Metropolis–Hastings via Classification

Tetsuya Kaji and Veronika Ročková
Booth School of Business, University of Chicago, Chicago, IL

ABSTRACT
This article develops a Bayesian computational platform at the interface between posterior sampling and optimization in models whose marginal likelihoods are difficult to evaluate. Inspired by contrastive learning and Generative Adversarial Networks (GAN), we reframe the likelihood function estimation problem as a classification problem. Pitting a Generator, who simulates fake data, against a Classifier, who tries to distinguish them from the real data, one obtains likelihood (ratio) estimators which can be plugged into the Metropolis–Hastings algorithm. The resulting Markov chains generate, at a steady state, samples from an approximate posterior whose asymptotic properties we characterize. Drawing upon connections with empirical Bayes and Bayesian misspecification, we quantify the convergence rate in terms of the contraction speed of the actual posterior and the convergence rate of the Classifier. Asymptotic normality results are also provided which justify the inferential potential of our approach. We illustrate the usefulness of our approach on examples which have challenged for existing Bayesian likelihood-free approaches. Supplementary materials for this article are available online.

1. Introduction

Many contemporary statistical applications require inference for models which are easy to simulate from but whose likelihoods are impossible to evaluate. This includes implicit (simulator-based) models Diggle and Gratton (1984), defined through an underlying generating mechanism, or models prescribed through intractable likelihood functions.

Statistical inference for intractable models has traditionally relied on some form of likelihood approximation (see Gutmann and Corander 2016 for a recent excellent survey). For example, Diggle and Gratton (1984) propose kernel log-likelihood estimates obtained from simulated realizations of an implicit model. Approximate Bayesian Computation (ABC) Beaumont, Zhang, and Balding (2002), Pritchard et al. (1999), and Tavaré et al. (1997) is another simulation-based approach which obviates the need for likelihood evaluations by (a) generating fake data \( \tilde{X}_\theta \) for parameter values \( \theta \) sampled from a prior, and (b) weeding out those pairs \( (X_\theta, \theta) \) for which \( \tilde{X}_\theta \) has low fidelity to observed data. The discrepancy between observed and fake data is evaluated by first reducing the two datasets to a vector of summary statistics and then measuring the distance between them. Both the distance function and the summary statistics are critical for inferential success. While eliciting suitable summary statistics often requires expert knowledge, automated approaches have emerged Blum et al. (2013), Gutmann et al. (2018), and Bernton et al. (2019). Notably, Fearnhead and Prangle (2011) proposed a semiautomated approach that approximates the posterior mean (a summary statistic that guarantees first-order accuracy) using a linear model regressing parameter samples onto simulated data. Subsequently, Jiang et al. (2017) elaborated on this strategy using deep neural networks which are expected to yield better approximations to the posterior mean. Beyond subtleties associated with summary statistics elicitation, ABC has to be deployed with caution for Bayesian model choice Robert et al. (2011) and Marin et al. (2014). Synthetic likelihood (SL) Wood (2010) and Price et al. (2018) is another approach for carrying out inference in intractable models by constructing a proxy Gaussian likelihood for a vector of summary statistics. Implicit in the success of both ABC and SL is the assumption that the generating process can produce simulated summary statistics that adequately represent the observed ones. If this compatibility is not satisfied (e.g., in misspecified models), both SL Frazier and Drovandi (2021) and ABC Frazier, Robert, and Rousseau (2020) can provide unreliable estimates. Besides SL, a wide range parameteric surrogate likelihood models have been suggested including normalizing flows, Gaussian processes or neural networks Gutmann and Corander (2016), Blum (2010), Durkan, Murray, and Papamakarios (2020), and Papamakarios and Murray (2016). Avoiding the need for summary statistics, Gutmann et al. (2018) proposed to use discriminability of the observed and simulated data as a discrepancy measure in ABC. Their accepting/rejecting mechanism separates samples based on a discriminator’s ability to tell the real and fake data apart. Similarly as their work, our paper is motivated by the observation that distinguishing two datasets is usually easier if they were simulated with very different parameter values. However, instead of deploying this strategy inside ABC, we embed it directly inside the Metropolis–Hastings algorithm using likelihood approximations obtained from classification.

