\theta})$. Importantly, the minimum is no longer guaranteed to be $\boldsymbol{\theta}_o$, but in practice it will be a point in the neighborhood. For this reason, we find the new $\boldsymbol{\theta}_o$:

$$\boldsymbol{\theta}_o = \arg\min_{\boldsymbol{\theta}} g^\star(\boldsymbol{\theta}). \tag{5.14}$$

The optimizer is the same as the one used for data acquisition in BO. The semiparametric model is used in the first two experiments. In both cases, we assume the mean

[6] Recall the definition given in Equation (2.44).

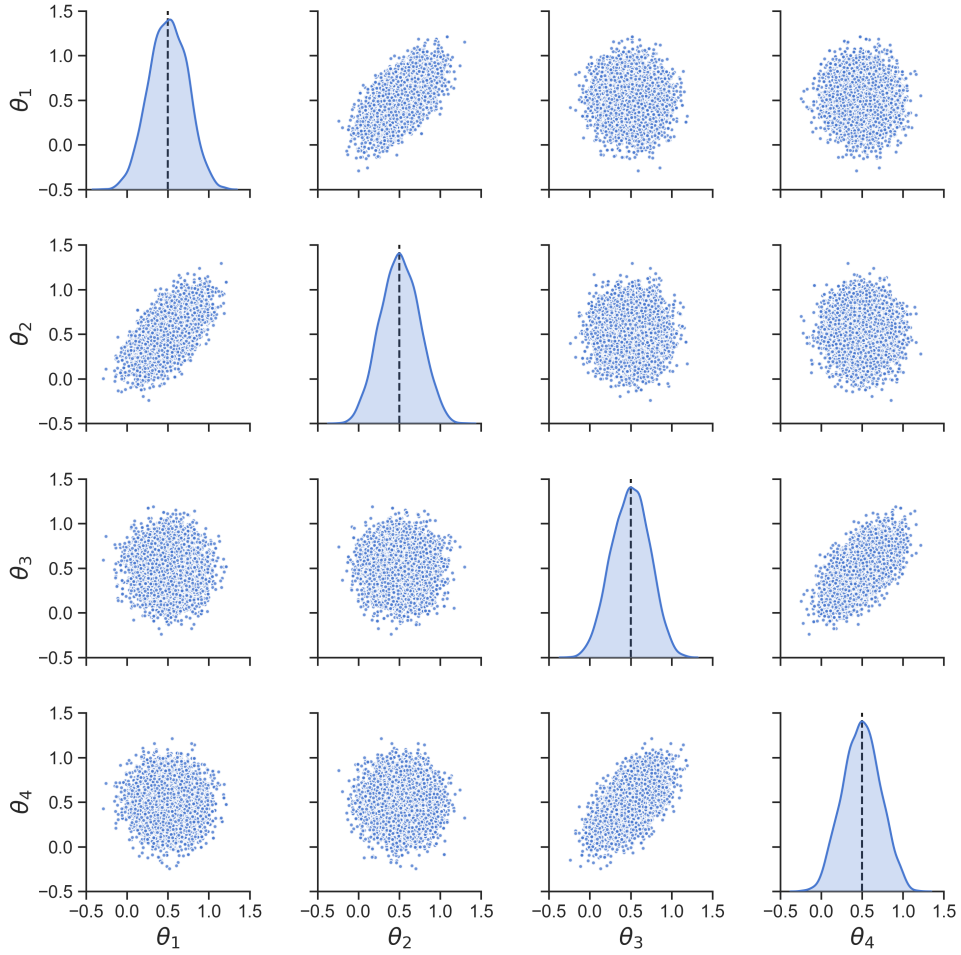

Figure 5.1: Normalized ground truth likelihood function of the parametric model.

function m_{g^*} to be known. Hence, the focus lies on the approximation to the (additive) nonparametric interaction term. Hyperparameters such as lengthscales and signal variances are also assumed known. We discuss the implications of the latter assumption in the next section.

At this point, it might be instructive to consider a specific example that shows the differences and similarities between the two models. Figures 5.1 and 5.2 show the normalized ground truth likelihood function[7] of the parametric and semiparametric models respectively. Both have been generated with $n = 4$ and 2×2 group matrices. Since the discrepancy function is a quadratic function in the parametric model and a (stochastic) function that emulates a quadratic in the semiparametric model, the

[7] As previously discussed, the normalized likelihood function is equivalent to the posterior distribution (uniform prior). More precisely, the equivalence is between a truncated version of the likelihood function and the posterior distribution. Nonetheless, both terms are used interchangeably.

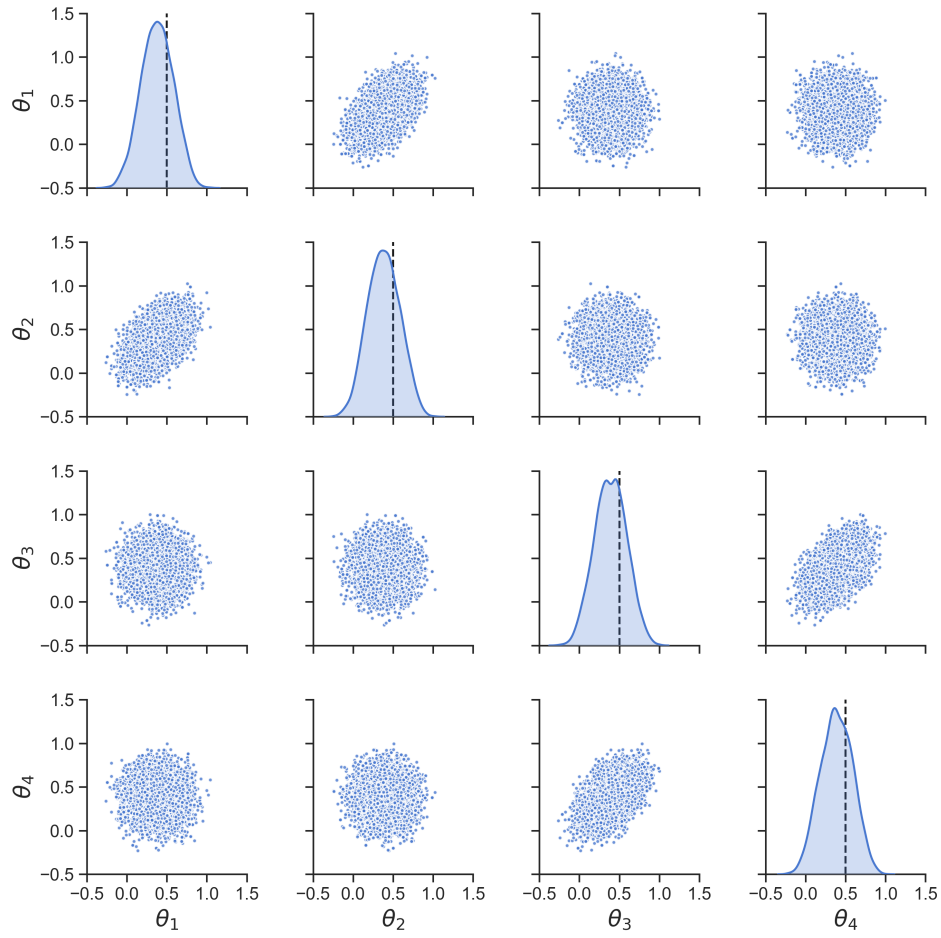

Figure 5.2: Normalized ground truth likelihood function of the semiparametric model.

(empirical) multivariate distributions resemble multivariate normal distributions[8]. In particular, a normal distribution where dimensions within a group are correlated, but independent otherwise. Notice that the mode is near $(0.5, ..., 0.5)$ in the semiparametric model. The mode changes each time the discrepancy function is generated.

Before proceeding with the analysis of the results, a final note on the use of the quadratic mean function and the assumption that the function is known in the semiparametric model.

In Section 4.2.1, we introduced the indicator NearB that assesses whether the expected minimum is near the boundary of the search space. We mentioned that a relatively unknown challenge in high-dimensional Bayesian optimization is the boundary issue, where the algorithm unintuitively spends a significant amount of time acquiring data

[8]This in turn suggests that the Laplace approximation provides a good approximation to the ground truth posterior distribution of both models.

near the boundary, but deferred a detailed explanation to this chapter. As stated in [93], the boundary search problem is a consequence of the curse of dimensionality and how Gaussian processes handle uncertainty. A common assumption in Bayesian optimization is that the search space is defined in such way that optimum values are expected to be near the center. On the other hand, if the BO algorithm has not acquired enough data, minimization of acquisition functions such as LCBSC (Equation (2.45)) implies that if the mean function is not sufficiently informative, as is the case of a zero (prior) mean function, the uncertainty term dominates. The focus is then on points that are distant from those that have been acquired thus far. As a result, the BO algorithm first explores regions near the boundary, since the search space is bounded (often a hypercube). Importantly, this behavior also occurs in low-dimensional spaces, but only becomes a problem as the number of dimensions increases. For instance, the number of corner points in a hypercube increases exponentially, which is disastrous for Bayesian optimization and BOLFI in particular, potentially reducing the statistical efficiency of the latter to a significant extent.

Possible solutions are to warp the search space such that the region near the center is expanded and the region near the boundaries is contracted or, alternatively, to introduce a quadratic mean function[9] [93]. In this work, we adopt the latter. However, a natural question that arises is related to the robustness of the Bayesian optimization process when the hyperparameters are learned via maximum likelihood. Indeed, the mean function parameters also need to be learned, and if the coefficients are set too small, the boundary issue may resurface. We argue that it is thus important to either adopt a full Bayesian approach by integrating the parameters out, or at least to learn the parameters via penalized maximum likelihood, where the regularization terms should be chosen carefully[10]. As previously mentioned, we do not adopt a full Bayesian approach due to time and computing constraints. Instead, we explore the use of penalized maximum likelihood[11] in the last experiment (parametric model). In the first two experiments (semiparametric model), the boundary issue is mitigated by explicitly defining the mean function of the approximation to be that of the ground truth, $m_{g^\star}$.

[9] Arguably, it could be any even degree polynomial.

[10] A full Bayesian approach may possibly be the only way to address the (lack of) robustness problem in situations where Gaussian process regression is to be treated as a black-box. In BOLFI, the question is whether non-Bayesian approaches can effectively be used for black-box high-dimensional inference.

[11] It is not clear to us whether GPy [34] jointly optimizes all hyperparameters. It seems to be the case. An alternative solution is then to optimize the hyperparameters in two stages, starting with those related to the mean function. A different package may be more appropriate.

5.4 Experiment 1: Semiparametric Model (2x2)

Thus far, we have introduced the methods and the models. We mentioned that in the semiparametric model, the hyperparameters are assumed to be known, avoiding the computational cost of optimization and the resulting estimation errors. The potential problem however is that data is generated according to a model with a certain additive structure, specifically each group has 2 dimensions, or variables. For the structure-learning methods, Gibbs and MH, it is not problematic to consider that the hyperparameters are that of the ground truth. If these methods work as expected, the ground truth partition is eventually recovered. On the other hand, a static partition is used in SSE and FAdd. The former assumes that all dimensions interact, whereas the latter assumes no interactions. Hence, setting a signal variance that is the same as the ground truth has different implications. In fact, the prior signal variance, at any given location, is $(n/2)\cdot\sigma_f^2$ for the ground truth, σ_f^2 for SSE and $n\cdot\sigma_f^2$ for FAdd. Thus, the question that arises is whether it is justifiable to proceed with the above definition. If we interpret SSE and FAdd as baselines that inform how the structure-learning methods would perform if they were to set a single group or one group per dimension, then it seems justifiable. However, it is also reasonable to define the signal variances according to a prior assumption on how the ground truth function is expected to behave. This in turn suggests to match the prior signal variances, in which case the value for SSE should be scaled up, $(n/2)\cdot\sigma_f^2$, and the value for FAdd should be scaled down, $(1/2)\cdot\sigma_f^2$. In order to avoid future confusion, we define the resulting methods as SSE(S) and FAdd(S), where S stands for scaled. In this experiment, we test both approaches.

Furthermore, in Section 4.1, we raised the question whether the formula for η_t^2 should remain the same in the additive case[12], and then mentioned that, based on work from a fellow student [24], the same formula was kept. The motivation was that our initial aim was not to test different acquisition rules. Although based on the previous discussion of the boundary issue, it should now be clear that a poorly calibrated η_t^2, in particular one that is set too large, can potentially aggravate the issue. The original formula for the trade-off parameter is:

$$\eta_t^2 = 2\log[t^{2n+2}\pi^2/(3\varepsilon_\eta)], \tag{5.15}$$

where $\varepsilon_\eta = 0.1$. Notice the dependence with n, i.e. the total number of dimensions.

[12]Recall that η_t^2 is an adjustable parameter in LCBSC and LCBSCA that controls the trade-off between exploitation and exploration.

