Likelihood-free MCMC with Approximate Likelihood Ratios

Joeri Hermans
joeri.hermans@doct.uliege.be
University of Liège, Belgium

Volodimir Begy
volodimir.begy@cern.ch
University of Vienna, Austria
CERN, Switzerland

Gilles Louppe
g.louppe@uliege.be
University of Liège, Belgium

Abstract

We propose a novel approach for posterior sampling with intractable likelihoods. This is an increasingly important problem in scientific applications where models are implemented as sophisticated computer simulations. As a result, tractable densities are not available, which forces practitioners to rely on approximations during inference. We address the intractability of densities by training a parameterized classifier whose output is used to approximate likelihood ratios between arbitrary model parameters. In turn, we are able to draw posterior samples by plugging this approximator into common Markov chain Monte Carlo samplers such as Metropolis-Hastings and Hamiltonian Monte Carlo. We demonstrate the proposed technique by fitting the generating parameters of implicit models, ranging from a linear probabilistic model to settings in high energy physics with high-dimensional observations. Finally, we discuss several diagnostics to assess the quality of the posterior.

1 Introduction

Bayesian inference is a major statistical tool in scientific domains. Scientific use cases are generally interested in a posterior $p(\boldsymbol{\theta} \,|\, \mathbf{x})$ which relates parameters $\boldsymbol{\theta}$ of a model to observations $\mathbf{x} \in \mathbb{R}^d$. Although the Bayesian formulation is a natural fit for such settings, the implied computation is generally not. Often the evidence $p(\mathbf{x}) = \int p(\boldsymbol{\theta}) p(\mathbf{x} \,|\, \boldsymbol{\theta}) d\theta$ is intractable, making posterior inference using Bayes' rule impractical. Methods such as Markov chain Monte Carlo (MCMC) [1, 2] bypass the dependence on the evidence by numerically approximating the posterior. Typically, this involves evaluating some form of the likelihood ratio. In this work, we consider an equally common setting where the likelihood $p(\mathbf{x} \,|\, \boldsymbol{\theta})$ is also intractable, thereby calling for tractable approximations. A large body of research in the area of likelihood-free inference aims to improve the quality and the computational efficiency of such approximations.

This work proposes a novel approach to perform likelihood-free posterior inference using MCMC. Our method relies on the observation that one can train a classifier to approximate the likelihood ratio [3]. This approximate ratio can be used to compute the transition probability in common MCMC samplers such as Metropolis-Hastings [1, 2]. In the case the classifier is differentiable, one can derive the score $\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x} \,|\, \boldsymbol{\theta})$, which makes the method applicable to modern MCMC samplers such as Hamiltonian Monte Carlo [4].

Contributions We adapt the training procedure proposed by [3] to improve the quality of the approximated likelihood ratio (Section 2.1). We introduce approximate likelihood ratio MCMC and derive likelihood-free Metropolis-Hastings and Hamiltonian Monte Carlo samplers (Section 3).

2 Background

2.1 Approximate likelihood ratio tests

When comparing two hypotheses $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ under a set of observations $\mathcal{O}$, the most powerful test-statistic [5] is the likelihood ratio

$$\Lambda(\mathcal{O}; \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) = \prod_{\mathbf{x} \in \mathcal{O}} \frac{p(\mathbf{x} \,|\boldsymbol{\theta}_0)}{p(\mathbf{x} \,|\boldsymbol{\theta}_1)}. \tag{1}$$

Previous work [3] shows that it is possible to reformulate the test-statistic under a change of variables $\mathbf{s}(\mathbf{x})$ such that the likelihood ratio can be written as

$$\Lambda(\mathcal{O}; \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) = \prod_{\mathbf{x} \in \mathcal{O}} \frac{\mathbf{s}(\mathbf{x})}{1 - \mathbf{s}(\mathbf{x})}. \tag{2}$$

This observation can be used in a supervised setting to train a classifier $\mathbf{s}(\mathbf{x})$ to distinguish samples $\mathbf{x} \sim p(\mathbf{x} \mid \boldsymbol{\theta}_0)$ and $\mathbf{x} \sim p(\mathbf{x} \mid \boldsymbol{\theta}_1)$. The decision function modeled by the optimal classifier $\mathbf{s}^*(\mathbf{x})$ is

$$\mathbf{s}^*(\mathbf{x}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta}_0)}{p(\mathbf{x} \mid \boldsymbol{\theta}_0) + p(\mathbf{x} \mid \boldsymbol{\theta}_1)}. \tag{3}$$

Under this decision function, the likelihood ratio of a sample $\mathbf{x}$ can be obtained using

$$r(\mathbf{x}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) \triangleq \frac{p(\mathbf{x} \mid \boldsymbol{\theta}_0)}{p(\mathbf{x} \mid \boldsymbol{\theta}_1)} \equiv \frac{\mathbf{s}^*(\mathbf{x})}{1 - \mathbf{s}^*(\mathbf{x})}. \tag{4}$$

In some cases, one might also be interested in comparing a larger set of hypotheses. However, training $\mathbf{s}(\mathbf{x})$ for every likelihood ratio test is not practical. A solution proposed by [3] is to train a parameterized classifier $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ against an arbitrary reference hypothesis $\boldsymbol{\theta}_{\text{ref}}$. Similarly, the decision function modeled by $\mathbf{s}^*(\mathbf{x}, \boldsymbol{\theta})$ is

$$\mathbf{s}^*(\mathbf{x}, \boldsymbol{\theta}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta})}{p(\mathbf{x} \mid \boldsymbol{\theta}) + p(\mathbf{x} \mid \boldsymbol{\theta}_{\text{ref}})}. \tag{5}$$

As the reference hypothesis $\boldsymbol{\theta}_{\text{ref}}$ is known, the likelihood ratio between two hypotheses of interest $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ against $\boldsymbol{\theta}_{\text{ref}}$ is equivalent to

$$r(\mathbf{x} \mid \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) \equiv \frac{r(\mathbf{x} \mid \boldsymbol{\theta}_0, \boldsymbol{\theta}_{\text{ref}})}{r(\mathbf{x} \mid \boldsymbol{\theta}_1, \boldsymbol{\theta}_{\text{ref}})}. \tag{6}$$

In practice, the choice of a (mathematically) arbitrary $\boldsymbol{\theta}_{\text{ref}}$ has a significant effect on the approximated likelihood ratio function. This is not only attributable to modeling choices in $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$, but also to issues arising in the absence of support between $p(\mathbf{x} \mid \boldsymbol{\theta})$ and $p(\mathbf{x} \mid \boldsymbol{\theta}_{\text{ref}})$. In these regions, the classifier $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ can take on an arbitrary decision function, which results in arbitrary approximated likelihood ratios. Additionally, the use of a reference hypothesis adds a sensitive hyperparameter to the optimization algorithm which requires careful tuning. To overcome the issue with reference hypotheses, we propose training $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ to classify samples from the evidence $p(\mathbf{x})$ and the likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ such that the decision function becomes

$$\mathbf{s}^*(\mathbf{x}, \boldsymbol{\theta}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta})}{p(\mathbf{x} \mid \boldsymbol{\theta}) + p(\mathbf{x})}, \tag{7}$$

which can be targeted using Algorithm 1. Under this formulation, one can use the likelihood-to-evidence ratio

$$r_e(\mathbf{x}, \boldsymbol{\theta}) \triangleq \frac{\mathbf{s}^*(\mathbf{x}, \boldsymbol{\theta})}{1 - \mathbf{s}^*(\mathbf{x}, \boldsymbol{\theta})} \equiv \frac{p(\mathbf{x} \mid \boldsymbol{\theta})}{p(\mathbf{x})} \tag{8}$$

to obtain the log likelihood ratio

$$\log r(\mathbf{x}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) \triangleq \log r_e(\mathbf{x}, \boldsymbol{\theta}_0) - \log r_e(\mathbf{x}, \boldsymbol{\theta}_1). \tag{9}$$