The Metropolis–Hastings (MH) method generates ergodic Markov chains through an accept–reject mechanism which
depends in part on likelihood ratios comparing proposed candidate moves and current states. For many latent variable models, the marginal likelihood is not available in closed form, making direct application of MH impossible (see Deligiannidis, Doucet, and Pitt 2018 for examples). The pseudo-marginal likelihood method Andrieu and Roberts (2009) offers a remedy by replacing likelihood evaluations with their (unbiased) estimates. Many variants of this approach have been proposed including the inexact MCWM method (described in O’Neill et al. 2000; Andrieu and Roberts 2009) and its elaborations that correct for bias O’Neill et al. (2000), reduce the variance of the likelihood ratio estimator Deligiannidis, Doucet, and Pitt (2018) or make sure that the resulting chain produces samples from the actual (not only approximate) posterior Beaumont (2003). The idea of using likelihood approximations within MH dates back to at least O’Neill et al. (2000) and has been implemented in a series of works (see, e.g., O’Ryan et al. 1998; Beaumont 2003 and references therein). Our approach is fundamentally different from many typical pseudo-marginal MH algorithms since it does not require a hierarchical model where likelihood estimates are obtained through simulation from conditionals of latent data. Our method can be thus applied in a wide range of generative models (where forward simulation is possible) and other scenarios (such as diffusion processes Heston 1993) where PM methods would be cumbersome or time-consuming to implement (as will be seen later in our examples).

Inspired by contrastive learning (CL) Hastie, Tibshirani, and Friedman (2009) and Gutmann and Hyvarinen (2012) we reframe the likelihood (ratio) estimation problem as a classification problem using the “likelihood-ratio trick” Durkan, Murray, and Papamakarios (2020), Cranmer, Pavez, and Louppe (2016), Thomas et al. (2021), and Sugiyama, Suzuki, and Kanamori (2012). Similarly as with generative adversarial networks (GANs) Goodfellow et al. (2014), we pit two agents (a Generator and a Classifier) against one another. Assessing the similitude between the fake data, outputted by the Generator, and observed data, the Classifier provides likelihood estimators which can be deployed inside MH. The resulting algorithm provides samples from an approximate posterior.

Our contributions are both methodological and theoretical. We develop a personification of Metropolis–Hastings (MH) algorithm for intractable likelihoods based on Classification, further referred to as MHC. We consider two variants: (a) a fixed generator design which may yield biased samples, and (b) a random generator design which may yield unbiased samples with increased variance. We then describe how and when the two can be combined in order to provide posterior samples with asymptotically correct location and spread. Contrastive learning has been suggested in the context of posterior simulation before Heermans, Begy, and Louppe (2020) and Pham, Nott, and Chaudhuri (2014). Our approach differs in the choice of the contrasting density and, in addition, we develop theory which was previously unavailable. Our theoretical analysis consists of new convergence rate results for a posterior residual (an approximation error) associated with the Classifier. These rates are then shown to affect the rate of convergence of the stationary distribution, in a similar way as the ABC tolerance level affects the convergence rate of ABC posteriors Frazier et al. (2018). Theoretical developments for related pseudo-marginal (PM) methods have been concentrating on convergence properties of the Markov chain such as mixing rates Andrieu and Roberts (2009) and Deligiannidis, Doucet, and Pitt (2018). Here, we provide a rigorous asymptotic study of the stationary distribution including convergence rates (drawing upon connections to empirical Bayes and Bayesian misspecification), asymptotic normality results and, in addition, polynomial mixing time characterizations of the Markov chain.

To illustrate that our MHC procedure can be deployed in situations when sampling from conditionals of latent data (often needed for PM) is not practical or feasible, we consider two examples. The first one entails discretizations of continuous-time processes which are popular in finance Heston (1993), and Cox, Ingersoll, and Ross (1985). The second one is a population-evolution generative model where PM is not straightforward and where ABC methods need strong informative priors and high-quality summaries. In both examples, we demonstrate that MHC offers a reliable practical inferential alternative which is straightforward to implement. We also show very good performance on a Bayesian model choice example (where ABC falls short) and on the famous Ricker model (Section 10 in the supplementary materials) Ricker (1954) analyzed by multiple authors Gutmann and Corander (2016), Wood (2010), and Fearnhead and Prange (2011).

The article is structured as follows. Sections 2 and 3 introduce the classification-based likelihood ratio estimator and the MHC sampling algorithm. Section 4 then describes the asymptotic properties of the stationary distribution. Section 5 shows demonstrations on simulated data and, finally, Section 6 wraps up with a discussion.

2. Likelihood Estimation with a Classifier

Our framework consists of a series of iid observations \( \{X_i\}_{i=1}^{\infty} \in \mathcal{X} \) realized from a probability measure \( P_{\theta_0} \) indexed by a parameter \( \theta_0 \in \Theta \) which is endowed with a prior \( \Pi_{\theta_0}() \). We assume that \( P_{\theta} \) for each \( \theta \in \Theta \), admits a density \( p_{\theta} \). Our objective is to draw observations from the posterior density \( X^{(n)} = (X_1, \ldots, X_n)' \) defined through

\[
\pi_{\theta}(X^{(n)}) = \frac{p_{\theta}^{(n)}(X^{(n)})\pi(\theta)}{\int_{\Theta} p_{\theta}^{(n)}(X^{(n)}) d\Pi(\theta)},
\]

where \( p_{\theta}^{(n)} = \prod_{i=1}^{n} p_\theta(X_i) \). Our focus is on situations where the likelihood \( p_{\theta}^{(n)} \) is too costly to evaluate but can be readily sampled from.