As the number of dimensions increases, η_t^2 becomes larger. This relationship in itself is not harmful. Indeed, as the volume of the search space increases, more emphasis needs to be placed on exploration so as not to miss potentially rewarding regions. However, in FAdd, each dimension is optimized separately. This in turn suggests that η_t^2 should instead depend on the number of dimensions in a group. This led us to test an alternative, as defined in [48]:

$$\eta_t^2 = 0.2d\log(2t), \tag{5.16}$$

where d is the maximum number of dimensions in any group. For FAdd, in particular, $d = 1$. In this experiment, we test both rules. However, due to time and computing constraints, we were only able to obtain results for FAdd(S), which we refer as FAdd(SA), where A stands for acquisition.

At this point, we are able to introduce the results, which have been obtained for 4, 8, 12, 16 and 20 dimensions. In addition, the methods used are SSE, SSE(S), FAdd, FAdd(S), FAdd(SA), Gibbs and MH. The reported results correspond to the means and one standard deviations over 10 runs. One-sided error band is shown in cases where a log scale is used, a two-sided band is reported otherwise[13].

Figure 5.4 shows the performance measures related to the approximation of the discrepancy function: instantaneous regret, location error, near boundary condition and discrepancy error. The first three evaluation metrics are complementary, providing information regarding the differences between the expected minimum and the ground truth minimum of the discrepancy function. On the other hand, the discrepancy error measures the quality of the approximation around the ground truth minimum or, equivalently, the region of small discrepancies[14].

In general, as the number of dimensions increases, finding the minimum and, by extension, approximating the region of small discrepancies becomes more difficult. However, this trend is more noticeable for methods that use a static partition. In the 4-dimensional problem, SSE, SSE(S), Gibbs and MH perform equally well, but the performance of SSE (and SSE(S)) rapidly degrades in higher-dimensional problems. Based on these results, it seems justifiable to use SSE in low-dimensional problems ($n \leq 4$) as it is computationally more efficient than Gibbs or MH. Structure-discovery methods should be used otherwise.

[13] Standard deviation is not computed for NearB. The corresponding average value indicates the percentage of runs where the near boundary condition is active.

[14] The ground truth discrepancy function is unimodal.

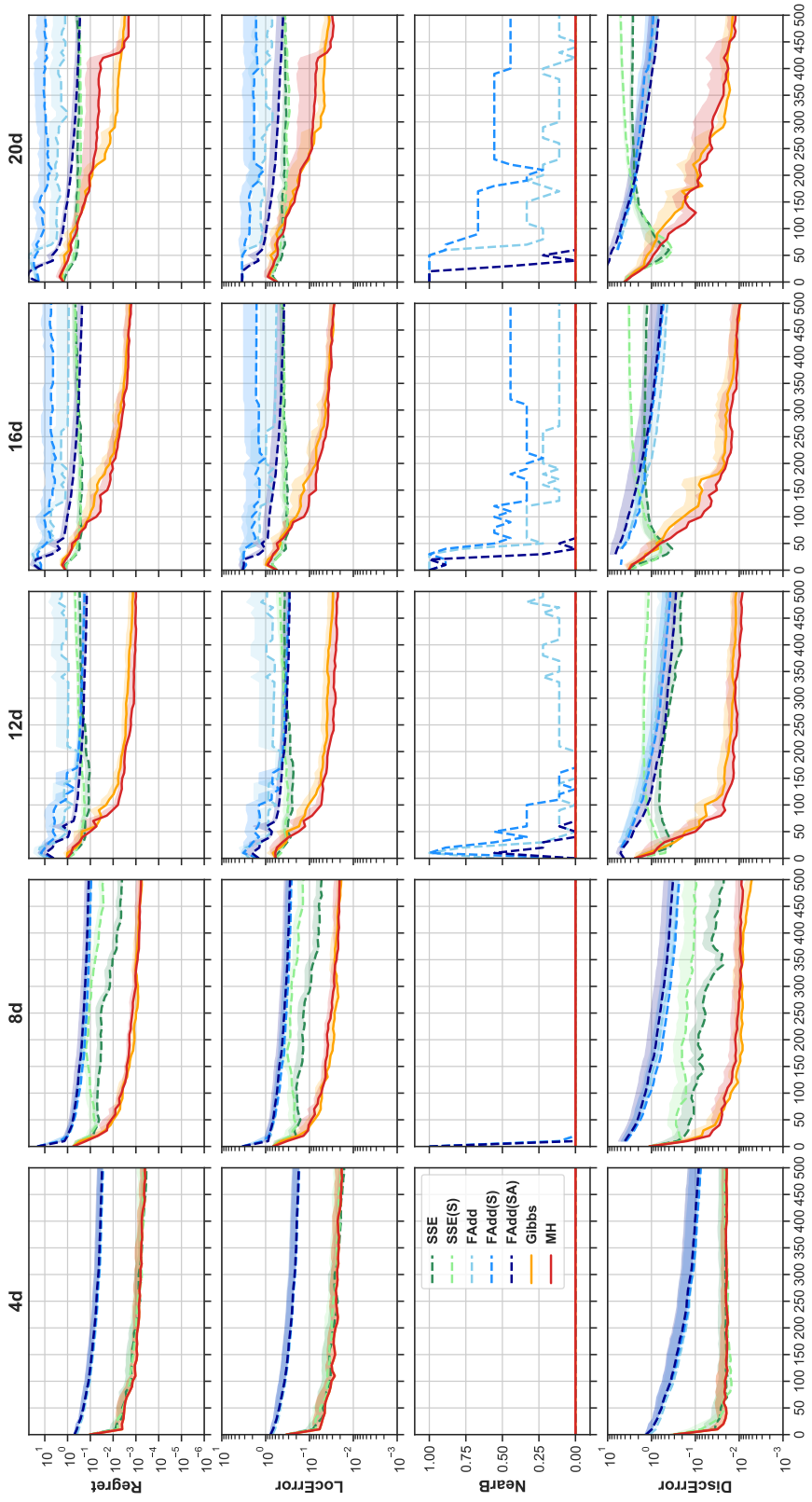

Figure 5.3: Semiparametric model (2x2): Timeplots of instantaneours regret, location error, near boundary condition and discrepancy error. Values averaged over 10 runs, one standard error band.

Interestingly, note that the performance of SSE(S) is generally worse than that of SSE. For this reason, in the next experiment, we only report the results of SSE, i.e. the signal variance is set to that of the ground truth.

We have not yet commented on the performance of the fully additive methods. Perhaps one would have expected a fully additive method to outperform SSE, since the semiparametric model (2×2) of the discrepancy function is almost fully additive (two dimensions per group). The results suggest that (static) fully additive methods should generally be avoided since they tend to yield misspecified regression models that are unable to explain interactions between dimensions. In these circumstances, since the surrogate model in BO is unable to capture the behavior of the (unknown) objective function, finding the minimum or the region near the minimum is difficult. Surprisingly, for $n \geq 16$, FAdd and variants outperformed SSE in terms of approximating the region of small discrepancies, but the approximation was admittedly still poor. FAdd(SA) seems to yield generally better results than any other variants and is effective in mitigating the boundary issue that occurs in FAdd and FAdd(S). In addition, scaling down the signal variance of FAdd did not improve performance, but due to time constraints we were unable to test FAdd(A). In the next experiment, we only report the results of FAdd(SA).

Despite some fluctuations, the results thus far seem to suggest that Gibbs and MH perform equally well. However, none of these indicators measure whether the methods are able to uncover the ground truth partition. For this reason, let us analyze the results in Figure 5.4. The evaluation metrics are accuracy, F-score, log likelihood ratio and, for convenience, the discrepancy error is again shown. Importantly, it should be noted that accuracy and F-score are only applicable to Gibbs and MH, but, for ease of reference, we also plot the values corresponding to a fully additive partition (FAdd and variants) and a fully dependent partition (SSE and variants). In this regard, it seems that, when compared to accuracy, the F-score might be a better performance measure. In the 4-dimensional problem, Gibbs and MH perform exactly the same, proposing the ground truth partition in the first update (10 acquisitions). Importantly, once the ground truth partition is found, the methods correctly reject any other candidate partitions. In higher-dimensional problems, uncovering the ground truth becomes increasingly difficult. Both seem able to identify the correct partition with relative ease in low- to medium-dimensional problems, i.e. $n < 12$, Gibbs to a lesser extent. As we discuss next, the differences become more noticeable in higher-dimensional problems.

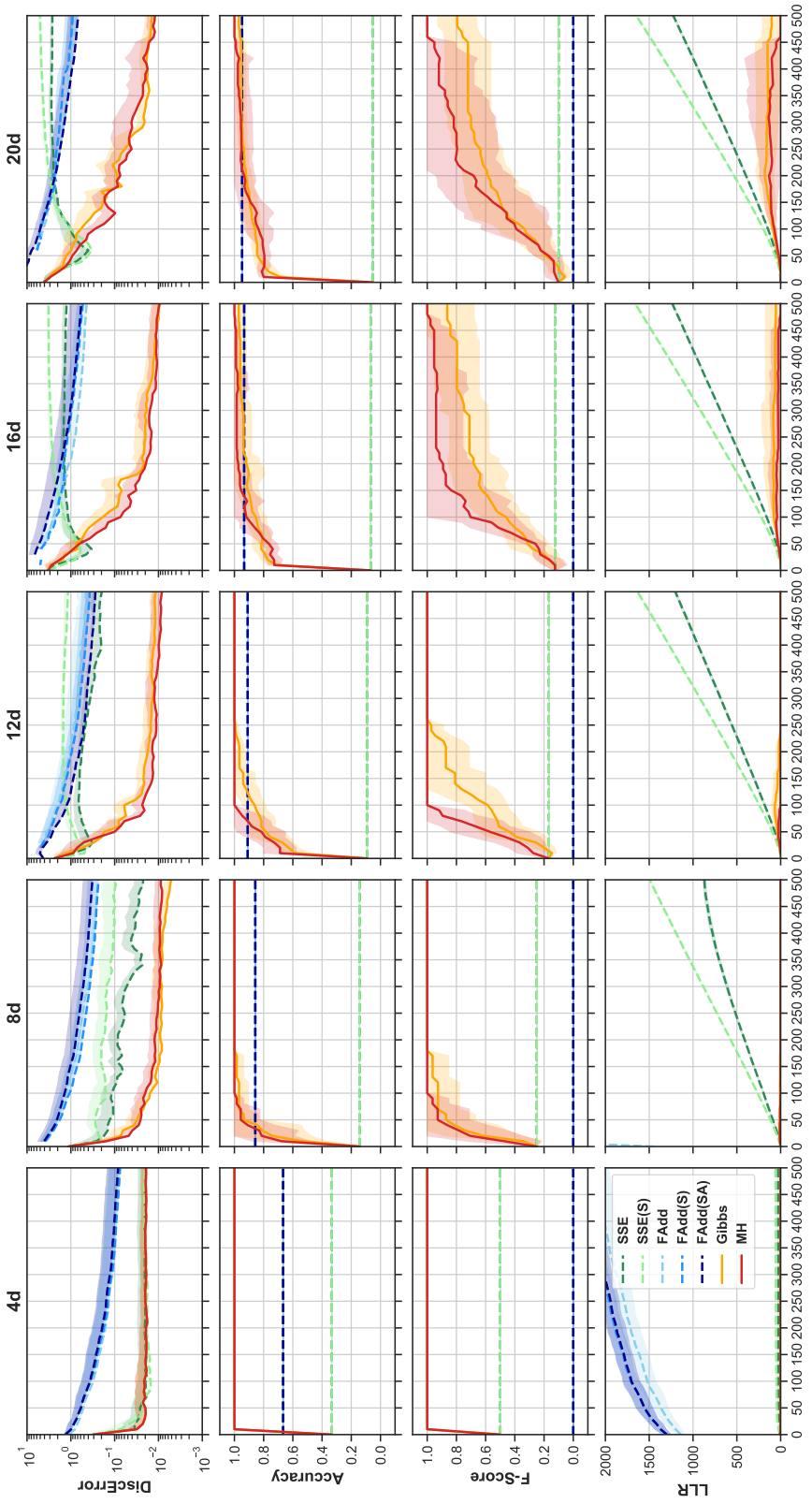

Figure 5.4: Semiparametric model (2x2): Timeplots of discrepancy error, accuracy, F-score and log likelihood ratio. Values averaged over 10 runs, one standard error band. Performance measures related to structure discovery (accuracy and F-score) are only applicable to Gibbs and MH.

Indeed, while MH is able to consistently find the correct additive structure first, it is also subject to higher variability. This can potentially be explained by the fact that MH can jump to non-neighboring partitions, whereas Gibbs cannot. The former operates on previously proposed groups and the latter on dimensions. As a result, MH can explore the partition space more quickly, but is more reliant on previous decisions, namely previous split and merge operations. In any case, it is reassuring to find that as the number of acquisitions increases, both methods are able to provide partitions that increasingly resemble that of the ground truth. Regarding the log likelihood ratio, the differences between structure-discovery methods and others is remarkable. The large values of SSE and FAdd indicate that the methods fit the data poorly. In fact, the fit is so poor in FAdd (and variants) that the corresponding curves are not shown for $n \geq 8$. On the other hand, the values observed for MH and Gibbs are significantly smaller, being zero once the correct partition is uncovered.