Under the optimized classifier $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ we obtain $\log \hat{r}(\mathbf{x}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)$ by computing

$$\log \hat{r}_e(\mathbf{x}, \boldsymbol{\theta}_0) - \log \hat{r}_e(\mathbf{x}, \boldsymbol{\theta}_1). \tag{10}$$

Contrary to other approaches which use a classifier to directly approximate the likelihood-to-evidence ratio [6], ours by construction has the desirable property that

$$\log \hat{r}(\mathbf{x}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) = -\log \hat{r}(\mathbf{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_0). \tag{11}$$

Targeting the evidence $p(\mathbf{x})$ has several additional advantages. First, we remove the reliance on a manually specified and sensitive $\boldsymbol{\theta}_{\text{ref}}$. Second, there is support between $p(\mathbf{x})$ and $p(\mathbf{x} \mid \boldsymbol{\theta})$ $\forall\, \boldsymbol{\theta} \in \boldsymbol{\Theta}$, resulting in a proper definition of the likelihood ratio. Third, a single uniform simulation dataset can be used to train $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$, simplifying the implementation. The decision functions of classifiers trained with both approaches are shown in Figure 1. Finally, we would like to mention that the additional loss term in Algorithm 1 (line seven) is not required when having access to an infinite amount of simulations [3]. However, empirical evaluations with limited simulation budgets indicate this term vastly improves the stability and convergence of the resulting parameterized classifier.

Algorithm 1 Optimization of $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$.

Inputs:	Generative model $p(\mathbf{x} \mid \boldsymbol{\theta})$.
	Prior $p(\boldsymbol{\theta})$.
Outputs:	Parameterized classifier $\mathbf{s}_\phi(\mathbf{x}, \boldsymbol{\theta})$.
Hyperparameters:	Batch-size M.

1: **while not converged do**
2: **Sample** $\boldsymbol{\theta} \leftarrow \{\boldsymbol{\theta}_m \sim p(\boldsymbol{\theta})\}_{m=1}^{M}$
3: **Sample** $\boldsymbol{\theta}' \leftarrow \{\boldsymbol{\theta}'_m \sim p(\boldsymbol{\theta})\}_{m=1}^{M}$
4: **Simulate** $\mathbf{x} \leftarrow \{\mathbf{x}_m \sim p(\mathbf{x} \mid \boldsymbol{\theta}_m)\}_{m=1}^{M}$
5: **Simulate** $\mathbf{x}' \leftarrow \{\mathbf{x}'_m \sim p(\mathbf{x} \mid \boldsymbol{\theta}'_m)\}_{m=1}^{M}$
6: $\mathcal{L}_a \leftarrow \textbf{BCE}(\mathbf{s}_\phi(\mathbf{x}, \boldsymbol{\theta}),\ \mathbb{1}) + \textbf{BCE}(\mathbf{s}_\phi(\mathbf{x}', \boldsymbol{\theta}),\ \mathbb{0})$
7: $\mathcal{L}_b \leftarrow \textbf{BCE}(\mathbf{s}_\phi(\mathbf{x}', \boldsymbol{\theta}'),\ \mathbb{1}) + \textbf{BCE}(\mathbf{s}_\phi(\mathbf{x}, \boldsymbol{\theta}'),\ \mathbb{0})$
8: $\mathcal{L} \leftarrow \mathcal{L}_a + \mathcal{L}_b$
9: $\phi \leftarrow \textbf{OPTIMIZER}(\phi,\ \nabla_\phi \mathcal{L})$
10: **end while**
11: **return** $\boldsymbol{s}$

2.2 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) methods are generally applied when one is interested in a posterior density

$$p(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{p(\boldsymbol{\theta}) p(\mathbf{x} \mid \boldsymbol{\theta})}{p(\mathbf{x})} \tag{12}$$

for which the evidence $p(\mathbf{x})$ is typically intractable, but point-wise evaluations of the likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ are possible [1, 2, 7]. MCMC samples are drawn from $p(\boldsymbol{\theta} \mid \mathbf{x})$ by

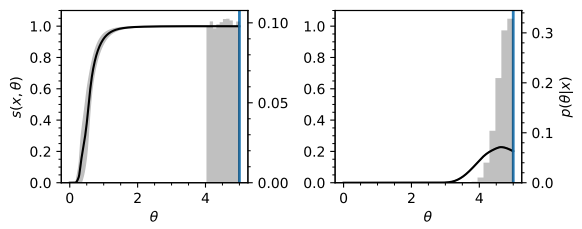


Figure 1: In both scenarios an ensemble of 10 classifiers is trained under $p(\boldsymbol{\theta}) = \mathcal{U}(-5, 5)$. We show the mean decision function of $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ and its standard deviation under an observation $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}^* = 5, 1)$. $\boldsymbol{\theta}^*$ is superimposed as a blue line. Additionally, we plot the approximate posterior (shaded area near $\boldsymbol{\theta}^*$). *(Left):* training procedure proposed by [3]. $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ is trained under $\boldsymbol{\theta}_{\text{ref}} = 0$. *(Right):* the proposed procedure. By classifying samples against the evidence, the resulting decision function has significantly less variance due to proper constraints for the optimization algorithm. This translates into $p(\boldsymbol{\theta} \mid \mathbf{x})$ putting more density on $\boldsymbol{\theta}^*$.

collecting dependent states $\boldsymbol{\theta}_{0:n}$ of a Markov chain. The mechanism for proposing the next state $\boldsymbol{\theta}'$ in the Markov chain is dependent on the algorithm at hand. However, accepting the transition $\boldsymbol{\theta}_t \to \boldsymbol{\theta}'$ is usually determined by evaluating some form of the ratio

$$\Lambda(\mathcal{O}; \boldsymbol{\theta}', \boldsymbol{\theta}_t) = \prod_{\mathbf{x} \in \mathcal{O}} \frac{p(\mathbf{x} \mid \boldsymbol{\theta}')}{p(\mathbf{x} \mid \boldsymbol{\theta}_t)}. \tag{13}$$

As a result, the normalizing constant is not required as $p(\mathbf{x})$ cancels out within the ratio

$$\frac{p(\boldsymbol{\theta}' \mid \mathbf{x})}{p(\boldsymbol{\theta} \mid \mathbf{x})} = \frac{p(\boldsymbol{\theta}') p(\mathbf{x} \mid \boldsymbol{\theta}')}{p(\boldsymbol{\theta}) p(\mathbf{x} \mid \boldsymbol{\theta})}. \tag{14}$$

3 Approximate Likelihood Ratio MCMC

In this work, we propose an approach to draw samples from a posterior using Markov chain Monte Carlo without likelihoods. We base our approach on the observation that MCMC samplers use the likelihood ratio to assess the quality of a candidate state $\boldsymbol{\theta}'$ against the current state $\boldsymbol{\theta}_t$. In many applications however, the evaluation of the likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ is intractable. Recent methods [8, 9, 10] address this by modeling an approximate density $\hat{p}(\mathbf{x} \mid \boldsymbol{\theta})$ and use $\hat{p}(\mathbf{x} \mid \boldsymbol{\theta})$ as a proxy in the likelihood ratio. Typically, this requires (complex) density estimators [11, 12, 9, 13] and expensive training procedures to properly approximate $p(\mathbf{x} \mid \boldsymbol{\theta})$. Instead, we directly model the likelihood ratio r using the technique presented in Section 2.1 such that $\hat{r}(\mathbf{x}, \boldsymbol{\theta}', \boldsymbol{\theta}_t) \approx r(\mathbf{x}, \boldsymbol{\theta}', \boldsymbol{\theta}_t)$. As a result, the optimization procedure reduces to a classification problem without making any modeling assumptions on $p(\mathbf{x} \mid \boldsymbol{\theta})$. After training, we can use the classifier $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ to transform common MCMC samplers into likelihood-free alternatives (Section 3.1 & Section 3.2). Building upon the results presented in this section, we propose a practical procedure in Appendix A which iteratively improves the approximate posterior.