We develop a Bayesian computational platform at the interface between sampling and optimization inspired by contrastive learning (CL) Hastie, Tibshirani, and Friedman (2009) and Gutmann and Hyvarinen (2012) and Generative Adversarial Networks (GAN) Goodfellow et al. (2014). The premise of GANs is to discover rich distributions over complex objects arising in artificial intelligence applications through simulation. The learning procedure consists of two entities pitted against one another. A Generator aims to deceive an Adversary by simulating samples that resemble the observed data while, at the same time, the Adversary learns to tell the fake and real data apart. This process iterates until the generated data are indistinguishable by the Adversary. While GANs have found
their usefulness in simulating from distributions over images, here we forge new connections to Bayesian posterior simulation.

Similarly as with GANs, we assume a Generator transforming a set of latent variables \( \bar{X} \in \hat{X} \) to collect samples from \( P_0 \) through a known deterministic mapping \( T_\theta : \hat{X} \rightarrow X \), that is, \( T_\theta(\bar{X}) \sim P_0 \) for \( \bar{X} \sim \hat{P} \) for some distribution \( \hat{P} \) on \( \hat{X} \). This implies that we can draw a single set of \( m \) observations \( \bar{X}^{(m)} \) and then filter them through \( T_\theta \) to obtain a sample \( \hat{X}^{(m)}_\theta = T_\theta(\bar{X}^{(m)}) \) from \( P_0 \) for any \( \theta \in \Theta \). Being able to easily draw samples from the model suggests the intriguing possibility of learning density ratios "by-comparison" Mohamed and Lakshminarayanan (2015). Indeed, the fact that density ratios can be computed by building a classifier that compares two datasets Durkan, Murray, and Papamakarios (2020) and Cranmer, Pavez, and Louppe (2016) has lead to an emergence of a rich ecosystem of algorithms for model-free inference Thomas et al. (2021), Heermans, Begy, and Louppe (2020), Papamakarios and Murray (2016), and Pham, Nott, and Chaudhuri (2014). Many of these machine learning procedures are based on variants of the "likelihood ratio trick" (LRT) which builds a surrogate classification model for the likelihood ratio. Similarly as Pham, Nott, and Chaudhuri (2014) and Heermans, Begy, and Louppe (2020), we embed the LRT within a classical Bayesian sampling algorithm and furnish our procedure with rigorous frequentist-Bayesian inferential theory.

Our approach relies on the simple fact that a cross-entropy classifier can be deployed to obtain an estimator of the likelihood ratio Hastie, Tibshirani, and Friedman (2009), Gutmann and Hyvärinen (2012), and Thomas et al. (2021). Recall that the classification problem with the empirical cross-entropy loss is defined through

\[
\max_{D \in \mathcal{D}} \left[ -\frac{1}{n} \sum_{i=1}^{n} \log D(X_i) + \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(X_i^{\theta})) \right],
\]

where \( \mathcal{D} \) is a set of measurable classification functions \( D : \mathcal{X} \rightarrow (0, 1) \) (1 for "real" and 0 for "fake" data) and \( X_i^{\theta} \sim T_\theta(X_i) \) for \( i = 1, \ldots, m \) are the "fake" data outputted by the Generator. If an oracle were to furnish the true model \( p_0 \), it is known that the population solution to (2.2) is the "Bayes classifier" (see Section 14.2.4 in Hastie, Tibshirani, and Friedman 2009; Proposition 1 in Goodfellow et al. 2014)

\[
D_\theta(X) := \frac{p_0(X)}{p_0(X) + \hat{p}_0(X)} \quad \text{for} \quad X \in \mathcal{X}. \tag{2.3}
\]

Reorganizing the terms in (2.3), the likelihood can be written (see, e.g., Thomas et al. 2021) in terms of the discriminator function as

\[
p_0^{(n)}(X^{(n)}) = p_0^{(n)}(X^{(n)}) \exp \left( \sum_{i=1}^{n} \log \frac{1 - D_\theta(X_i)}{D_\theta(X_i)} \right). \tag{2.4}
\]

The oracle discriminator \( D_\theta(\cdot) \) depends on \( p_0 \) but can be estimated by simulation. Indeed, one can deploy the Generator to simulate the fake data \( \hat{X}^{(m)}_\theta = T_\theta(\bar{X}^{(m)}) \) and train a Classifier to distinguish them from \( X^{(n)} \). The Classifier outputs an estimator \( \hat{D}^{(n)}_{n,m} \), for which we will see examples, and which can be plugged into (2.4) to obtain the following likelihood estimator

\[
\hat{p}_0^{(n)}(X^{(n)}) = \frac{p_0^{(n)}(X^{(n)})}{\exp \left( \sum_{i=1}^{n} \log \frac{1 - \hat{D}^{(n)}_{n,m}(X_i)}{\hat{D}^{(n)}_{n,m}(X_i)} \right)}, \tag{2.5}
\]