Finding the region close to the minimum and proposing partitions that resemble that of the ground truth is certainly important, but only a step towards accurate inference. Figure 5.5 shows the timeplots of the performance measures that assess the quality of the posterior approximation. Importantly, except for KL, all evaluation metrics rely on posterior sampling[15]. Figure B.1 (Appendix B) contains the related diagnostics, namely ESS and PSRF. Effective sample sizes tend to be larger[16] than 3000 and PSRF values are close to 1, suggesting convergence.

First, a note related to the performance measures. It seems that the KL (divergence between Laplace approximations), the maximum mean discrepancy (MMD) and the posterior predictive discrepancy errors (mean and var) are generally in agreement. The performance curves of KL and MMD are quite similar. As mentioned in Section 5.3, the Laplace approximation is known to work reasonably well, because the ground truth posterior distribution resembles that of a normal, albeit with lighter tails. Still, it is reassuring to observe that both measures are in such strong agreement, providing further evidence that MMD is a powerful test even in relatively high dimensions. Regarding the posterior predictive discrepancy errors (PPDE), it seems that PPDME and PPDVE can provide useful information, similar to a certain extent to that of KL and MMD.

[15]None of these measures are computed if NearB is active. Hence, the reported results for FAdd and FAdd(S) are optimistic in some cases.

[16]A notable exception is SSE(S). Scaling up the signal variance in SSE allows a wider range of fluctuations. It might be that the support of the corresponding posterior is narrow, causing sticky behavior. The reasons are not completely clear. Traceplots could have been used to find the cause. However, at this point, we have already established that SSE is better than SSE(S).

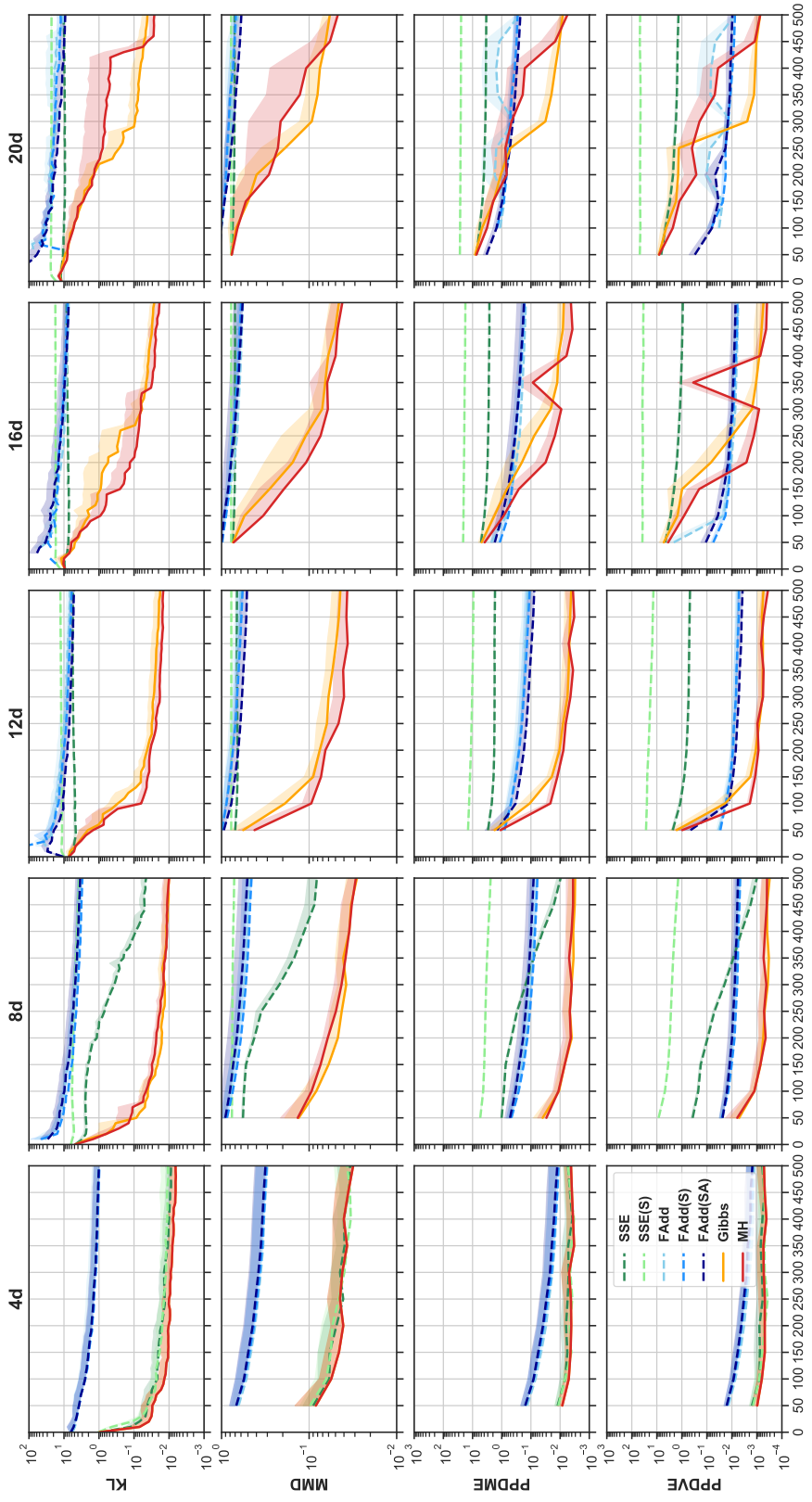

Figure 5.5: Semiparametric model (2x2): Timeplots of Kullback-Leibler divergence with Laplace approximations, maximum mean discrepancy and posterior predictive discrepancy mean and variance errors. Values averaged over 10 runs, one standard error band.

A peculiarity of PPDE seems to be that large values indicate a poor approximation, but low values do not necessarily imply an accurate approximation. For instance, PPDE suggests that FAdd(SA) is better than SSE, but, according to KL and MMD, both methods provide equally poor approximations to high-dimensional distributions. So, in this regard, PPDE seems to provide limited information. Nonetheless, these performance measures should be seen as complementary and not as alternatives.

Regarding the methods, we observe that, in low-dimensional problems ($n \leq 4$), SSE is able to provide an approximation that is as good as Gibbs and MH. Interestingly, the approximation given by SSE(S) is just as accurate, revealing that there is a certain insensitivity to the choice of hyperparameters. However, due to the curse of dimensionality, the differences become noticeable as the number of dimensions increases. In any case, Gibbs and MH yield the most efficient estimators, never requiring more data than any other method. In practice, if sampling from the simulator is indeed costly, Gibbs and MH should be chosen over others, even in low-dimensional problems. Naturally, the question that arises is whether MH should be preferred over Gibbs, or vice-versa. Based on these results, it seems difficult to establish that one method is clearly better than the other. Both reach equally good approximations after 500 acquisitions. MH seems slightly more efficient than Gibbs in 12- and 16-dimensional problems, whereas Gibbs is able to approximate the posterior sufficiently well after 300 acquisitions in the 20-dimensional problem, as opposed to 450 acquisitions with MH.

At this point, it is instructive to show the posterior predictive discrepancy distributions (for a particular run). Figure 5.6 depicts the corresponding distributions obtained with SSE, FAdd(SA) and RABC, where RABC stands for rejection ABC (Algorithm 3) and is included in this analysis only to reinforce the statement that a PPD analysis is also applicable to inference methods that do not rely on the BOLFI framework. Similarly, it serves to demonstrate that in order to benchmark different methods, it might suffice to specify the simulator-based models in terms of a stochastic discrepancy function. Rejection ABC, in this case, acquires 500,000 samples from the pseudo-simulator[17] and forms an empirical approximation to the posterior distribution based on the 5000 samples that yield the lowest discrepancies (equivalently, the threshold is the 1st percentile). In comparison to BOLFI methods, RABC queries the pseudo-simulator by a factor that is almost as high as 1000. Before proceeding with a brief analysis, allow us to also introduce the PPD obtained with Gibbs and MH, both shown in Figure 5.7.

[17] Recall that the output is a realized discrepancy.

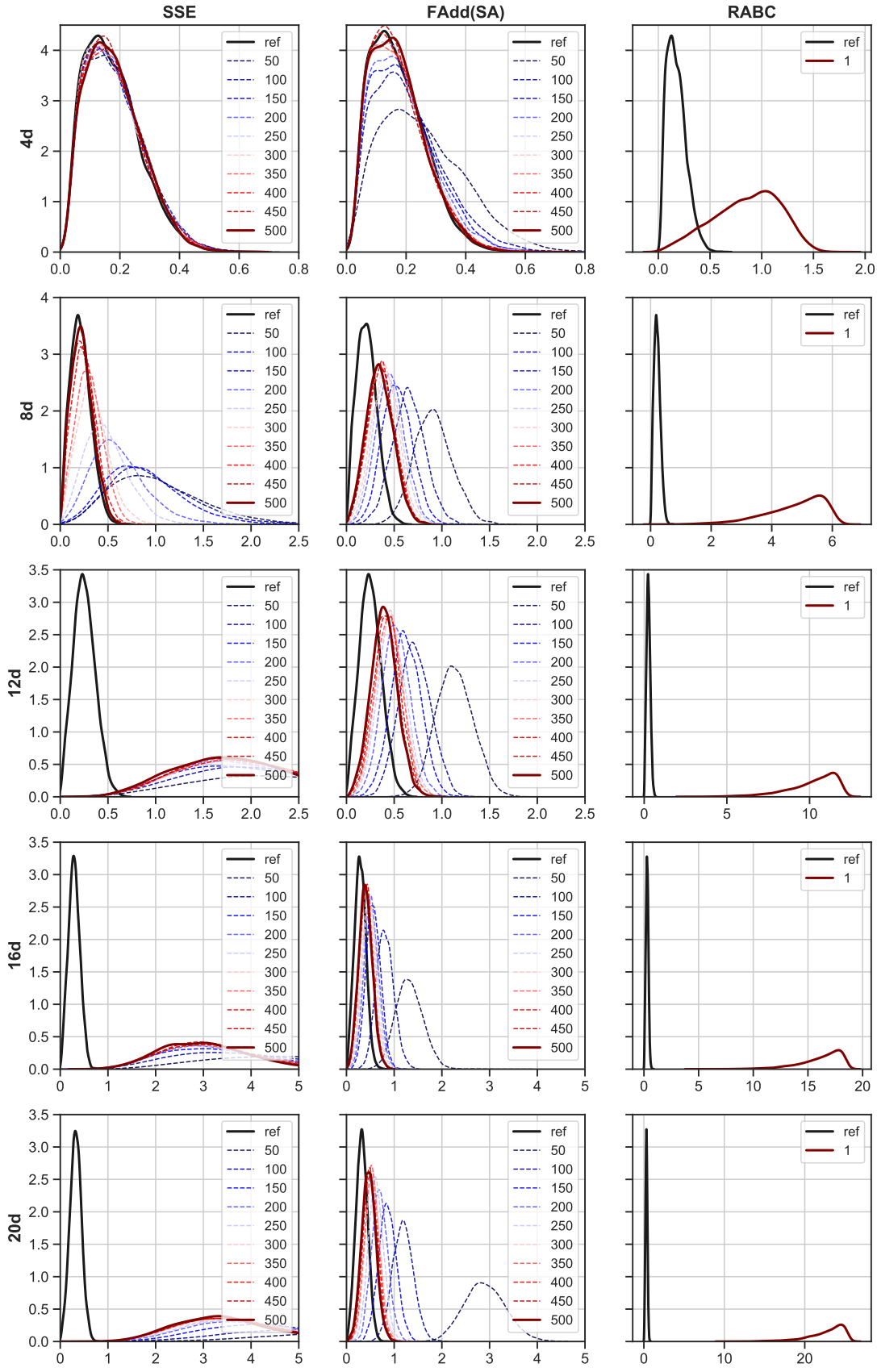

Figure 5.6: Semiparametric model (2x2): Posterior predictive discrepancy distributions obtained with SSE, FAdd(SA) and RABC. ref corresponds to the posterior predictive discrepancy distribution obtained with the ground truth posterior distribution.