3.1 Metropolis-Hastings

Adapting the Metropolis-Hastings (MH) algorithm [1, 2] is done by replacing the likelihood ratio test in the original Metropolis mechanism, making the sampler likelihood-free. We summarize the modified MH sampler in Algorithm 2 for completeness.

Algorithm 2 Likelihood-free Metropolis-Hastings

Inputs: Initial parameter $\boldsymbol{\theta}_0$.
Transition distribution $q(\boldsymbol{\theta})$.
Parameterized classifier $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$.
Observations $\mathcal{O}$.
Outputs: Markov chain $\boldsymbol{\theta}_{0:n}$
Hyperparameters: Steps n.

$t \leftarrow 0$
$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_0$
for $t < n$ **do**
  $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta} \mid \boldsymbol{\theta}_t)$
  $\lambda \leftarrow \sum_{\mathbf{x} \in \mathcal{O}} \log \hat{r}_e(\mathbf{x}, \boldsymbol{\theta}') - \sum_{\mathbf{x} \in \mathcal{O}} \log \hat{r}_e(\mathbf{x}, \boldsymbol{\theta}_t)$
  $\rho \leftarrow \min \left\{ \exp(\lambda) \frac{q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}')}{q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}_t)},\ 1 \right\}$
  $\boldsymbol{\theta}_{t+1} \leftarrow \begin{cases} \boldsymbol{\theta}' & \text{with probability } \rho \\ \boldsymbol{\theta}_t & \text{with probability } 1 - \rho \end{cases}$
  $t \leftarrow t + 1$
end for
return $\boldsymbol{\theta}_{0:n}$

3.2 Hamiltonian Monte Carlo

Hybrid or Hamiltonian Monte Carlo (HMC) [14, 4, 15] improves the sampling efficiency of vanilla MCMC algorithms by reducing the autocorrelation within the sequence of random Monte Carlo samples. Improving the sampling efficiency is especially useful in scenarios where designing a proper transition distribution is difficult. In HMC, this is achieved by modeling the density $p(\mathbf{x} \mid \boldsymbol{\theta})$ as a potential energy function

$$U(\boldsymbol{\theta}) \triangleq -\log p(\mathbf{x} \mid \boldsymbol{\theta}), \tag{15}$$

and attributing some kinetic energy

$$K(\mathbf{m}) \triangleq \frac{1}{2} \mathbf{m}^2, \tag{16}$$

with momentum $\mathbf{m} \sim p(\mathbf{m})$ to the current state $\boldsymbol{\theta}_t$. A new state $\boldsymbol{\theta}'$ can be proposed by simulating the Hamiltonian dynamics of $\boldsymbol{\theta}_t$. This is achieved by integrating $\nabla_{\boldsymbol{\theta}}\, U(\boldsymbol{\theta})$ over a fixed number of steps with initial momentum $\mathbf{m}$. Afterwards, the Metropolis mechanism uses

$$\lambda \leftarrow \exp\left(U(\boldsymbol{\theta}_t) - U(\boldsymbol{\theta}') + K(\mathbf{m}) - K(\mathbf{m}')\right) \quad (17)$$

to accept or reject the transition to the candidate $\boldsymbol{\theta}'$. The first step in making HMC likelihood-free, is by showing $U(\boldsymbol{\theta}_t) - U(\boldsymbol{\theta}')$ reduces to the log likelihood ratio,

$$\begin{aligned} U(\boldsymbol{\theta}_t) - U(\boldsymbol{\theta}') &= \log p(\mathbf{x}\,|\boldsymbol{\theta}') - \log p(\mathbf{x}\,|\boldsymbol{\theta}_t) \quad (18)\\ &= \log r(\mathbf{x}, \boldsymbol{\theta}', \boldsymbol{\theta}_t). \quad (19)\end{aligned}$$

To simulate the Hamiltonian dynamics of $\boldsymbol{\theta}_t$, we require a likelihood-free definition of $\nabla_{\boldsymbol{\theta}}\, U(\boldsymbol{\theta})$. Current likelihood-free implementations of HMC rely on finite differences to estimate this gradient [16]. Instead, we make the observation that $\nabla_{\boldsymbol{\theta}}\, U(\boldsymbol{\theta})$ can be rewritten as

$$\nabla_{\boldsymbol{\theta}}\, U(\boldsymbol{\theta}) = -\frac{\nabla_{\boldsymbol{\theta}}\, p(\mathbf{x}\,|\,\boldsymbol{\theta})}{p(\mathbf{x}\,|\,\boldsymbol{\theta})}. \quad (20)$$

This form can be recovered by the optimal classifier $\mathbf{s}^*(\mathbf{x}, \boldsymbol{\theta})$ under the assumption that the classifier is differentiable. To show this, consider

$$\nabla_{\boldsymbol{\theta}}\, U(\boldsymbol{\theta}) \equiv -\frac{\nabla_{\boldsymbol{\theta}}\, r_e(\mathbf{x}, \boldsymbol{\theta})}{r_e(\mathbf{x}, \boldsymbol{\theta})}. \quad (21)$$

By expanding the likelihood-to-evidence ratio $r_e(\mathbf{x}, \boldsymbol{\theta})$, Equation 21 reduces to

$$\begin{aligned} \nabla_{\boldsymbol{\theta}}\, U(\boldsymbol{\theta}) &\equiv -\frac{\nabla_{\boldsymbol{\theta}}\, r_e(\mathbf{x}, \boldsymbol{\theta})}{r_e(\mathbf{x}, \boldsymbol{\theta})}, \quad (22)\\ &\equiv -\frac{\frac{1}{p(\mathbf{x})}\nabla_{\boldsymbol{\theta}}\, p(\mathbf{x}\,|\,\boldsymbol{\theta})}{\frac{p(\mathbf{x}\,|\,\boldsymbol{\theta})}{p(\mathbf{x})}}, \quad (23)\\ &\equiv -\frac{\nabla_{\boldsymbol{\theta}}\, p(\mathbf{x}\,|\,\boldsymbol{\theta})}{p(\mathbf{x}\,|\,\boldsymbol{\theta})}. \quad (24)\end{aligned}$$

As an illustration, Figure 2 shows the ability of an approximate classifier $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ to produce accurate estimates of $\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}\,|\,\boldsymbol{\theta})$. In particular, the example shows the gradient of observations $\mathbf{x} \sim \mathcal{N}(0, 1)$ with respect to $\boldsymbol{\theta}$ and highlights that estimates are accurate when samples of $p(\mathbf{x}\,|\,\boldsymbol{\theta})$ have been presented to $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ during training.

Finally, having likelihood-free forms for $U(\boldsymbol{\theta}) - U(\boldsymbol{\theta}')$ and $\nabla_{\boldsymbol{\theta}}\, U(\boldsymbol{\theta})$, we can now replace these components in HMC to obtain a likelihood-free HMC sampler (LF-HMC).

Algorithm 3 Likelihood-free Hamiltonian Monte Carlo

Inputs: Initial parameter $\boldsymbol{\theta}_0$.
Momentum distribution $q(\mathbf{m})$.
Parameterized classifier $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$.
Observations $\mathcal{O}$.
Outputs: Markov chain $\boldsymbol{\theta}_{0:n}$
Hyperparameters: Steps n.
Leapfrog-integration steps l.
Leapfrog-integration stepsize η.