where

\[
u_\theta(X^{(n)}) := \sum_{i=1}^{n} \log \frac{1 - \hat{D}^{(n)}_{n,m}(X_i)}{\hat{D}^{(n)}_{n,m}(X_i)} \tag{2.6}
\]

will be further referred to as the log-posterior residual. In other words, (2.5) is a deterministic functional of auxiliary random variables \( \bar{X}^{(m)} \) and the observed data \( X^{(n)} \), and can be computed (up to a norming constant) from \( \hat{D}^{(n)}_{n,m} \). The posterior density \( \pi_n(\theta | X^{(n)}) \) can be then estimated by replacing \( D_\theta \) with \( \hat{D}^{(n)}_{n,m} \) in the likelihood expression to obtain

\[
\hat{\pi}_{n,m}(\theta | X^{(n)}) := \exp \left( \sum_{i=1}^{n} \log \frac{1 - \hat{D}^{(n)}_{n,m}(X_i)}{\hat{D}^{(n)}_{n,m}(X_i)} \right),
\]

\[
\pi(\theta) \propto \pi_n(\theta | X^{(n)}) e^{u_\theta(X^{(n)})}. \tag{2.7}
\]

Two observations ought to be made. First, the estimator (2.7) targets the posterior density only up to a norming constant. This will not be an issue in Bayesian algorithms involving posterior density ratios (such as the Metropolis–Hastings algorithm considered here). Second, the estimator (2.7) performs exponential tilting of the original posterior, where the quality of the approximation crucially depends on the statistical properties of \( u_\theta(X^{(n)}) \). Note that \( u_\theta(X^{(n)}) \) depends also on the latent data \( \bar{X}^{(m)}_\theta \). We devote the entire Section 4.1 to statistical properties of \( u_\theta(X^{(n)}) \). The idea of estimating likelihood ratios via discriminative classifiers has emerged in various contexts including hypothesis testing Cranmer, Pavez, and Louppe (2016) and posterior density estimation Thomas et al. (2021). An important distinguishing feature of our approach is that we contrast observed and fake data, using the truth \( \theta_0 \) as a fixed reference point. This is different from the marginal approach in Thomas et al. (2021) which contrasts two fake datasets generated from the marginal and conditional likelihoods. We highlight the connections in Section 9 in the supplementary materials.

\section{Metropolis Hastings via Classification}

The Metropolis–Hastings (MH) algorithm is one of the mainstays of Bayesian computation. The deployment of unbiased likelihood estimators within MH has shown great promise in models whose likelihoods are not available Beaumont, Zhang, and Balding (2002), Andrieu and Roberts (2009), and Andrieu and Vihola (2015). In the previous section, we have suggested how classification may be deployed to obtain estimates of likelihood ratios. This suggests a compelling question: Can we deploy these classification-based estimators within MH? This section explores this intriguing possibility and formalizes an MH variant that we further refer to as MHC, Metropolis Hastings via Classification.

Our objective is to simulate values from an (approximate) posterior distribution \( \Pi_n(\cdot | X^{(n)}) \) with a density \( \pi_n(\theta | X^{(n)}) \propto p_0^{(n)}(X^{(n)}) \pi(\theta) \) over \((\Theta, \mathcal{B})\) using the MH routine. Recall that
MH simulates a Markov chain according to the transition kernel $K(\theta, \theta') := \rho(\theta, \theta') q(\theta' | \theta) + \delta_0(\theta') \int_\Theta (1 - \rho(\theta, \hat{\theta}))q(\hat{\theta} | \theta')d\hat{\theta}$, where

$$\rho(\theta, \theta') := \min \left\{ \frac{p(\theta)(X^{(n)} | \theta')} {p(\theta')(X^{(n)} | \theta)} q(\theta' | \theta) \right\}$$

and where $q(\cdot | \theta)$ is a proposal density generating candidate values $\theta'$ for the next move.

It is often the case in practice that we cannot directly evaluate $p(\theta)(X^{(n)})$ but have access to its (unbiased) estimator (see, Doucet et al. 2014 for a recent overview). In Bayesian contexts, an unbiased likelihood estimator can be constructed using importance sampling Beaumont (2003) or particle filters Andrieu, Doucet, and Holenstein (2009) and Andrieu and Roberts (2009) via data augmentation through the introduction of auxiliary latent variables, say $X^{(m)}$. The perhaps simplest variant of such strategies is the Monte Carlo within Metropolis (MCWM) algorithm O’Neill et al. (2000) and Andrieu and Roberts (2009), which requires independently simulating $m$ replicates of the auxiliary data for each likelihood evaluation at each iteration. Other, so called pseudo-marginal Andrieu and Roberts (2009), variants have been suggested with latent data recycled from the previous iterations (Grouped Independent MH (GIMH) described in Beaumont, Zhang, and Balding 2002) or with correlated latent variables for the numerator and the denominator of the acceptance ratio Deligiannidis, Doucet, and Pitt (2018). In this work, we propose replacing $p(\theta)(X^{(n)})$ in the acceptance ratio (3.1) with the classification-based likelihood estimator (2.5) outlined in Section 2. This estimator, similarly as with pseudo-marginal (PM) methods, also relies on the introduction of latent variables $\tilde{X}^{(m)}$. However, unlike with related MH methods O’Neill et al. (2000) and Andrieu and Roberts (2009), we do not require an explicit hierarchical model where sampling from the conditional distribution of the latent data is feasible. Later in Section 5.2 we show an example of a generative model, where our approach fares favorably while the PM-style approaches are not straightforward, if at all possible. As we have seen earlier, our likelihood estimator can be rewritten in terms of the estimated discriminator as