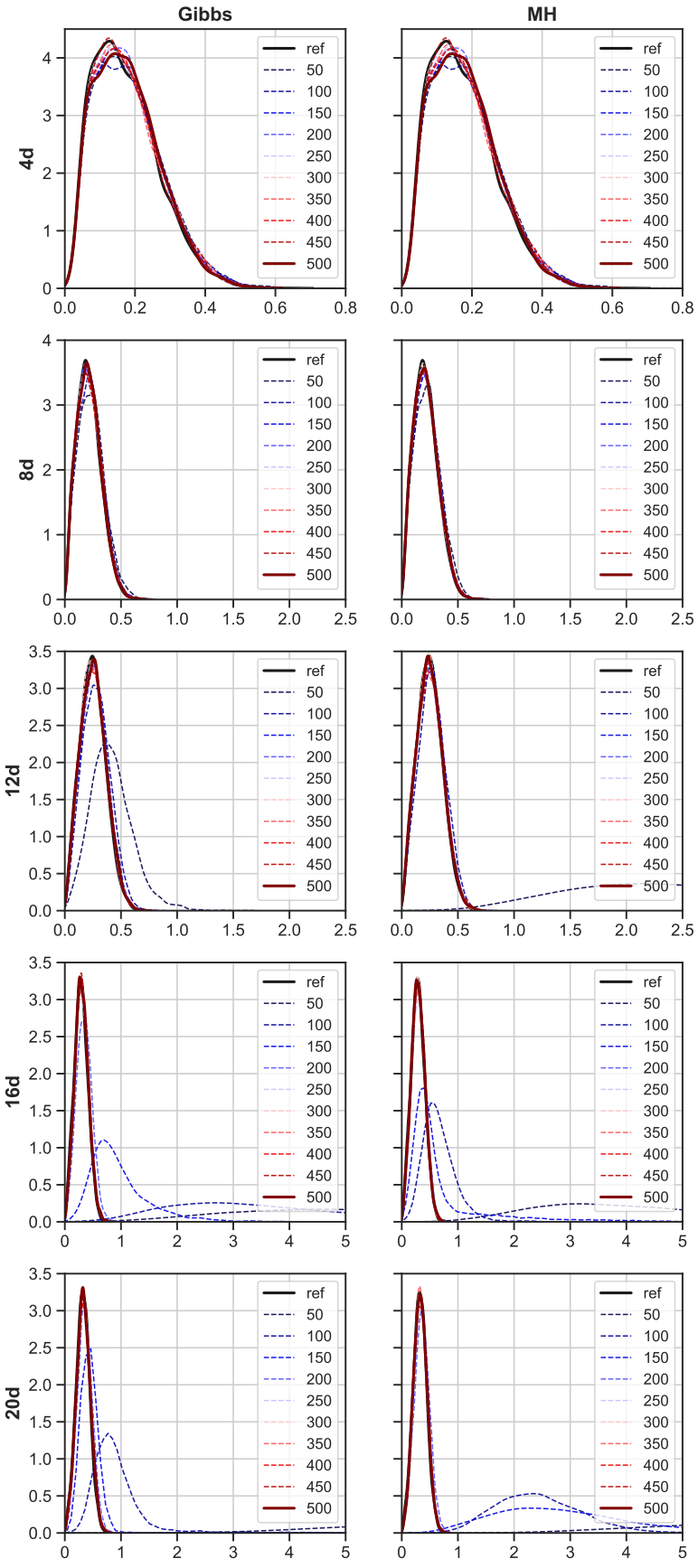

Figure 5.7: Semiparametric model (2x2): Posterior predictive discrepancy distributions obtained with Gibbs and MH. ref corresponds to the posterior predictive discrepancy distribution obtained with the ground truth posterior distribution.

According to Figure 5.6, the posterior predictive discrepancy distribution can indeed be used to identify obviously poor posterior approximations. The shape of the distribution obtained with SSE becomes increasingly dissimilar to that of the ground truth. However, a similar analysis of the distribution associated with FAdd(SA) seems to suggest that the posterior fit is significantly better than that of SSE. Yet, both KL and MMD seemed to disagree. On the other hand, the distributions obtained with both Gibbs and MH are able to match almost exactly the ground truth distribution[18]. In general, note that if the ground truth is available, the information conveyed by PPD can be summarized. For instance, Figure 5.8 shows the KL divergence between the ground truth PPD and an approximation. The analysis is similar to that of PPDME and PPDVE.

Finally, a brief note regarding the acquisition rule. As discussed at the start of this section, the value of η_t^2 was originally set too large. The fully additive method was particularly susceptible to this weight, whereas even with a poorly calibrated weight Gibbs and MH were able to avoid the boundary issue and to outperform any other method. Due to time constraints, we were unable to run new experiments, raising the question whether the new rule could have improved the performance of these methods.

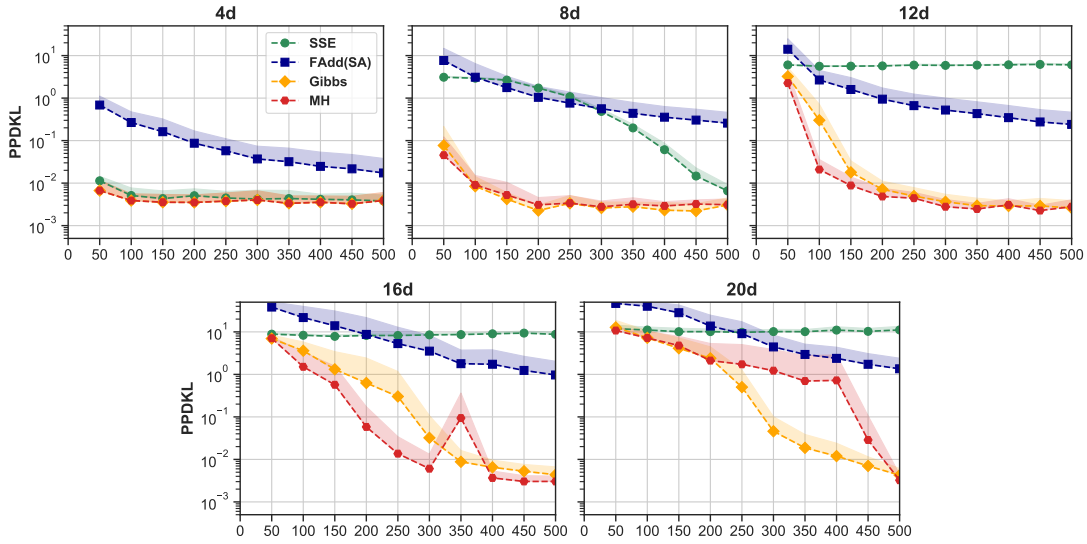

Figure 5.8: Semiparametric model (2x2): Kullback-Leibler divergence between the kernel density estimates of the ground truth PPD and an approximation.

[18]It can be shown that if the discrepancy function is a quadratic function (without added noise) and the true posterior is the corresponding Laplace approximation, the posterior predictive discrepancy distribution is given by a gamma distribution. However, in this case the true posterior has lighter tails (Figure B.2).

5.5 Experiment 2: Semiparametric Model (4x4)

In the previous experiment, we have provided an ample analysis of the results obtained for the semiparametric model with 2×2 group matrices, i.e. two dimensions per group. The reason was as much to discuss the general behavior of the different methods as it was to demonstrate that the different performance measures can effectively be used to provide a more complete analysis. Importantly, we have seen that a PPD analysis can help identifying poor approximations, but relying on such procedure to identify good approximations seems to have its shortcomings. Is is thus important to also take into account complementary performance measures. In this experiment, the analysis provided is admittedly more terse, focusing only on a subset of measures and pointing to key results. Furthermore, since there is no need to distinguish between different versions of FAdd, we refer to FAdd(SA) simply as FAdd. The problems we consider have 8, 12, 16 and 20 dimensions. NearB is not shown because the boundary issue does not occur.

Figure 5.9 shows that, when compared to the previous experiment, Gibbs and MH have greater difficulty in finding the minimum in high dimensions. After 500 acquisitions and $n = 20$, the location error is slightly above 10^{-1}, whereas in the previous case it was approximately 4×10^{-2}. Similarly, approximating the region of small discrepancies seems more difficult – the discrepancy error is about 10 times larger. Gibbs and MH still outperform the other methods, but the differences become less marked. In addition, being able to find an additive structure that resembles that of the ground truth is important in terms of approximating the discrepancy function in an efficient manner. In fact, uncovering the ground truth consistently leads to a discrepancy error of 10^{-2}. In some circumstances ($n \geq 16$) and when compared to MH, Gibbs is able to achieve a smaller discrepancy error, despite a lower F-score. Still, the error differences between both methods do not seem significant in general.

On the other hand, the measures depicted in Figure 5.10 aim to assess the approximation to the posterior. Since the discrepancy error is also related to the quality of the approximation, it is again shown for convenience. In order to determine MMD and PPDKL, it is necessary to sample from the posterior. The corresponding MCMC diagnostics are shown in Figure C.1 (Appendix C). The ESS of SSE was notably smaller than the ESS of other methods. In terms of posterior approximation, DiscError, KL and MMD convey similar information. No clear differences between SSE, Gibbs and MH

in the 8-dimensional problem. However, as the number of dimensions increases, SSE becomes unable to provide a good approximation. MH performed better than Gibbs in the 12-dimensional problem, but higher-dimensional problems suggest that one is not necessarily better than the other.

Finally, PPDKL is able to detect poor approximations. For instance, in this case, poor approximations seem to be those that yield a value greater than 1. However, unless the values are very small (i.e. around 10^{-2}), it seems unable to provide reliable information regarding the quality of the approximations. For instance, in the 16-dimensional problem, all other measures indicate that the approximation in FAdd is of inferior quality to that of MH. Yet, according to PPDKL, the approximation provided by the former seems to be marginally better.

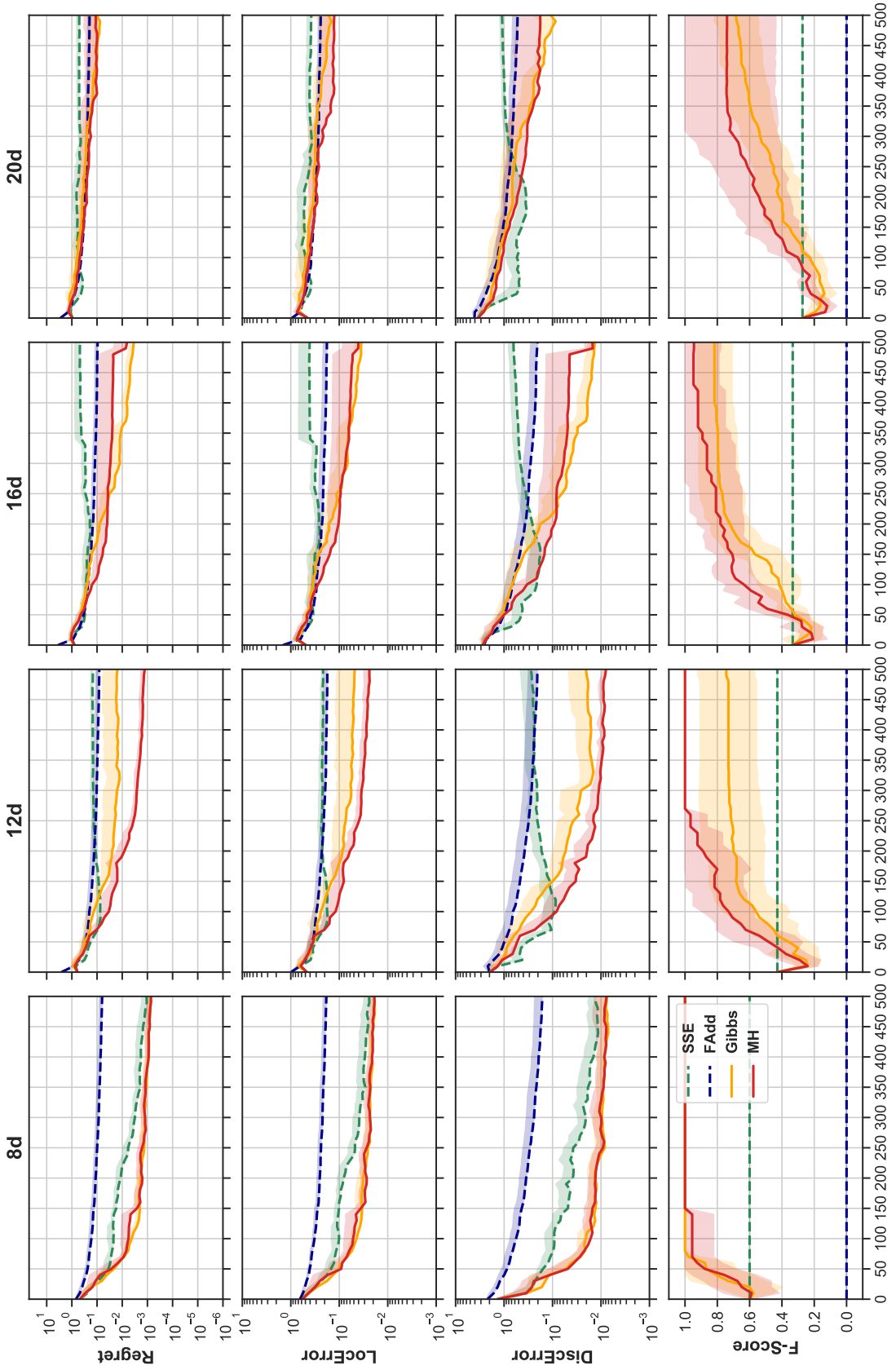

Figure 5.9: Semiparametric model (4x4): Timeplots of instantaneous regret, location error, discrepancy error and F-score. Values averaged over 10 runs, one standard error band.