$t \leftarrow 0$
$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_0$
for $t < n$ **do**
  $\mathbf{m}_t \sim q(\mathbf{m})$
  $k \leftarrow 0$
  $\mathbf{m}_k \leftarrow \mathbf{m}_t$
  $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_t$
  for $k < l$ **do**
    $\mathbf{m}_k \leftarrow \mathbf{m}_k + \frac{\eta}{2|\mathcal{O}|} \sum_{\mathbf{x}\in\mathcal{O}} \frac{\nabla_{\boldsymbol{\theta}}\, s(\mathbf{x}, \boldsymbol{\theta}_k)}{s(\mathbf{x}, \boldsymbol{\theta}_k)}$
    $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k + \eta \mathbf{m}_k$
    $\mathbf{m}_k \leftarrow \mathbf{m}_k + \frac{\eta}{2|\mathcal{O}|} \sum_{\mathbf{x}\in\mathcal{O}} \frac{\nabla_{\boldsymbol{\theta}}\, s(\mathbf{x}, \boldsymbol{\theta}_k)}{s(\mathbf{x}, \boldsymbol{\theta}_k)}$
    $k \leftarrow k + 1$
  end for
  $r \leftarrow \sum_{\mathbf{x}\in\mathcal{O}} \log \hat{r}(\mathbf{x}, \boldsymbol{\theta}_k, \boldsymbol{\theta}_t) + K(\mathbf{m}_t) - K(\mathbf{m}_k)$
  $\rho \leftarrow \min(\exp(r), 1)$
  $\boldsymbol{\theta}_{t+1} \leftarrow \begin{cases} \boldsymbol{\theta}_k & \text{with probability } \rho \\ \boldsymbol{\theta}_t & \text{with probability } 1 - \rho \end{cases}$
  $t \leftarrow t + 1$
end for
return $\boldsymbol{\theta}_{0:n}$

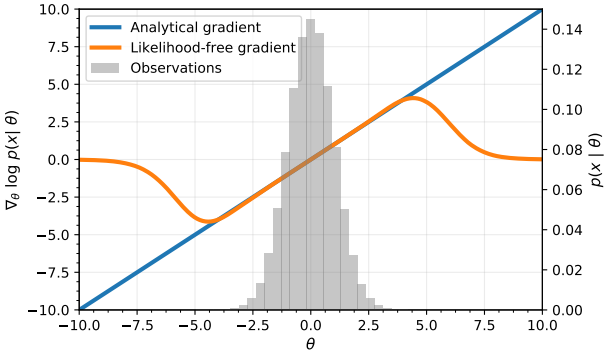
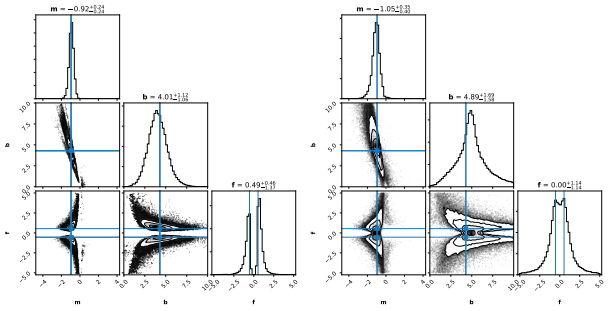
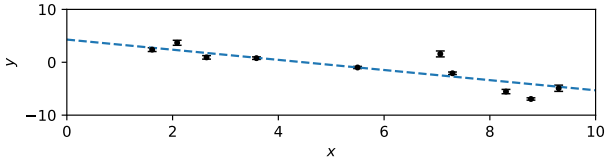


Figure 2: This figure shows the ability of $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ to produce gradient estimates of the log likelihood with respect to the analytical form. The discrepancy originates from the absence of density in these regions during training. In particular, the classifier did not observe any samples there, while the closed-form expression for the gradient attributes some density to these regions.

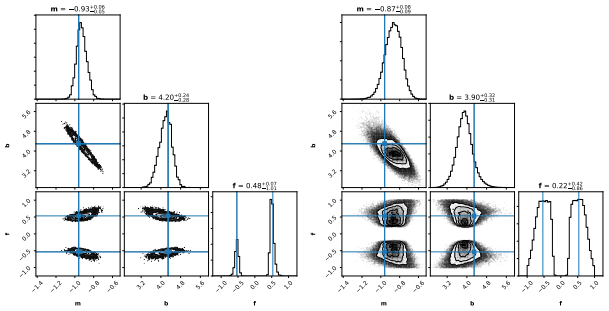


(a) LF-MH (b) ABC

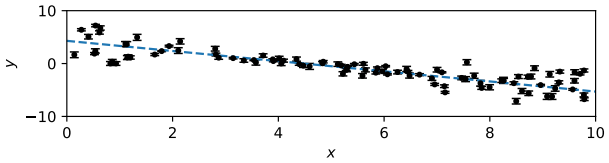


(c) Observations

Figure 3: Approximate posterior samples for 10 observations drawn from $\boldsymbol{\theta}^*$. The acceptance threshold for ABC has been set to $\epsilon = 0.01$. The dashed blue line in Subfigure 3c indicates the true slope $\mathbf{m}$ and intercept $\mathbf{b}$.



(a) LF-MH (b) ABC



(c) Observations

Figure 4: Approximate posterior samples for 100 observations drawn from $\boldsymbol{\theta}^*$. The ABC acceptance threshold has been set to $\epsilon = 0.001$. Modes for $\mathbf{f}$ are unbalanced, which is attributable to the dependence among MCMC samples.

4 Experiments

We demonstrate likelihood-free MCMC in several problem settings. We report the classifier structure, and the considered $p(\boldsymbol{\theta})$ during the training of $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ in every experiment. We do not apply the iterative technique proposed in Appendix A as the classifiers were sufficiently powerful. The posterior is approximated by $100,000$ samples. We also dedicate the computational effort to collect $100,000$ Approximate Bayesian Computation (ABC) posterior samples for benchmarking. Additionally, MCMC includes a short burnin period.

The source code to reproduce these experiments can be found in the following repository[1]. To make the proposed technique directly applicable, we wrapped the method and associated diagnostics in a Python package called HYPOTHESIS[2].

4.1 Linear regression model

In this first illustrative experiment, we are interested in inferring the posterior over the parameters of a linear probabilistic generative model[3]

$$y' = \mathbf{m} \cdot x + \mathbf{b}, \tag{25}$$

$$y = \epsilon \cdot |\mathbf{f} \cdot y'| + 0.05\epsilon, \tag{26}$$

with $\epsilon \sim \mathcal{N}(0, 1)$, and $\boldsymbol{\theta} \triangleq (\mathbf{m}, \mathbf{b}, \mathbf{f})$. Figure 3 and Figure 4 show the posterior for 10 and 100 observations respectively. For each of these plots, the same parameterized classifier is used to draw posterior samples. We would like to note two modes should be present for $\mathbf{f}$ as y depends on the absolute value of $\mathbf{f}$.

Setup We generate $1,000,000$ simulations for $\boldsymbol{\theta}$ drawn from a uniform prior $p(\boldsymbol{\theta}) = \mathcal{U}([-5, 5] \times [0, 10] \times [-5, 5])$. These simulations are used to train $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$. The architecture of $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ is a multilayer-perceptron with SELU [17] non-linearities. No regularization nor any data normalization technique is applied. The classifier is trained using Adam [18]. The summary statistic for ABC is the combination of the slope, intercept and resulting r-value of a linear regression. Finally, the Euclidean distance is computed between the summary statistics to assess the fit of drawn parameter against the observations. Observations are simulated from $\boldsymbol{\theta}^* \triangleq (-0.9594, 4.294, -0.534)$.