$$\tilde{p}(\theta)(X^{(n)}) \propto \exp \left( \sum_{i=1}^n \log \left( 1 - \frac{\tilde{D}_{n,m}(X_i)}{\tilde{D}_{n,m}(\tilde{X}_i)} \right) \right).$$

The evaluation of $\tilde{p}(\theta)(X^{(n)})$ can be carried out by merely computing $\tilde{D}_{n,m}(X_i)$ where $\tilde{D}_{n,m}$ is a trained classifier distinguishing $X^{(n)}$ from $\tilde{X}^{(m)}$. Putting the pieces together, one can replace the intractable likelihood ratio in the acceptance probability (3.1) with

$$\rho_0(\theta, \theta') := \min \left\{ \frac{\tilde{p}(\theta)(X^{(n)} | \theta') q(\theta' | \theta)} {\tilde{p}(\theta')(X^{(n)} | \theta) q(\theta' | \theta')} \right\}.$$ 

Note that the proportionality constant in the likelihood expression (3.2) cancels out in (3.3), allowing $\rho_0(\theta, \theta')$ to be directly computable. We consider two variants. The first one, called a fixed generator design, assumes that the randomness of $\tilde{D}_{n,m}$ for each given $\theta$ and $X^{(n)}$, is determined by latent variables $\tilde{X}^{(m)}$ shared by all steps of the algorithm. This corresponds to the case when $m$ auxiliary data points $\tilde{X}^{(m)}_i = \{\tilde{X}^{(m)}_i\}_{i=1}^m$ are obtained through a deterministic mapping $\tilde{X}^{(m)} = \tilde{T}_n(X)$ for some $\tilde{T}_i \sim \tilde{P}$ that are not changed throughout the algorithm. The second version, called a random generator design, assumes that the underlying latent variables $\tilde{X}^{(m)} = \{\tilde{X}^{(m)}_i\}_{i=1}^m$ are refreshed at each step. While the difference between these two versions is somewhat subtle, we will see important bias-variance implications (discussed in more detail below). While technically our MHC sampling procedure follows the footsteps of a standard MH algorithm, we still find it useful to summarize the computations in an algorithm box (see Table 1).

### 3.1. Fixed Generator MHC

We inquire whether and how the likelihood approximation affects the stationary distribution of the resulting Markov chain. Due to the exponential tilt $e^{\rho_0}(X^{(n)})$ in the likelihood approximation (2.5), Algorithm 1 (Table 1) does not yield the correct posterior $\pi_n(\theta | X^{(n)})$ at its steady state. Indeed, under standard assumptions (see sec. 7.3.1 of Robert and Casella 2004), the stationary distribution of the Markov chain, conditional on $X^{(n)}$, writes as (see, e.g., Theorem 7.2 in Robert and Casella 2004)

$$\pi^*(\theta | X^{(n)}) = \frac{p(\theta)(X^{(n)}) \times e^{\rho_0}(X^{(n)}) \times \pi(\theta)} {\int_{\Theta} p(\theta)(X^{(n)}) \times e^{\rho_0}(X^{(n)}) \times \pi(\theta)d\theta}.$$ 

We do not view this property as unsurmountable. Other approximate MH algorithms (e.g., the MCWM method) may also not yield $\pi_n(\theta | X^{(n)})$ as their stationary distribution, provided that it in fact exists Andrieu and Roberts (2009). However, the samples generated by Algorithm 1 will be distributed according an approximate posterior (3.4) whose statistical properties we describe in detail in Section 4. In Section 7, supplementary materials we further quantify the speed of MHC convergence in large samples under the assumption of asymptotic normality. As will be seen in Section 4, the exponential tilt induces certain bias where the pseudo-posterior (3.4) concentrates around a projection of the true parameter $\theta_0$. Despite the bias, the limiting curvature of the approximate posterior can be shown to match the limiting curvature of the actual posterior (under differentiability assumptions in Section 4.1). The random generator version, introduced in the next section, works the other way around. Under some assumptions, it can lead to a correct location (no bias) but, possibly, at the expense of an enlarged variance.