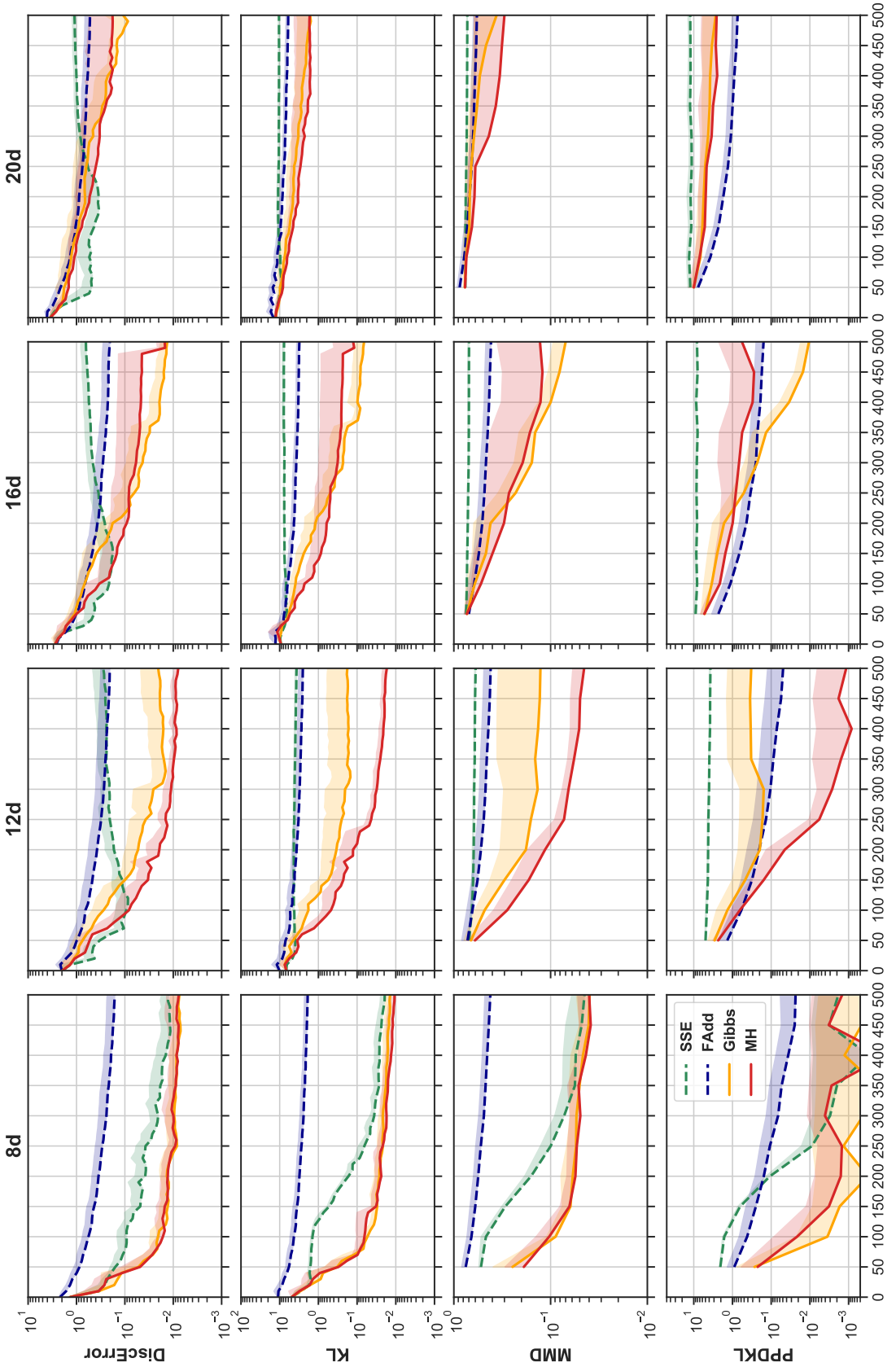

Figure 5.10: Semiparametric model (4x4): Timeplots of discrepancy error, KL between Laplace approximations, maximum mean discrepancy and posterior predictive discrepancy KL (PPDKL). Values averaged over 10 runs, one standard error band.

5.6 Experiment 3: Parametric Model (2x2)

Thus far, we have presented and discussed some results obtained with the semiparametric model. In this experiment, we revisit the case where each group contains two dimensions[19], but under the parametric model described in Section 5.3. Recall that hyperparameter optimization is now required and, since we adopt a penalized maximum likelihood approach, there is the need to define the regularization terms in the form of prior distributions. Recalling that the mean function is given by:

$$m(\boldsymbol{\theta}) = \sum_j a_j \theta_j^2, b_j \theta_j + c, \tag{5.17}$$

we define the following priors for each coefficient:

$$\pi(a_j) = \text{Lognormal}(a_j; 0, 1), \qquad j = 1, ..., n \tag{5.18}$$

$$\pi(b_j) = \text{StudentT}(b_j; 0, 1, 3), \qquad j = 1, ..., n \tag{5.19}$$

$$\pi(c) = \text{StudentT}(c; 0, 1, 3), \tag{5.20}$$

where $\text{StudentT}(0,1,3)$ is a standard t-distribution with 3 degrees of freedom, which is one of the priors recommended in Stan [16][20]. In addition, the lengthscale prior relies on the observation that values greater than the distance between the upper and lower bound of the search space (per dimension) are never observed. Hence, in order to mitigate this non-identifiability, the prior should place small mass to values above this limit, $\zeta_u = 6$. Similarly, negligible mass should be placed on values that are too small so as to avoid perfect interpolation, $\zeta_l = 6/100$. Given these conditions, the lengthscale prior is given by[21]:

$$\pi(\lambda) = \text{InvGamma}(\lambda; 11, 31). \tag{5.21}$$

Regarding the signal variance, we use

$$\pi(\sigma_f^2) = \text{Gamma}(\sigma_f^2; 1, 0.01). \tag{5.22}$$

The shape is similar to that of an exponential distribution, so that the prior mode is 0. Importantly, this prior is used in all methods, but not in the fully additive method[22].

[19] Due to time and computing constraints, it was not possible to also test the 4×4 case.

[20] Stan is not used in this work, but the reference manual contains useful practical information on how to set regularization terms.

[21] In order to find the corresponding parameters of the distribution, we used a normal approximation to the tails and solved the resulting equations using SymPy [60].

[22] We tested several signal variance priors in FAdd, including the above, but the corresponding results are not shown because they were obtained using the original acquisition rule.

Instead, the signal variance prior in FAdd is given by:

$$\pi_{FAdd}(\sigma_f^2) = \text{HalfStudentT}(\sigma_f^2; 1, 3), \tag{5.23}$$

i.e. a half Student-t distribution with scale 1 and 3 degrees of freedom.

As in Section 5.4, the problems we consider have 4, 8, 12, 16 and 20 dimensions. Importantly, the reported values have been averaged over 10 runs, except for FAdd. FAdd uses the new acquisition rule (new η_t^2) and it was only possible to collect results from 5 independent runs. The following analysis is as terse as that of the previous experiment, emphasizing key differences to that of the first experiment, while focusing only on a subset of performance measures.

Figure 5.11 shows the metrics related to Bayesian optimization and structure discovery, i.e. instantaneous regret, location error, near boundary condition and F-score. One of the key observations is that hyperparameter optimization can be problematic in Bayesian optimization, even more so when high dimensional. Indeed, in the 16- and 20-dimensional test cases, Gibbs and MH now exhibit a probability that can be as high as 50% (75% for MH) of acquiring data near the boundary during the first 50 acquisitions (plus the initial 20) and, in some runs, this issue still persists after acquiring 100 observations. However, based on the differences between FAdd(S) and FAdd(SA) of experiment 1, it is likely that the problem can be solved, or at least mitigated, by adopting the new acquisition rule. Unfortunately, the need to optimize the mean function coefficients might still cause unwanted problems. Nonetheless, it is interesting to note how these methods, even with a poorly calibrated η_t^2, can be more robust than FAdd. Given that the boundary issue also occurs, albeit to a lesser extent, in low-dimensional test cases, the main problem of FAdd seems to be that of functional form misspecification. Even with relatively strong regularization of signal variances, the method still occasionally focuses on pure exploration in both low- and high-dimensional problems, suggesting that, when compared to the coefficients of the mean function, the optimized variances in some dimensions might be large.

In terms of minimization, SSE exhibits similar behavior to that of Gibbs and MH in test cases up to 8 dimensions and to a less extent in 12 dimensions. Again, in higher-dimensional problems, the only methods that are able to perform relatively well are Gibbs and MH. In this regard, MH seems to perform slightly better than Gibbs. It is interesting to note that a sharp increase in F-score translates to a sudden drop in location error. Therefore, structure discovery seems to play an important role in

optimization and, by extension, inference, as we discuss next.

Figure 5.12 shows a subset of the measures that assess the quality of the posterior approximation. In particular, the KL between Laplace approximations, MMD and PPDKL are shown. Posterior sampling diagnostics are depicted in Figure D.1 (Appendix D). Before proceeding with the analysis of the posterior approximations, it is interesting to note that not only ESS is in general smaller in this experiment than that of experiment 1, it also manifests higher variability and larger fluctuations. In turn, this suggests again that hyperparameter optimization is the culprit. Indeed, SSE seems particularly susceptible, exhibiting exceptionally small ESS to the point that it can invalidate the respective MMD and PPDKL in the 16- and 20-dimensional test cases. In fact, based on the comparison between the ESS of SSE and SSE(S) of experiment 1 (Figure B.1), it is likely that the signal variance in SSE (of experiment 3) is being set to a value that is too large, despite regularization. Still on this note, it should also be pointed that the PSRF of FAdd is unusually high, indicating that the respective chain has not converged. A final interesting observation is that, in the high-dimensional cases, the ESS of Gibbs and MH is at first small and gradually becomes larger, suggesting that it is particularly difficult to sample when the nonparametric BOLFI approximation is crude. This in turn poses a question related to the effectiveness of sampling-based measures to assess the BOLFI approximation. Although possibly biased, measures that rely on further model-based approximations (e.g. Laplace approximation) do not suffer from this problem.

Regarding the posterior approximation, SSE performs as well as Gibbs and MH in the 4- and 8-dimensional test cases, but, in higher-dimensional cases, only Gibbs and MH exhibit good performance. In fact, in this experiment, MH tends to yield a more efficient estimator. However, to ascertain whether this statement holds more generally, additional experiments involving the parametric model would be required.

Finally, note that PPDKL seems able to distinguish between poor and good approximations, by applying the rule described in the previous section. In particular, crude approximations yield a value greater than 1, whereas a value of 10^{-2} suggests an accurate approximation. Intermediate values should be interpreted with care.

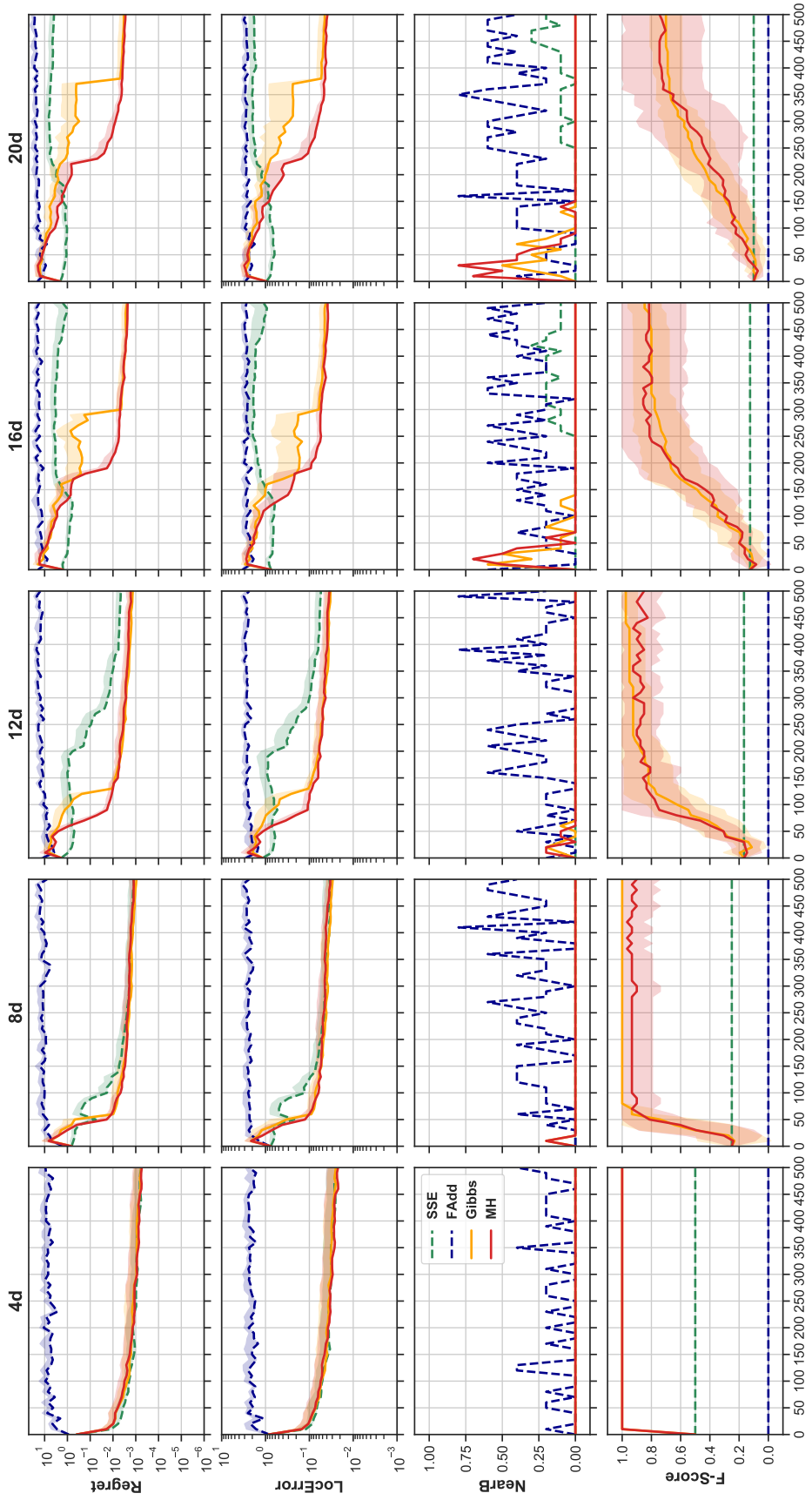

Figure 5.11: Parametric model (2x2): Timeplots of instantaneous regret, location error, near boundary condition and F-score. Values averaged over 10 runs (5 in FAdd), one standard error band.