Results The approximate posteriors computed by the proposed method and by ABC are shown in Figures 3 and Figure 4, for 10 and 100 observations respectively. Plots suggest agreement between the two approximate posteriors, showing similar relations between parameters. In both settings, the same pre-trained parameterized classifier is used, which shows the independence between the expensive training procedure and inference. While our method requires a static number of simula-

[1] github.com/montefiore-ai/likelihood-free-mcmc
[2] github.com/montefiore-ai/hypothesis
[3] Adapted from dfm.io/emcee/current/user/line/

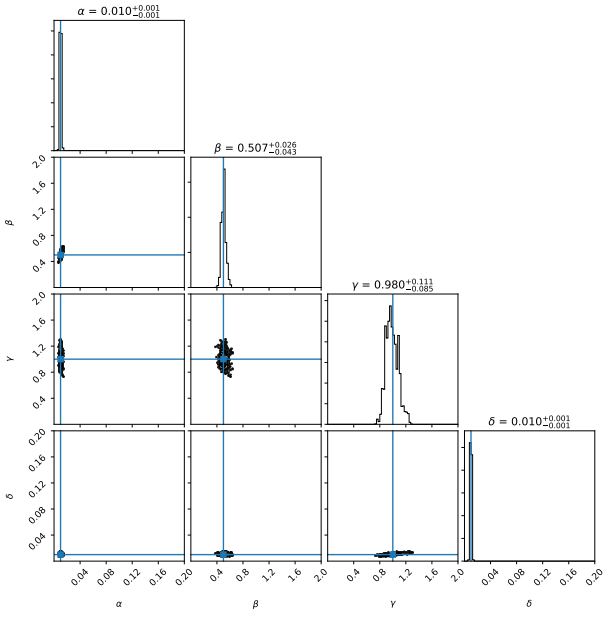
tions, rejection sampling schemes have to rely on active calls to the simulator. This approach might be inefficient for small acceptance thresholds under a specific prior. Additionally, the proposed approach can be trained on pre-simulated datasets (common in many scientific fields), thereby limiting the reliance on expensive simulator calls during training.

4.2 Lotka-Volterra population model

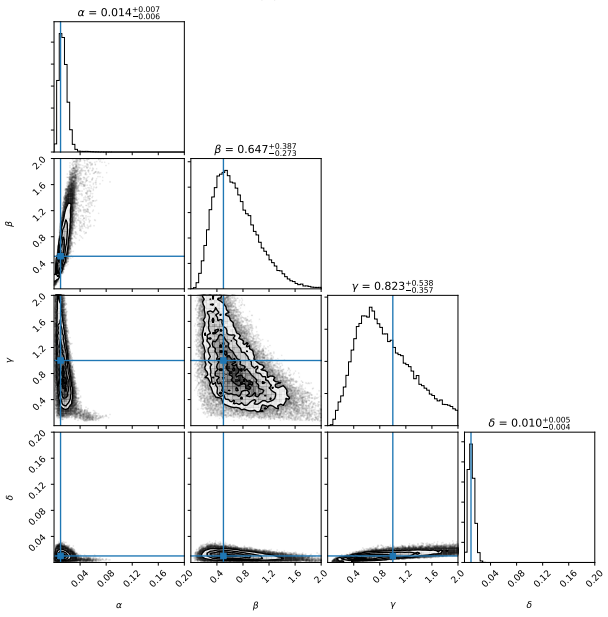
The Lotka-Volterra model [19] describes the evolution of predator and prey populations over time. The population dynamics are driven by a set of differential equations with several parameters describing the intrinsic population properties: the reproduction rate of the predator α, the death rate of the predator β, the reproduction rate of the prey γ, and the predation rate of the prey δ, thereby summarizing the model parameter as $\boldsymbol{\theta} \triangleq (\alpha, \beta, \gamma, \delta)$.

Setup The evolution of the populations is modeled as a Markov jump process. We consider the same simulation model as in [8]. As before, $1,000,000$ simulations are generated from the prior $p(\boldsymbol{\theta}) = \mathcal{U}([0, 0.2] \times [0, 2] \times [0, 2] \times [0, 0.2])$. We start with an initial prey population of 100, and an initial predator population of 50 and simulate the population dynamics for 30 time-units with a resolution of 0.2. This results in a 2×151-dimensional matrix representing the state of the prey and predator populations over time. Subsequently, the content of the matrix is normalized between $[0, 1]$ to dampen the effect of large population sizes on the neural network. The processed matrix and a standardized $\boldsymbol{\theta}$ are fed into a batch normalized MLP with RELU non-linearities. The standardization of $\boldsymbol{\theta}$ is computed using the mean and variance of $p(\boldsymbol{\theta})$, which is known a priori. The summary statistic in ABC follows the definition from [8]: the mean of the population sizes and the autocorrelation of the population time series at different lags. Observations are simulated from $\boldsymbol{\theta}^* \triangleq (0.01, 0.5, 1.0, 0.01)$.

Results The approximated posteriors are depicted in Figure 5. The posteriors approximated by LF-MH and ABC exhibit similar shapes. As the ABC posterior is smeared out by the acceptance threshold ϵ, the variance of the LF-MH posterior is expected to be smaller. Additionally, for the considered prior, LF-MH is significantly more simulation-efficient than ABC. The acceptance rate of ABC under an acceptance threshold of $\epsilon = 2.5$ was $\approx 1.8 \times 10^{-4}$ indicating that ABC computed in the order of 5×10^9 simulations for an equivalent amount of posterior samples. In contrast, the proposed method has a constant simulation cost. However, the acceptance rate of ABC could be improved by relying on more efficient sampling procedures such as SMC-ABC [20].



(a) LF-MH



(b) ABC

Figure 5: Approximate posterior for the Lotka-Volterra population model. ABC samples have been drawn under an acceptance threshold of $\epsilon = 2.5$.

4.3 Particle physics detector alignment

To inspect the performance of the proposed method in the presence of high-dimensional observations, we turn to a particle physics inference problem. We consider the PYTHIA simulator [21] configured according to the parameters derived by the Monash tune [22, 23]. Under this configuration, PYTHIA simulates $e^- e^+$ collisions

at a center-of-mass energy of 91.2 GeV. The product of these collisions is a Z boson which decays to some quarks. These quarks are then observed by a detector, which we design as a 1-layered pixelized spherical detector [24] with a radius of 30 units, emulating a 32×32 grid in pseudorapidity η and azimuthal angle ϕ, covering $(\eta, \phi) \in [-5, 5] \times [0, 2\pi]$. Additionally, the detector does not record the energy of a particle passing through its material, but only indicates a hit (i.e., a tracker). The detector is parameterized by an offset parameter $\boldsymbol{\theta}$ in the z-axis parallel to the beam axis. An offset of $\boldsymbol{\theta} = 0$ means that the center of the spherical detector is centered at the collision point.

Setup We generate $1,000,000$ simulations with $\boldsymbol{\theta}$ drawn from the prior $p(\boldsymbol{\theta}) = \mathcal{U}(-30, 30)$ and 100 e^-e^+ collisions per simulation. The parameterized classifier is a modification of VGG-11 [25] with batch-normalization [26] in the convolutional layers. The dependence on $\boldsymbol{\theta}$ is added in the final fully-connected part of the classifier. While VGG might be overpowered for this particular problem, we made this decision to demonstrate that off-the-shelf convolutional neural networks can be used in the proposed method. Due to the high-dimensional and binary nature of the observations we apply the cosine-similarity as the distance function in ABC with an acceptance threshold of $\epsilon = 0.03$. Lowering the acceptance threshold has a significant computational impact, making the rejection sampling scheme impractical.

Results Figure 6 clearly shows the proposed technique is outperforming ABC. The large variance and flatness of ABC posteriors might be attributable to the insufficiency of the summary statistic. It is especially challenging to construct a sufficient summary statistic due to the high variability of the high-dimensional observations caused by the underlying physical processes. Despite the absence of domain knowledge, the classifier is able to extract an approximate posterior which is in agreement with the generating parameter for 100 e^-e^+ collisions.