### 3.2. Random Generator MHC

The random generator variant proceeds as Algorithm 1 but refreshes $\tilde{X}^{(m)} \sim \tilde{P}$ at each step before computing the acceptance ratio. We denote the density associated with $\tilde{P}$ by $\tilde{P}$. For simplicity, we have dropped the subscript $m$ in $\tilde{X}^{(m)}$ while describing the algorithm in Table 1. The acceptance probability now also involves $\tilde{X}$ and writes as

$$\tilde{\rho}_0(\theta, \tilde{X}; \tilde{X}^{(m)}) := \min \left\{ \frac{\tilde{p}(\theta)(X^{(n)} | \theta') \tilde{p}(\tilde{X}^{(m)} \mid \tilde{X}) q(\theta' | \theta) \tilde{D}_{n,m}(\tilde{X})} {\tilde{p}(\theta')(X^{(n)} | \theta) \tilde{p}(\tilde{X}^{(m)} | \tilde{X}) q(\theta' | \theta) \tilde{D}_{n,m}(X)} \right\}.$$ 

To glean more insights into this variant, it is helpful to regard $(\theta^{(l)}, \tilde{X}^{(l)})$ jointly as a Markov chain with an augmented
propose density \( q(\theta', \tilde{X}' | \theta, \tilde{X}) = q(\theta' | \theta) \tilde{q}(\tilde{X}' | \tilde{X}) \) where \( \tilde{q}(\tilde{X}' | \tilde{X}) \) possibly depends on \( \tilde{X} \). In order to make the dependence on \( \tilde{X} \) in \( u_0(X^{(n)}) \) more transparent, we will denote \( \tilde{u}_0(X^{(n)}) \) in (3.8) can average out. While \( u_0(X^{(n)}) \) in (2.6) is fixed in \( \tilde{X} \) (creating a nonvanishing bias term), \( \tilde{u}_0(X^{(n)}) \) in (3.8) can average out to 0 (depending on \( \tilde{q}(\cdot | \cdot) \)), erasing the bias and yielding the actual posterior as the stationary distribution.

### Table 2. Bias correction with Algorithm 3.

| Algorithm 3: Bias Correction |
|-------------------------------|
| (1) Generate a sample \( \{\theta_1^{(t)}\}_{t=1}^T \) using Algorithm 1 |
| (2) Generate a sample \( \{\theta_2^{(t)}\}_{t=1}^T \) using Algorithm 2 |
| (3) Debias \( \{\theta_1^{(t)}\} \) using \( \{\tilde{u}_0^{(t)}\} \), that is, construct a sample \( \{\theta(t)\} \) by \( \theta(t) := \theta_1^{(t)} - \frac{1}{T} \sum_{t=1}^T \theta_1^{(t)} + \frac{1}{T} \sum_{t=1}^T \theta_2^{(t)} \). |

### 3.3. Debiasing

Algorithm 1 and 2 can be combined to produce a more realistic representation of the true posterior. We mentioned that Algorithm 1, under the differentiability assumptions, has the same asymptotic curvature (as \( n \to \infty \)) as the actual posterior but has a nonvanishing shift. Algorithm 2, on the other hand, has a reduced bias due to the averaging aspect in (3.8). We can thus diminish the bias of the fixed generator design by shifting the location toward the mean of samples obtained with the random generator. This leads to a hybrid procedure summarized in Table 2. While Algorithms 1 and 2 can be deployed as a standalone, the de-biasing variant might increase the quality of the samples. Note that if \( u_0(X^{(n)}) = 0 \), Algorithm 2 will be exact, yielding the actual posterior as its stationary distribution. If inexact, in Section 4.1 we provide sufficient conditions under which Algorithm 3 yields samples from an object which, at least, has the same limit as the actual posterior.

### 4. Theory for MHC

We now shift attention from the computational aspects of MHC to its potential as a statistical inference procedure. To understand the qualitative properties of the MHC scheme, we provide an asymptotic study of its stationary distribution (convergence rates in Section 4.2 and asymptotic normality in Section 8, supplementary materials), drawing upon its connections to empirical Bayes methods (Section 4.2.1) and Bayesian misspecification (Section 4.2.2). Before delving into the stationary distribution, however, we first derive rates of convergence for the posterior.
residual $u_0(X^{(m)})$ in (2.6) which plays a fundamental role. For additional theory showing fast mixing of our Markov chains (i.e., polynomial mixing times) see Section 7 in the supplementary materials.