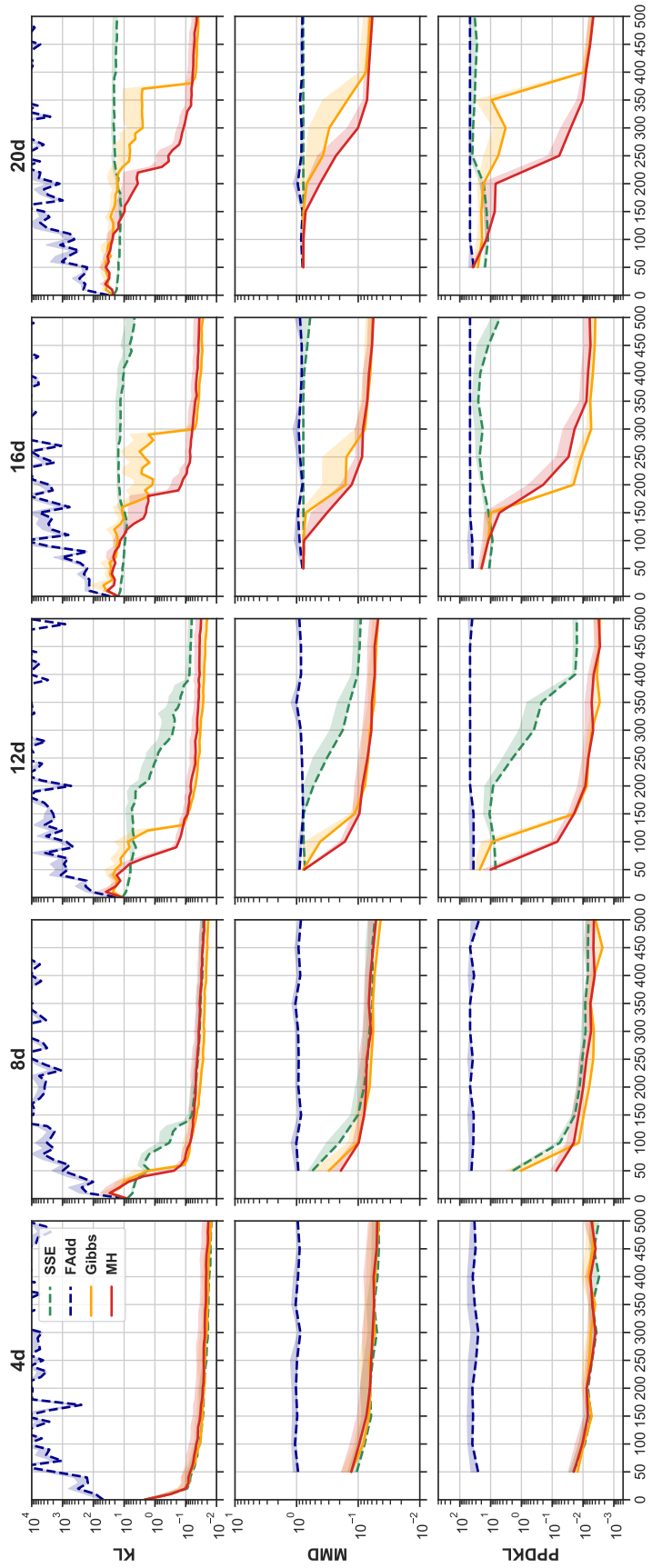

Figure 5.12: Parametric model (2x2): Timeplots of KL between Laplace approximations, maximum mean discrepancy and posterior predictive discrepancy KL (PPDKL). Values averaged over 10 runs (5 in FAdd), one standard error band.

Chapter 6

Concluding Remarks

6.1 Summary and Conclusion

In this work, we have studied the problem of high-dimensional inference in simulator-based models, i.e. models whose data generating process is defined in terms of a parametrized stochastic program. In this setting, the first difficulty lies in the fact that to draw conclusions from observed data, probabilistic inference methods, and in particular Bayesian inference methods, rely on the availability of the likelihood function, a function that measures the plausibility of a given input to generate data that resembles the observed data. However, this function is only implicitly defined in these models, as opposed to standard statistical models where its functional form is readily available. By exploring the connections between the two types of models, we have seen that the problem of inference in simulator-based models, also known as likelihood-free inference (LFI), can be reduced to conditional density estimation of a surrogate likelihood function, which is in general an approximation to the implicit likelihood function. We then focused on methods that attempt to estimate the surrogate likelihood function in an efficient manner, partly motivated by the fact that simulator-based models with a large number of input parameters tend to be computationally expensive. Efficiency in this case is associated with the estimation of high density regions of the surrogate likelihood function, which in turn led us to the Bayesian optimization for likelihood-free inference (BOLFI) framework. High-dimensional likelihood-free inference becomes thus a task of high-dimensional Bayesian optimization.

High-dimensional Bayesian optimization is a difficult task in itself, posing both com-

putational and statistical problems. However, recent advances have been able to push this field forward. A possible strategy has been to assume that the function to be optimized has an additive structure which, in turn, can be learned via Markov chain Monte Carlo (MCMC). In this context, we focused on two competing methods, one based on Metropolis-Hastings (MH) and another based on Gibbs sampling. On the other hand, the baselines were methods that assumed a specific fixed structure, namely fully additive (FAdd) and fully dependent structures. The latter in particular is the default choice in Bayesian optimization, which consists in the use of a standard squared exponential (SSE) kernel. The challenge at this point was then how to benchmark these methods. High-dimensional likelihood-free inference is a problem that has been posed recently, without *de facto* standards in terms of models and performance measures. Consequently, part of this work consisted in the investigation, design and implementation of evaluation metrics that scaled well to high-dimensional problems. Similarly, in order to test the methods in a number of different conditions, we designed two scalable models based on the observation that simulator-based models can be interpreted as stochastic functions. In particular, scalar-valued stochastic functions which, when given a parameter configuration, an auxiliary model and a number of adjustable hyperparameters, output a noisy estimate of the likelihood, or a related quantity (discrepancy between observed and generated data). At this point, likelihood-free inference has effectively been reduced to a problem of Bayesian optimization whose objective function is stochastic.

Regarding the main findings, we have observed that MH and Gibbs are generally successful in learning the hidden additive structure. In all experiments we conducted, structure discovery played an important role in optimization and, by extension, inference. Finding the region near the minimum (small discrepancies) appears to be the major difficulty in high-dimensional problems. For that reason, many of the recent advances in high-dimensional Bayesian optimization, including high-dimensional regression, seem to be directly applicable to high-dimensional likelihood-free inference. A well-calibrated acquisition rule and the ability to learn a flexible, yet robust regression model are particularly important. In this context, we found the proposed near boundary condition to be an invaluable diagnostic tool. Explicitly evaluating the mismatch between the expected minimum and the ground truth minimum also provides valuable information, perhaps even more so than instantaneous regret. In addition, our experiments have shown that hyperparameter optimization should be set with care, even when

adopting a penalized maximum likelihood strategy. MH and Gibbs were fairly robust, but SSE with optimized hyperparameters not only performed significantly worse in high-dimensional problems, but also yielded posteriors that were particularly difficult to sample from. Although costly, a Bayesian approach may be preferable.

In terms of assessing the quality of posterior approximations, we found the Kullback-Leibler (KL) divergence between Laplace approximations to be a computationally efficient performance measure. Sampling from the posterior using the BOLFI likelihood approximation can be computationally intensive, so it seems important to investigate alternative evaluation metrics that do not necessarily rely on sampling, especially in circumstances where performance needs to be continually monitored. For instance, we were able to monitor the performance using KL at every 10 acquisitions, but performance measures that required sampling were only performed at every 50 acquisitions. Furthermore, KL, maximum mean discrepancy (MMD) and discrepancy error were generally in agreement, suggesting that the information was accurate. We have also explored performance measures based on the posterior predictive discrepancy distribution and found that it is useful in the identification of poor approximations, but it should be interpreted with care otherwise.

Finally, it should be noted that SSE seemed effective in low-dimensional problems, achieving a similar performance to that of Gibbs or MH. However, if sampling from the simulator is indeed computationally expensive, then it is also justifiable to use the structure-discovery methods. In high-dimensional problems, Gibbs and MH are clearly superior. Still on the same note, we were in general unable to detect significant differences in performance between MH and Gibbs. MH was to a certain extent more efficient than Gibbs in the experiment involving hyperparameter optimization, but additional experiments would be required in order to ascertain whether this would hold more generally.

6.2 Future Work

In this work, we have explored an additive structure with non-overlapping groups, but recent work has also proposed the use of overlapping groups[1]. The assumption of a non-overlapping decomposition allows to simplify the optimization problem to a great

[1] Roughly based on my proposal.

extent, allowing terms to be optimized separately. However, it can be argued that it may be too restrictive in practical scenarios where the objective function is not additive, which in this work we designed it to be. The MCMC algorithms we used can easily be extended to this more general case. The problem lies thus in the optimization of the acquisition function. One approach is to use a graph to represent dependencies and then optimization can be performed by message passing [5]. In this context, different strategies have been proposed. For instance, in [83], the authors convert the graph into a tree which then allows efficient message passing, whereas, in [41], the authors explore loopy message passing in a decentralized scenario.

We have used Gibbs and MH to learn plausible partitions, but importantly the objective is not to estimate the corresponding posterior distribution. In this case, the MCMC algorithms can be treated as randomized search algorithms and, in principle, other algorithms can be explored. Similarly, other structure-learning approaches can be adopted. For instance, in [32], the authors map a highly structured space of molecules to a latent Euclidean space, where Bayesian optimization can be performed. It is then possible to map the points back to the original space, recovering a specific molecule with certain desirable properties. The same idea can be used to learn structure and type of kernels in Bayesian optimization. Recent work follows this approach [57].

At this point, we have discussed at length the boundary issue and how high-dimensional Bayesian optimization is particularly susceptible to this problem. We mentioned two possible strategies, one based on the specification of a quadratic mean function, and other on warping the search space [93]. However, an alternative, based on the idea of warping the search space, is to consider a different kernel type, in particular a kernel that is non-stationary. In [67], the authors propose a cylindrical kernel and show that it can lead to better performance than stationary kernels, including the additive structure that we use in this work. Interestingly, they also show that a Matérn kernel can outperform the additive squared exponential kernels. These results in turn suggest that it may not be necessary to specify an additive structure to obtain good performance in high-dimensional Bayesian optimization and, by extension, high-dimensional likelihood-free inference.

In BOLFI, the relationship between the (stochastic) scalar-valued discrepancy function and the parameters is modeled. However, it seems natural to wonder whether also introducing the information provided by the vector of summary statistics in a multi-output Gaussian process can alleviate some of the problems in high-dimensional

likelihood-free inference.

Finally, other extensions to this work have already been hinted in previous sections. Some extensions include: more realistic models of the discrepancy function, heteroskedastic and non-Gaussian; models specified by copula and marginal distributions; replacing hyperparameter learning via penalized maximum likelihood by the corresponding Bayesian approach; better acquisition rules, specifically those designed for likelihood-free inference [45]; extending the performance measure based on Laplace approximations.