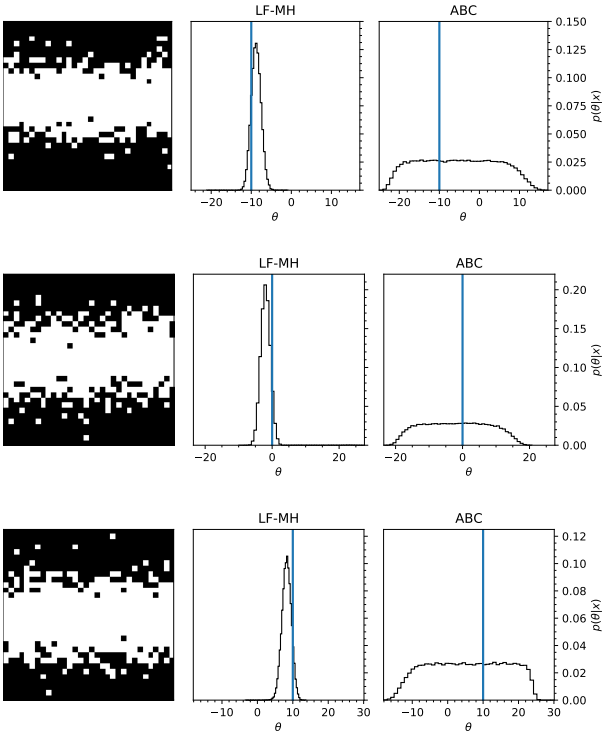


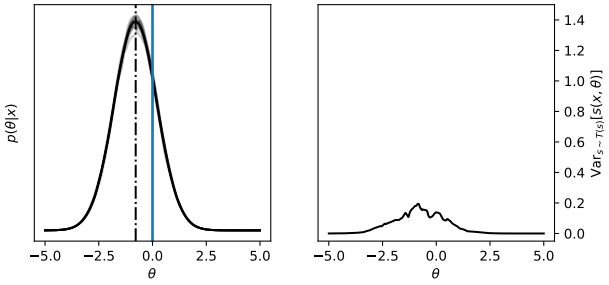
Figure 6: Particle tracker alignment under a single observation with 100 e^-e^+ collisions. We consider three settings from PYTHIA simulations. *(*Top): $\boldsymbol{\theta}^* = -10$. While ABC has significantly higher variance, the density is concentrated on the negative range of the offset, suggesting the detector offset is in $\boldsymbol{\theta}^-$. No conclusive decision can be made under this particular ϵ-threshold. Similar arguments can be made for *(*Middle): $\boldsymbol{\theta}^* = 0$, and *(*Bottom): $\boldsymbol{\theta}^* = 10$. Employing LF-MH, the same classifier has been used across all settings. This shows that a single likelihood ratio model can be applied to a variety of problems after an expensive training procedure.

5 Diagnostics

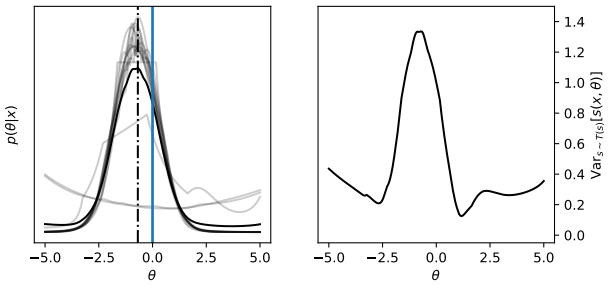
5.1 Classifier convergence

We make the observation that given a fixed and sufficiently large dataset, the increase in the capacity or representational power of a classifier $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ leads to a decrease in the variance of $\hat{r}_e(\mathbf{x}, \boldsymbol{\theta})$ over classifiers sampled from a optimization procedure $\mathbf{s} \sim T(\mathbf{s})$, as the error with respect to the best probabilistic classifier decreases.

The probabilistic classifier can be used to obtain the likelihood-to-evidence ratio. As this ratio is unique for a given observation $\mathbf{x}$ and model parameter $\boldsymbol{\theta}$, one can assess the classifier convergence by computing $\mathrm{Var}_{\mathbf{s}\sim T(\mathbf{s})}\left[\hat{r}_e(\mathbf{x}, \boldsymbol{\theta})\right]$. If the variance is large, then the classifier capacity is insufficient to approximate the likelihood-to-evidence ratio, which might result in a miss-estimation of the posterior. If the variance is small, we cannot necessarily conclude that the posterior is well-estimated. However, if the classifier capacity is large enough, then the approximation error of the likelihood-to-evidence ratio is likely to be small, which would result in a good approximate posterior. Figure 7 demonstrates this principle empirically. Additionally, this diagnostic can be useful in an architecture-search scheme, in which the architecture of the parameterized classifier is optimized to minimize this variance.



(a) 64 hidden units



(b) 2 hidden units

Figure 7: We train 15 high and low capacity classifiers, each having 64 and 2 hidden units respectively with a single hidden layer. We perform posterior inference with a single observation $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta} = 0, 1)$. The sample variance indicates that the resolution error of the low capacity classifiers is much greater compared to the high capacity classifiers. In some regions the sample variance of the classifier is 0, indicating a perfect approximation.

5.2 Simulation based calibration

Earlier work [27] proposes a method to verify the Bayesian computation of software. The diagnostic is based on the observation that if $\boldsymbol{\theta}' \sim p(\boldsymbol{\theta})$ and $\mathbf{x} \sim p(\mathbf{x} \mid \boldsymbol{\theta})$, then exact posterior quantiles for each parameter will be uniformly distributed, provided that the posteriors are absolutely continuous [28]. However, this diagnostic is not guaranteed to function on posterior samples [28]. Simulation Based Calibration (SVB) [28] circumvents this issue by discretizing the function f into a rank-statistic $f \colon \Theta \mapsto \mathbb{R}$. SVB ranks L independent samples of the posterior and approximates the quantiles by histogramming. The only assumption is the availability of an implicit model (simulator), thereby making the proposed technique perfectly suitable for diagnosing likelihood-free Bayesian approximations. The vanilla technique is not directly applicable to MCMC samplers (due to sample auto-correlation), as the technique requires *independent* posterior samples. This can be addressed by ensuring that the effective sample size is at least L after uniform thinning of the Markov chain. Figure 8 shows a histogram of rank-statistics for a classifier $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$. Bins outside of the shaded area would indicate a poor approximation of the posterior.

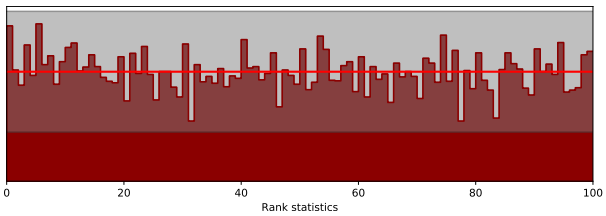


Figure 8: Histogram over $L = 100$ rank statistics and $10,000$ independent tests computed using SVB [28]. Bins outside the shaded indicate a poor approximation.