### 4.1. Convergence of the Posterior Residual

We denote the sample objective function in (2.2) with $M_{n,m}^\theta(D) := \mathbb{P}_n \log D + M_{n,m}^\theta \log(1 - D)$, where we employed the operator notation for expectation, for example, $\mathbb{P}_n f = \frac{1}{m} \sum_{i=1}^m f(X_i)$ and $\mathbb{P}_m^\theta f = \frac{1}{m} \sum_{i=1}^m f(X_i^\theta)$ (see the notation Section 1 in the supplementary materials for further details). Throughout this section, we will use a simplified notation $u_0$ instead of $u_0(X^{(m)})$ and similarly for $\hat{P}_{\theta}$ and $\hat{P}_{\theta}^{(m)}$. We denote by $P$ the probability measure that encompasses all randomness, for example, as $O_P(1)$. The estimated classifier is seen to satisfy

$$\hat{D}_{n,m}^\theta := \max_{D \in D_n} \mathbb{M}_{n,m}^\theta(D)$$

where $D_n$ constitutes a sieve of classifiers that expands with the sample size and that is not too rich (as measured by the bracketing entropy $N_D((\epsilon, F, d))$). In practice, the estimator $\hat{D}_{n,m}^\theta$ can be obtained by deploying a variety of classifiers ranging from logistic regression to deep learning (see Assumption 3 in Kaji, Manresa, and Pouliot (2020) for a sieve construction using neural network classifiers). The discrepancy between two classifiers will be measured by a Bellinger-type distance (see Kaji, Manresa, and Pouliot (2020) and Patilea (2008) for more discussion) $d_0(D_1, D_2) := \sqrt{h_0(D_1, D_2)^2} + h_0(1 - D_1 - D_2)^2$,

$$h_0(D_1, D_2) = \sqrt{(P_0 + P_0)(\sqrt{D_1 - 1} - \sqrt{D_2 - 1})^2}.$$

The rate of convergence of the Classifier was previously established by Kaji, Manresa, and Pouliot (2020) under assumptions reviewed below. In the following, we denote with $D_{n,\delta}^\theta := \{D \in D_n : d_0(D, D_{\delta}) \leq \delta\}$ the neighborhood of the oracle classifier within the sieve.

**Assumption 1.** Assume that $n/m$ converges and that an estimator $\hat{D}_{n,m}^\theta$ exists that satisfies $\mathbb{M}_{n,m}^\theta(\hat{D}_{n,m}^\theta) \geq \mathbb{M}_{n,m}^\theta(D_\delta) - O_P(\delta^2)$ for a nonnegative sequence $\delta_n$. Moreover, assume that the bracketing entropy integral satisfies $I_1(\delta_n, D_{n,\delta}^\theta, d_0) \lesssim \delta_n^2 \sqrt{n}$ and that there exists $\alpha < 2$ such that $I_1(\delta, D_{n,\delta}^\theta, d_0)/\delta^{\alpha}$ has a majorant decreasing in $\delta$.

The assumption requires that the synthetic sample size $m$ is at least as large as the actual sample size $n$, including the case when $n/m$ converges to 0. The second assumption requires that the training algorithm for the discriminator can find a sufficiently good approximate maximizer. The third assumption requires that the entropy of the sieve is not too large in order to avoid overfitting. For example, the bracketing entropy of a neural network sieve was shown to be bounded (Kaji, Manresa, and Pouliot 2020, Lemma 2).

Under **Assumption 1**, for a given $\theta \in \Theta$, Kaji, Manresa, and Pouliot (2020) conclude (see their Theorem 1) the following convergence rate result for the classifier: $d_0(\hat{D}_{n,m}^\theta, D_\theta) = O_P(\delta_n)$. While Kaji, Manresa, and Pouliot (2020) focused mainly on the convergence of $\hat{D}_{n,m}^\theta$, here we move the investigation further by establishing the rate of convergence of $u_0(\cdot)/n$ as well as its limiting shape. To this end, we assume the following support compatibility assumption, a refinement of the bounded likelihood ratio condition in nonparametric maximum likelihood (Theorem 3.4.4 in van der Vaart and Wellner 1996; Lemma 8.7 in Ghosal, Ghosh, and van der Vaart 2000).

**Assumption 2.** There exists $M > 0$ such that for every $\theta \in \Theta$, $P_{\theta}(P_{\theta}/P_{\theta})$ and $P_{\theta}(P_{\theta}/P_{\theta})^2$ are bounded by $M$ and

$$\sup_{D \in D_{\delta}^\theta,n} P_{\theta_0} \left( \frac{D_0}{D} \left| \frac{D_0}{D} \geq \frac{25}{16} \right. \right) < M,$$

$$\sup_{D \in D_{\delta}^\theta,n} P_{\theta_0} \left( \frac{1 - D_0}{1 - D} \left| \frac{1 - D_0}{1 - D} \geq \frac{25}{16} \right. \right) < M$$

for $\delta_n$ in **Assumption 1**. The brackets in **Assumption 1** can be taken so that $P_{\theta_0}(\sqrt{\frac{1}{\theta_0} - 1}) = o(d_0(u, \ell))$. For the cross-entropy loss, it is essential to control the tail behavior of the discriminator. **Assumption 2** restricts the tail of the discriminator so that the residual of the cross-entropy can be bounded with the bracketing entropy. For example, for a logistic discriminator, the tail of $D$ is proportional to an exponential function $e^{-x^2}$ for some $\beta$. Therefore, if $P_0$ has an exponential tail and $D_{n,\delta}^\theta$ gives a compact support for $\beta$, **Assumption 2** is satisfied. By analogy, we see that **Assumption 2** is reasonable for neural network discriminators that use sigmoid activation functions.