Appendix A

Derivatives of Model-Based Nonparametric Likelihood

In this section, we derive the closed-form expressions for the gradient and Hessian of the (log) model-based nonparametric likelihood approximation, i.e.

$$f_{nBOLFI}(\boldsymbol{s}_o \mid \boldsymbol{\theta}) = F_{\mathcal{N}}\left(\frac{\varepsilon - \mu_t(\boldsymbol{\theta})}{\sqrt{v_t(\boldsymbol{\theta}) + \sigma_n^2}}\right). \tag{A.1}$$

First, let $\boldsymbol{A} = [\boldsymbol{K}_t + \sigma_n^2 \boldsymbol{I}]^{-1}$, the derivatives of the posterior predictive mean and variance are given by

$$\frac{\partial \mu_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial m(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + (\mathbf{g}_t - \mathbf{m}_t)^\intercal \boldsymbol{A} \frac{\partial \mathbf{k}_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{A.2}$$

$$\frac{\partial v_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial k(\boldsymbol{\theta}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} - 2\mathbf{k}_t(\boldsymbol{\theta})^\intercal \boldsymbol{A} \frac{\partial \mathbf{k}_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{A.3}$$

and the corresponding Hessian matrices by

$$\left[\frac{\partial^2 \mu_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal}\right]_{(i,\cdot)} = \left[\frac{\partial^2 m(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal}\right]_{(i,\cdot)} + (\mathbf{g}_t - \mathbf{m}_t)^\intercal \boldsymbol{A} \frac{\partial}{\partial \theta_i} \frac{\partial \mathbf{k}_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{A.4}$$

$$\left[\frac{\partial^2 v_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal}\right]_{(i,\cdot)} = \left[\frac{\partial^2 k(\boldsymbol{\theta}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal}\right]_{(i,\cdot)} - 2\left(\frac{\partial \mathbf{k}_t(\boldsymbol{\theta})^\intercal}{\partial \theta_i} \boldsymbol{A} \frac{\partial \mathbf{k}_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \mathbf{k}_t(\boldsymbol{\theta})^\intercal \boldsymbol{A} \frac{\partial}{\partial \theta_i} \frac{\partial \mathbf{k}_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right). \tag{A.5}$$

Letting $z(\boldsymbol{\theta}) = (\varepsilon - \mu_t(\boldsymbol{\theta}))/\sqrt{v_t(\boldsymbol{\theta}) + \sigma_n^2}$, the gradient of the log nonparametric BOLFI approximation corresponds to

$$\nabla_{\boldsymbol{\theta}} \log F_{\mathcal{N}}(z(\boldsymbol{\theta})) = \frac{\mathcal{N}(z(\boldsymbol{\theta}); 0, 1)}{F_{\mathcal{N}}(z(\boldsymbol{\theta}))} \nabla_{\boldsymbol{\theta}} z(\boldsymbol{\theta}) \tag{A.6}$$

$$= \frac{\mathcal{N}(z(\boldsymbol{\theta}); 0, 1)}{F_{\mathcal{N}}(z(\boldsymbol{\theta}))} \left(\frac{-\nabla_{\boldsymbol{\theta}} \mu_t(\boldsymbol{\theta})}{\sqrt{v_t(\boldsymbol{\theta}) + \sigma_n^2}} - \frac{1}{2} \frac{\varepsilon - \mu_t(\boldsymbol{\theta})}{(v_t(\boldsymbol{\theta}) + \sigma_n^2)^{3/2}} \nabla_{\boldsymbol{\theta}} v_t(\boldsymbol{\theta}) \right), \tag{A.7}$$

and the respective Hessian is given by

$$\boldsymbol{H}_{nBOLFI}(\boldsymbol{\theta}) = -\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal} \log F(z(\boldsymbol{\theta})) \tag{A.8}$$

$$= \frac{\mathcal{N}(z(\boldsymbol{\theta}); 0, 1)}{F(z(\boldsymbol{\theta}))} \left[\left(z(\boldsymbol{\theta}) + \frac{\mathcal{N}(z(\boldsymbol{\theta}); 0, 1)}{F(z(\boldsymbol{\theta}))} \right) \nabla_{\boldsymbol{\theta}} z(\boldsymbol{\theta}) \otimes \nabla_{\boldsymbol{\theta}} z(\boldsymbol{\theta}) - \frac{\partial^2 z(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal} \right], \tag{A.9}$$

where

$$\frac{\partial^2 z(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal} = -\frac{1}{\sqrt{v_t(\boldsymbol{\theta}) + \sigma_n^2}} \frac{\partial^2 \mu_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal} + \frac{1}{2} \frac{\nabla_{\boldsymbol{\theta}} v_t(\boldsymbol{\theta}) \otimes \nabla_{\boldsymbol{\theta}} \mu_t(\boldsymbol{\theta})}{(v_t(\boldsymbol{\theta}) + \sigma_n^2)^{3/2}} \tag{A.10}$$

$$- \frac{1}{2} \left(\frac{-\nabla_{\boldsymbol{\theta}} \mu_t(\boldsymbol{\theta}) \otimes \nabla_{\boldsymbol{\theta}} v_t(\boldsymbol{\theta})}{(v_t(\boldsymbol{\theta}) + \sigma_n^2)^{3/2}} - \frac{3}{2} \frac{\varepsilon - \mu_t(\boldsymbol{\theta})}{(v_t(\boldsymbol{\theta}) + \sigma_n^2)^{5/2}} \nabla_{\boldsymbol{\theta}} v_t(\boldsymbol{\theta}) \otimes \nabla_{\boldsymbol{\theta}} v_t(\boldsymbol{\theta}) \right.$$

$$\left. + \frac{\varepsilon - \mu_t(\boldsymbol{\theta})}{(v_t(\boldsymbol{\theta}) + \sigma_n^2)^{3/2}} \frac{\partial^2 v_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal} \right).$$

Appendix B

Semiparametric Model (2x2)

B.1 Posterior Sampling Diagnostics

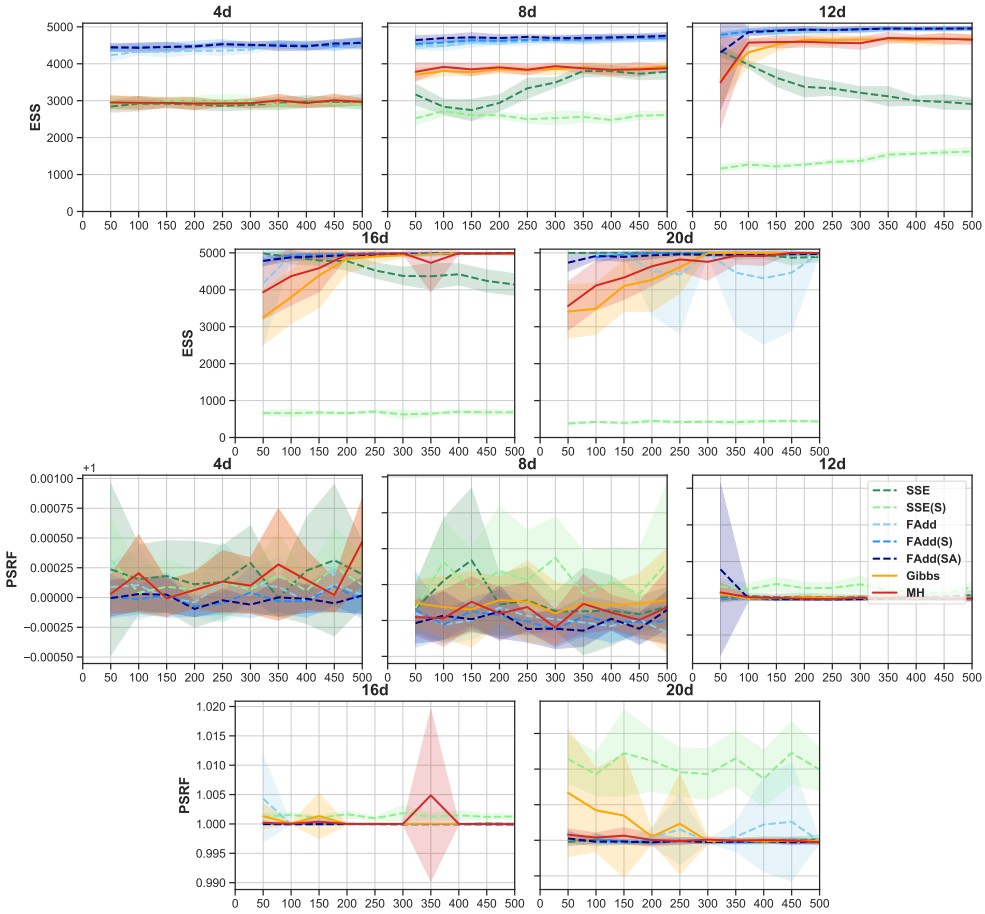

Figure B.1: Semiparametric model (2x2): Timeplots of MCMC diagnostics. Large ESS is desirable and PSRF close to 1 indicates convergence.

B.2 Posterior Predictive Discrepancy

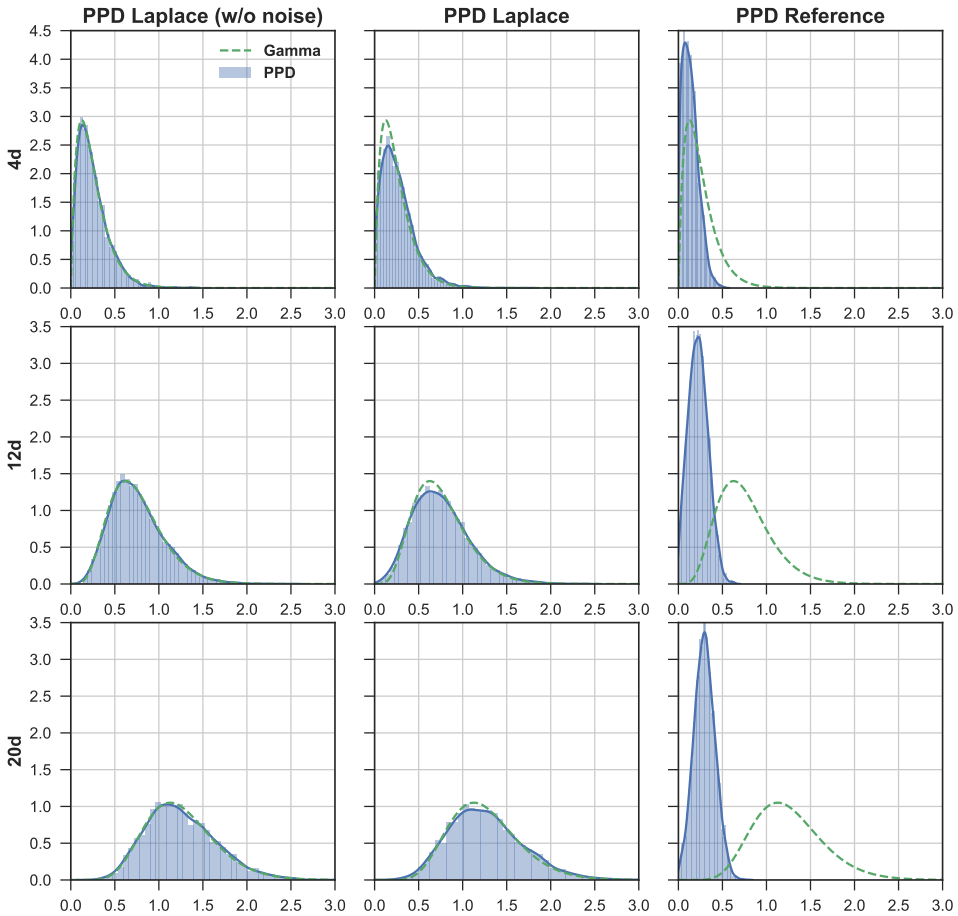

Figure B.2: Posterior predictive discrepancy (PPD) follows a gamma distribution if the discrepancy function is a quadratic function and the posterior distribution is the corresponding Laplace approximation. Once (Gaussian) observation noise is added to the discrepancy function, PPD becomes more diffuse. However, under a posterior distribution with lighter tails, the high density interval is narrower and closer to zero.

Appendix C

Semiparametric Model (4x4)

C.1 Posterior Sampling Diagnostics

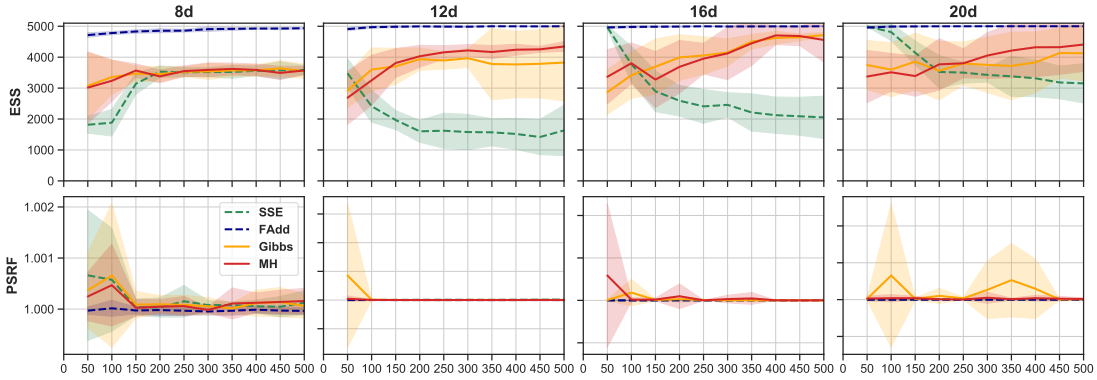

Figure C.1: Semiparametric model (4x4): Timeplots of MCMC diagnostics. Large ESS is desirable and PSRF close to 1 indicates convergence.