6 Related work

Sampling from an implicit model $p(\mathbf{x} \mid \boldsymbol{\theta})$ to perform statistical inference [29, 30] is a common technique in science to bypass the intractability of the likelihood. Works applying this technique often rely on rejection sampling schemes such as *A*pproximate Bayesian Computation [31, 32]. ABC approximates the posterior by accepting proposed states $\boldsymbol{\theta}'$. This is done by comparing simulated samples $\mathbf{x} \sim p(\mathbf{x} \mid \boldsymbol{\theta}')$ against observations $\mathbf{x}_o$ under a sufficient summary statistic $\mathbf{s}$. A proposed state $\boldsymbol{\theta}' \sim p(\boldsymbol{\theta})$ is accepted when the distance of these summary statistics under some divergence ρ is smaller or equal to ϵ, i.e., $\rho(\mathbf{s}(\mathbf{x}), \mathbf{s}(\mathbf{x}_o)) \leq \epsilon$. Even when $\mathbf{s}(\mathbf{x})$ is low-dimensional, the acceptance rate of proposals $\boldsymbol{\theta}'$ can severely be affected by the prior $p(\boldsymbol{\theta})$, i.e., a large misspecified prior can cause a lot of rejections. In recent years variants of ABC have been proposed to improve the acceptance rate by guiding simulations based on previously accepted states [20, 33, 34, 35]. Such approaches are crucial, as the acceptance rate drastically decreases when $\epsilon \to 0$. On the other hand, approaches such as [36] remove the need for handcrafted summary statistics. While ABC yields unbiased samples of $p(\boldsymbol{\theta} \mid \mathbf{x})$ when $\epsilon \to 0$, it suffers from high variance as $\epsilon > 0$ in most settings.

Other approaches take a new perspective and cast inference as an optimization problem [37, 38]. In variational inference (VI), a parameterized posterior $q_\psi(\boldsymbol{\theta} \mid \mathbf{x})$ over parameters of interest $\boldsymbol{\theta}$ is optimized to minimize $\mathrm{KL}(q \mid\mid p)$ [39]. Amortized inference [40, 41] expands on this idea by using generative models to capture inference mappings. However, this typically leads to an amortization gap [42, 43] which has been extensively studied by [44]. Recent work in [45] proposes a novel form of variational inference by introducing an adversary in combination with REINFORCE-estimates [46, 47] to optimize a parameterized prior. This work lends itself

naturally to a likelihood-free setting, but only provides point-estimates.

Similarly, approaches like Sequential Neural Posterior Estimation (SNPE) [10] borrow ideas from VI literature and iteratively adjusts a parameterized posterior defined as a Mixture Density Network [48]. Instead of learning the posterior directly, Sequential Neural Likelihood (SNL) [8] uses modern conditional density estimators such as autoregressive flows [11, 12, 9, 13] which serve as a proxy for the likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$. Having access to this proxy, one can use traditional MCMC-techniques to draw samples from the posterior. While SNL relies on invertible operations, it also requires careful tuning to test the expressiveness of the autoregressive flow. On the other hand, employing our approach one can rely on standard optimization techniques to train the parameterized classifier. Additionally, the sequential nature of SNL makes the technique susceptible to bias when initial samples do not sufficiently represent the likelihood.

Early likelihood-free MCMC samplers relied on the ABC rejection sampling scheme to draw proposals. These samples are in turn evaluated by the Metropolis-mechanism under a summary statistic [34, 49]. Other likelihood-free alternatives with theoretical guarantees can be constructed by utilizing pseudo-marginal [50, 51] approaches, as these only require unbiased point estimates. Typically, such unbiased estimators are obtained by importance sampling.

An important aspect of likelihood-free inference is minimizing the number of simulation calls while retaining the inference quality. Active simulation strategies such as BOLFI [52] and others [53, 54] achieve this by employing Bayesian optimization to make efficient use of the simulator. Other approaches attempt to learn approximate versions of the simulator [55] which is used to perform efficient inference, adapting a similar strategy as in world models for reinforcement learning [56]. Our method relies on the training of an accurate likelihood-to-evidence ratio model, which would typically come with a high initial cost. However, recent works [57, 58] make it possible to significantly reduce the cost of this step, provided joint likelihood ratios and scores can be extracted from the simulator.

7 Summary & discussion

We present an approach to perform likelihood-free posterior sampling by approximating likelihood ratios and embedding these ratios in Markov chain Monte Carlo samplers. The likelihood ratios are approximated by training a parameterized classifier to distinguish samples from the likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ and evidence $p(\mathbf{x})$. In turn, the transformed output of the classifier can be used to approximate the likelihood ratio between arbitrary model parameters. Due to the MCMC nature of the sampling procedure, the presented approach does not suffer from the computational inefficiencies that are present in traditional rejection sampling schemes. Experimental results demonstrate the efficiency, robustness and ease-of-use of the proposed method, enabling the use of existing high-performance neural architectures. Contrary to existing schemes, ours does not require the classifier to be retrained when presented with a different set of observations. However, this comes at the significant upfront cost of generating training set tuples $(\boldsymbol{\theta}, \mathbf{x}_{\boldsymbol{\theta}})$ and additional requirements for sufficient classifier capacity.

Likelihood-free MCMC is of interest in domain sciences such as astronomy and high-energy physics which typically rely on computer simulations to describe complex generative processes, as these provide a natural way to perform inference. However, before making any conclusion in a scientific process, a formal understanding of the effect of bias and variance in the classifier would be of interest.

Other interesting directions for future work include the extraction of state proposals from the parameterized classifier instead of using fixed transition distributions or relying on an expensive computation of the Hamiltonian dynamics.

Acknowledgements

Joeri Hermans would like to thank the F.R.S.-FNRS for his FRIA scholarship. Partial Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

Appendix

A Iterative posterior estimation

In case the capacity of a trained classifier is insufficient with respect to a given prior $p(\boldsymbol{\theta})$, its output may be biased. The bias originates from the approximation error of the likelihood-to-evidence ratio and is attributed to the discrepancy between the estimator and the optimal classifier. Typically, this bias will over- or underestimate the likelihood-to-evidence ratio, which translates into an over- or underestimation of the approximated posterior. While we do not describe the effects of the bias on the approximate posterior formally, empirical results indicate that the bias is expected to overestimate the posterior density. This observation motivates the design of a posterior estimation technique which frees up some capacity of $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ by iteratively reducing the bounds of a prior. This procedure is summarized in Algorithm 4.

Algorithm 4 Iterative likelihood-free MCMC

Inputs: Initial prior $p(\boldsymbol{\theta})$.
Observations $\mathcal{O}$.
Outputs: Markov chain $\boldsymbol{\theta}_{0:n}$
Hyperparameters: Rounds k.

$t \leftarrow 0$
for $t < k$ **do**
  $\mathbf{s} \leftarrow \text{OPTIMIZE}(p(\boldsymbol{\theta}))$
  $\boldsymbol{\theta}_{0:n} \leftarrow \text{LF-MCMC}(\mathbf{s}, \mathcal{O})$
  $p(\boldsymbol{\theta}) \leftarrow \text{REFINE}(\boldsymbol{\theta}_{0:n})$
  $t \leftarrow t + 1$
end for
return $\boldsymbol{\theta}_{0:n}$

A.1 Demonstration

Consider the following inference task. An implicit model generates 32×32 images of circles within a $[-1, 1] \times [-1, 1]$ coordinate system. The simulator takes $\boldsymbol{\theta} \triangleq (x, y, r)$ as input, where x and y are the coordinates of the circle's center and r is its radius. The goal is to infer $\boldsymbol{\theta}^* = (0, 0, 0.5)$ given a single observation $\mathbf{x}_o$. The parameterized classifier for this task is a simple MLP with SELU non-linearities. We intentionally limit its capacity to demonstrate the proposed procedure. Initially, the prior at round $t = 0$ is $p(\boldsymbol{\theta}) = \mathcal{U}([-1, 1] \times [-1, 1] \times [-1, 1])$. After the first iteration of Algorithm 4, we obtain a Markov chain $m \triangleq \boldsymbol{\theta}_{0:n}$. Figure 9 shows the approximate posterior as estimated by m. At the beginning of the second round, we adjust the prior bounds whenever the approximate posterior shrinks them. In this case, we set the lower bound of the prior in round $t+1$ to $\min(m) - \sigma(m)$, where $\sigma(\cdot)$ computes the standard deviation of the specified chain. Likewise, the upper bound is set to $\max(m) + \sigma(m)$. Both quantities are constrained by the initial prior. While the margins introduced by σ are in essence not required, we introduce them to account for possible Monte Carlo error of the sampler and the bias of the parameterized classifier. After setting the refined prior $p(\boldsymbol{\theta})$, we launch the second iteration of classifier training and MCMC sampling. Figure 10 shows the final approximate posterior.