Appendix D

Parametric Model (2x2)

D.1 Posterior Sampling Diagnostics

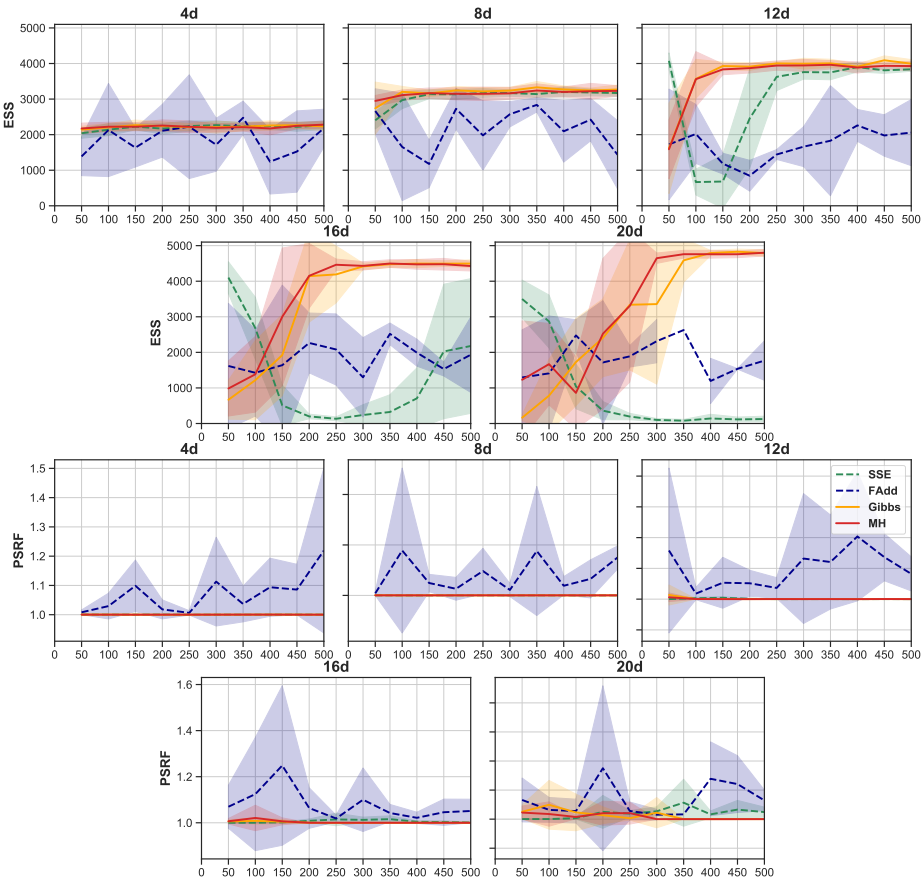

Figure D.1: Parametric model (2x2): Timeplots of MCMC diagnostics. Large ESS is desirable and PSRF close to 1 indicates convergence.

Bibliography

[1] Murray Aitkin. Posterior bayes factors. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–142, 1991.

[2] Ziwen An, Leah F. South, David J. Nott, and Christopher C. Drovandi. Accelerating bayesian synthetic likelihood with the graphical lasso. Technical report, Queensland University of Tehcnology, 2016.

[3] Ioannis Andrianakis, Ian R Vernon, Nicky McCreesh, Trevelyan J McKinley, Jeremy E Oakley, Rebecca N Nsubuga, Michael Goldstein, and Richard G White. Bayesian history matching of complex infectious disease models using emulation: a tutorial and a case study on hiv in uganda. *PLoS computational biology*, 11(1):e1003968, 2015.

[4] Christophe Andrieu, Gareth O Roberts, et al. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[5] David Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[6] Mark A Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P Robert. Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[7] Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[8] David M. Blei. Build, Compute, Critique, Repeat: Data Analysis with Latent Variable Models. *Annual Review of Statistics and Its Application*, 1(1):203–232, 2014.

[9] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 2017.

[10] Michael GB Blum and Olivier François. Non-linear regression models for approximate bayesian computation. *Statistics and Computing*, 20(1):63–73, 2010.

[11] Luke Bornn, Natesh S Pillai, Aaron Smith, and Dawn Woodard. The use of a single pseudo-sample in approximate bayesian computation. *Statistics and Computing*, 27(3):583–590, 2017.

[12] Paola Bortot, Stuart G Coles, and Scott A Sisson. Inference for stereological extremes. *Journal of the American Statistical Association*, 102(477):84–92, 2007.

[13] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[14] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[15] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

[16] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

[17] Bo Chen, Rui Castro, and Andreas Krause. Joint optimization and variable selection of high-dimensional gaussian processes. *arXiv preprint arXiv:1206.6396*, 2012.

[18] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. An adaptive sequential monte carlo method for approximate bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012.

[19] Josip Djolonga, Andreas Krause, and Volkan Cevher. High-dimensional gaussian process bandits. In *Advances in Neural Information Processing Systems*, pages 1025–1033, 2013.

[20] Christopher C Drovandi. Abc and indirect inference. *arXiv preprint arXiv:1803.01999*, 2018.

[21] Christopher C Drovandi, Clara Grazian, Kerrie Mengersen, and Christian Robert. Approximating the likelihood in approximate bayesian computation. *arXiv preprint arXiv:1803.06645*, 2018.

[22] John Duchi. Derivations for linear algebra and optimization. *Berkeley, California*, 3, 2007.

[23] David K Duvenaud, Hannes Nickisch, and Carl E Rasmussen. Additive gaussian processes. In *Advances in neural information processing systems*, pages 226–234, 2011.

[24] Babak Esmaeili. High dimensional bayesian optimization for likelihood-free inference of generative models. Master's thesis, University of Edinburgh, 2017.

[25] Y Fan and SA Sisson. ABC samplers. *arXiv preprint arXiv:1802.09650*, 2018.

[26] Paul Fearnhead and Dennis Prangle. Constructing summary statistics for approximate bayesian computation: semi-automatic approximate bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):419–474, 2012.

[27] Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. Visualization in bayesian workflow. *arXiv preprint arXiv:1709.01449*, 2017.

[28] Jacob Gardner, Chuan Guo, Kilian Weinberger, Roman Garnett, and Roger Grosse. Discovering and exploiting additive structure for bayesian optimization. In *Artificial Intelligence and Statistics*, pages 1311–1319, 2017.

[29] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2013.

[30] Andrew Gelman, Donald B Rubin, et al. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

[31] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[32] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-

Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[33] Virgilio Gómez-Rubio and Håvard Rue. Markov chain monte carlo with the integrated nested laplace approximation. *Statistics and Computing*, pages 1–19, 2017.

[34] GPy. GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.

[35] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[36] Michael U Gutmann and Jukka Corander. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *The Journal of Machine Learning Research*, 17(1):4256–4302, 2016.

[37] Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Likelihood-free inference via classification. *Statistics and Computing*, 28(2):411–425, 2018.

[38] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[39] José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang Bui, and Richard Eric Turner. Black-box α-divergence minimization. 2016.

[40] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE, 2007.

[41] Trong Nghia Hoang, Quang Minh Hoang, Ruofei Ouyang, and Kian Hsiang Low. Decentralized high-dimensional bayesian optimization with factor graphs. *arXiv preprint arXiv:1711.07033*, 2018.

[42] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *JMLR*, 14(1):1303–1347, 2013.

[43] Matthew D Hoffman, Eric Brochu, and Nando de Freitas. Portfolio allocation for bayesian optimization. In *UAI*, pages 327–336. Citeseer, 2011.

[44] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[45] Marko Järvenpää, Michael U Gutmann, Arijus Pleska, Aki Vehtari, and Pekka Marttinen. Efficient acquisition rules for model-based approximate bayesian computation. *arXiv preprint arXiv:1704.00520*, 2017.

[46] Bai Jiang. Approximate bayesian computation with kullback-leibler divergence as data discrepancy. In *International Conference on Artificial Intelligence and Statistics*, pages 1711–1721, 2018.

[47] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[48] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*, pages 295–304, 2015.

[49] Diederik P Kingma, Tim Salimans, and Max Welling. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

[50] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[51] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474, 2017.

[52] Florent Leclercq. Bayesian optimisation for likelihood-free cosmological inference. *arXiv preprint arXiv:1805.07152*, 2018.

[53] Jingjing Li, David J Nott, Yanan Fan, and Scott A Sisson. Extending approximate bayesian computation methods to high dimensions via a gaussian copula model. *Computational Statistics & Data Analysis*, 106:77–89, 2017.

[54] Yingzhen Li and Richard E Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081, 2016.

[55] Jarno Lintusaari, Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Fundamentals and recent developments in approximate bayesian computation. *Systematic biology*, 66(1):e66–e82, 2017.

[56] Jarno Lintusaari, Henri Vuollekoski, Antti Kangasrääsiö, Kusti Skytén, Marko Järvenpää, Michael Gutmann, Aki Vehtari, Jukka Corander, and Samuel Kaski. Elfi: Engine for likelihood free inference. *arXiv preprint arXiv:1708.00707*, 2017.

[57] Xiaoyu Lu, Javier Gonzalez, Zhenwen Dai, and Neil Lawrence. Structured variationally auto-encoded optimization. In *International Conference on Machine Learning*, pages 3273–3281, 2018.

[58] Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

[59] Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

[60] Aaron Meurer, Christopher P. Smith, et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3, January 2017.

[61] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

[62] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[63] Iain Murray, Ryan Prescott Adams, and David J. C. MacKay. Elliptical slice sampling. 9:541–548, 2010.

[64] David J Nott, Y Fan, L Marshall, and SA Sisson. Approximate bayesian computation and bayes linear analysis: toward high-dimensional abc. *Journal of Computational and Graphical Statistics*, 23(1):65–86, 2014.

[65] David J Nott, Victor MH Ong, Y Fan, and SA Sisson. High-dimensional ABC. *arXiv preprint arXiv:1802.09725*, 2018.

[66] Elina Numminen, Lu Cheng, Mats Gyllenberg, and Jukka Corander. Estimating

the transmission dynamics of streptococcus pneumoniae from strain prevalence data. *Biometrics*, 69(3):748–757, 2013.

[67] ChangYong Oh, Efstratios Gavves, and Max Welling. Bock: Bayesian optimization with cylindrical kernels. *arXiv preprint arXiv:1806.01619*, 2018.

[68] Victor MH Ong, David J Nott, Minh-Ngoc Tran, Scott A Sisson, and Christopher C Drovandi. Likelihood-free inference in high dimensions with synthetic likelihood. Technical report, Queensland University of Tehcnology, 2017.

[69] George Papamakarios. Sequential neural likelihood. `https://github.com/gpapamak/snl`, 2018.

[70] Dennis Prangle et al. Adapting the abc distance function. *Bayesian Analysis*, 12(1):289–309, 2017.

[71] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[72] Leah F Price, Christopher C Drovandi, Anthony Lee, and David J Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11, 2018.

[73] Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, 1999.

[74] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[75] Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. In *Advances in Neural Information Processing Systems*, pages 496–504, 2016.

[76] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2005.

[77] Oliver Ratmann, Christophe Andrieu, Carsten Wiuf, and Sylvia Richardson. Model criticism based on likelihood-free inference, with an application to pro-

tein network evolution. *Proceedings of the National Academy of Sciences*, 106(26):10576–10581, 2009.

[78] Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[79] Christian Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer, 2005.

[80] Christian P Robert, Víctor Elvira, Nick Tawn, and Changye Wu. Accelerating MCMC algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2018.

[81] Guilherme Rodrigues. *New methods for infinite and high-dimensional approximate Bayesian computation*. PhD thesis, UNSW Sydney, 2017.

[82] Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*, pages 6928–6937, 2017.

[83] Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional bayesian optimization via additive models with overlapping groups. *arXiv preprint arXiv:1802.07028*, 2018.

[84] Gian-Carlo Rota. The number of partitions of a set. *The American Mathematical Monthly*, 71(5):498–504, 1964.

[85] Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392, 2009.

[86] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1218–1226, 2015.

[87] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[88] 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.

[89] SA Sisson, Y Fan, and MA Beaumont. Overview of approximate bayesian computation. *arXiv preprint arXiv:1802.09720*, 2018.

[90] M Sklar. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231, 1959.

[91] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[92] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[93] Kevin Jordan Swersky. *Improving Bayesian Optimization for Machine Learning using Expert Priors*. PhD thesis, 2017.

[94] Mark M Tanaka, Andrew R Francis, Fabio Luciani, and SA Sisson. Using approximate bayesian computation to estimate tuberculosis transmission parameters from genotype data. *Genetics*, 173(3):1511–1520, 2006.

[95] Michalis K Titsias. Learning model reparametrizations: Implicit variational inference by fitting mcmc distributions. *arXiv preprint arXiv:1708.01529*, 2017.

[96] Fei Wang, Tanveer Syeda-Mahmood, Baba C Vemuri, David Beymer, and Anand Rangarajan. Closed-form jensen-renyi divergence for mixture of gaussians and applications to group-wise shape registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 648–655. Springer, 2009.

[97] Zi Wang. Structural kernel learning for HDBBO. `https://github.com/zi-w/Structural-Kernel-Learning-for-HDBBO`, 2017.

[98] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched high-dimensional bayesian optimization via structural kernel learning. *arXiv preprint arXiv:1703.01973*, 2017.

[99] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, Nando De Freitas,

et al. Bayesian optimization in high dimensions via random embeddings. In *IJCAI*, pages 1778–1784, 2013.

[100] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference (Springer Texts in Statistics)*. Springer, 2004.

[101] Simon N Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102, 2010.

[102] Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in variational inference. *arXiv preprint arXiv:1711.05597*, 2017.