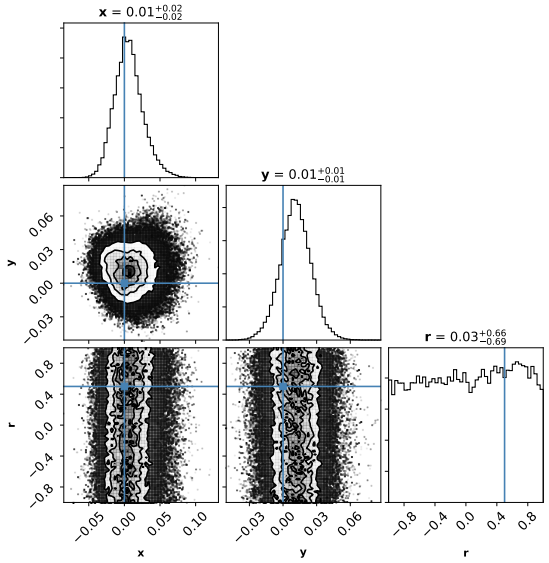


Figure 9: Approximate posterior after the first round. $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ is not able to capture the radius.

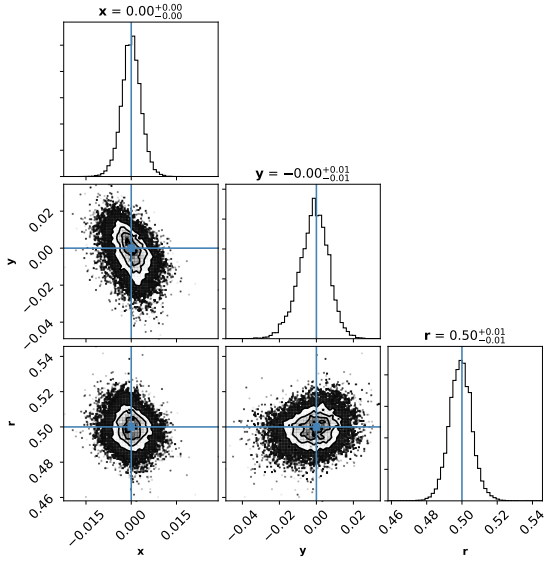


Figure 10: Approximate posterior after the second round.

References

[1] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

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

[3] Kyle Cranmer, Juan Pavez, and Gilles Louppe. Approximating likelihood ratios with calibrated discriminative classifiers. *arXiv preprint arXiv:1506.02169*, 2015.

[4] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

[5] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933.

[6] Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. Likelihood-free inference by ratio estimation. *arXiv preprint arXiv:1611.10242*, 2016.

[7] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[8] George Papamakarios, David C Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. *arXiv preprint arXiv:1805.07226*, 2018.

[9] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *Journal of Machine Learning Research*, 17(205):1–37, 2016.

[10] Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems*, pages 1289–1299, 2017.

[11] George Papamakarios, Iain Murray, and Theo Pavlakou. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

[12] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[13] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: masked autoencoder for distribution estimation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *JMLR: W&CP*, pages 881–889, 2015.

[14] Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987.

[15] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[16] Edward Meeds, Robert Leenders, and Max Welling. Hamiltonian abc. *arXiv preprint arXiv:1503.01916*, 2015.

[17] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 971–980, 2017.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Darren J Wilkinson. *Stochastic modelling for systems biology*. Chapman and Hall/CRC, 2006.

[20] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael PH Stumpf. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2008.

[21] Torbjörn Sjöstrand, Stefan Ask, Jesper R Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O Rasmussen, and Peter Z Skands. An introduction to pythia 8.2. *Computer physics communications*, 191:159–177, 2015.

[22] Peter Skands, Stefano Carrazza, and Juan Rojo. Tuning pythia 8.1: the monash 2013 tune. *The European Physical Journal C*, 74(8):3024, 2014.

[23] Philip Ilten, Mike Williams, and Yunjie Yang. Event generator tuning using bayesian optimization. *Journal of Instrumentation*, 12(04):P04028, 2017.

[24] Maxim Borisyak. Pythia-mill. https://gitlab.com/mborisyak/pythia-mill/, 2018.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[27] Samantha R Cook, Andrew Gelman, and Donald B Rubin. Validation of software for bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3):675–692, 2006.

[28] Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788*, 2018.

[29] Peter J Diggle and Richard J Gratton. Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 193–227, 1984.

[30] Florian Hartig, Justin M Calabrese, Björn Reineking, Thorsten Wiegand, and Andreas Huth. Statistical inference for stochastic simulation models–theory and application. *Ecology letters*, 14(8):816–827, 2011.

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

[32] Jean-Michel Marin, Pierre Pudlo, Christian P Robert, and Robin J Ryder. Approximate bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

[33] Simon Tavaré, David J Balding, Robert C Griffiths, and Peter Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.

[34] 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.

[35] Daniel Wegmann, Christoph Leuenberger, and Laurent Excoffier. Efficient approximate bayesian computation coupled with markov chain monte carlo without likelihood. *Genetics*, 2009.

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

[37] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[38] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[39] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226, 2015.

[40] Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36, 2014.

[41] Daniel Ritchie, Paul Horsfall, and Noah D Goodman. Deep amortized inference for probabilistic programs. *arXiv preprint arXiv:1610.05735*, 2016.

[42] Rahul G Krishnan, Dawen Liang, and Matthew Hoffman. On the challenges of learning with inference networks on sparse, high-dimensional data. *arXiv preprint arXiv:1710.06085*, 2017.

[43] Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. *arXiv preprint arXiv:1801.03558*, 2018.

[44] Joseph Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. *arXiv preprint arXiv:1807.09356*, 2018.

[45] Gilles Louppe, Joeri Hermans, and Kyle Cranmer. Adversarial variational optimization of non-differentiable simulators. *arXiv preprint arXiv:1707.07113*, 2017.

[46] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[47] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[48] Christopher M Bishop. Mixture density networks. Technical report, Citeseer, 1994.

[49] Scott A Sisson and Yanan Fan. Likelihood-free markov chain monte carlo. *arXiv preprint arXiv:1001.2058*, 2010.

[50] 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.

[51] Iain Murray and Matthew Graham. Pseudo-marginal slice sampling. In *Artificial Intelligence and Statistics*, pages 911–919, 2016.

[52] 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.

[53] Victor MH Ong, David J Nott, Minh-Ngoc Tran, Scott A Sisson, and Christopher C Drovandi. Variational bayes with synthetic likelihood. *Statistics and Computing*, 28(4):971–988, 2018.

[54] Edward Meeds and Max Welling. Gps-abc: Gaussian process surrogate approximate bayesian computation. *arXiv preprint arXiv:1401.2838*, 2014.

[55] Jan-Matthis Lueckmann, Giacomo Bassetto, Theofanis Karaletsos, and Jakob H Macke. Likelihood-free inference with emulator networks. *arXiv preprint arXiv:1805.09294*, 2018.

[56] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[57] Johann Brehmer, Gilles Louppe, Juan Pavez, and Kyle Cranmer. Mining gold from implicit models to improve likelihood-free inference. *arXiv preprint arXiv:1805.12244*, 2018.

[58] Johann Brehmer, Kyle Cranmer, Gilles Louppe, and Juan Pavez. A guide to constraining effective field theories with machine learning. *Physical Review D*, 98(5):052004, 2018.