The following Theorem will be crucial for understanding theoretical properties of our MHC sampling algorithm, where the rate of convergence of $u_0(\cdot)/n$ will be seen to affect the rate of convergence of the stationary distribution of our Markov chains.

**Theorem 4.1.** Let Assumptions 1 and 2 hold for a given $\theta \in \Theta$, then

$$u_0/n = \mathbb{P}_n \left( \log \frac{1 - \hat{D}_{n,m}^\theta}{1 - D_\theta} - \log \frac{\hat{D}_{n,m}^\theta}{D_\theta} \right) = O_P(\delta_n).$$

**Proof.** Section 2 in the supplementary materials.

One seemingly pessimistic conclusion from **Theorem 4.1** is that $u_0(\cdot)$ does not vanish. Kaji, Manresa, and Pouliot (2020) shows that if the true likelihood ratio has a low-dimensional representation and an appropriate neural network is used for the discriminator, the rate $\delta_n$ depends only on the underlying dimension and not on the original dimension of $X_i$. In spite of the nonvanishing tilting term $u_0(X^{(m)})$, it turns out that Algorithm 1 can be refined (de-biased) to produce reasonable samples as long as $\hat{D}_{n,m}^\theta$ estimates the score well (see Section 3.3). In the sequel, we show quadratic approximability for $u_0$ at a much faster rate than **Theorem 4.1** when the model and the classifier are differentiable in some suitable sense.

**Assumption 3 (Differentiability of $p_\theta$).** There exists $\theta_0 \in \Theta \subset \mathbb{R}^d$ such that $P_\theta = P_{\theta_0}$. The model $\{p_\theta\}$ is differentiable in quadratic...
mean at \( \theta_0 \), that is, there exists a measurable function \( \hat{\theta}_0 : \mathcal{X} \to \mathbb{R}^d \) such that

\[
\int \left[ \sqrt{p_{\theta_0 + h} - \sqrt{p_{\theta_0}} - \frac{1}{2}h' \hat{\ell}_{\theta_0} \sqrt{p_{\theta_0}}} \right]^2 = O(||h||^2).
\]

This is a classical assumption (see, e.g., sec. 5.5 of van der Vaart 1998) which implies local asymptotic normality. Going back to (2.5), we write \( p_0(X^{(n)}) = \prod_{i=1}^n p_0(X_i) \), where

\[
\hat{p}_0 = p_0 - \frac{\hat{\ell}_{\theta_0}}{\ell_{\theta_0}}
\]

is an estimator of \( p_0 \) that is possibly unscaled so that \( \int \hat{p}_0 \) may not be one. The scaling constant will be denoted by \( c_0 := \int \hat{p}_0 \).

In general, \( \hat{p}_0 \) is not observable since \( p_{\theta_0} \) is not available. From (2.6), we can see that \( u_0 = n \pi_0 \log \frac{1 - \ell_{\theta_0}}{\ell_{\theta_0}} - n \pi_0 \log \frac{1 - \hat{\ell}_{\theta_0}}{\ell_{\theta_0}} = n \pi_0 \log \frac{\hat{h}}{\pi_0} - n \pi_0 \log \frac{\hat{p}_0}{\pi_0} \) and, under Assumption 3, (van der Vaart 1998, Theorem 7.2) derives convergence of the first term under the assumption below, we derive convergence of the first term under the a similar assumption.

**Assumption 4 (Differentiability of \( \hat{p}_0 \)).**

1. The estimator \( \hat{p}_0 \) is differentiable in quadratic mean in probability at \( \theta_0 \) with a cubic rate, which we define as \( \hat{p}_0 \hat{\ell}_{\theta_0} \hat{\ell}_{\theta_0}' \to_p I_{\theta_0} \) and

\[
(P_{\theta_0} + \hat{p}_0)(\sqrt{\frac{\hat{p}_0}{p_{\theta_0}} - 1 - \frac{1}{2}h' \hat{\ell}_{\theta_0}})^2 = O_p(||h||^3),
\]

where \( \hat{\ell}_{\theta_0} : \mathcal{X} \to \mathbb{R}^d \) is the score function in Assumption 3.

2. Dependence of \( \pi_0 \) and \( \hat{p}_0 \) is asymptotically ignorable in the sense that for every compact \( K \subset \mathbb{R}^d \), in outer probability,

\[
\sup_{h \in K} \left| n(\pi_0 - P_{\theta_0})(\sqrt{\frac{\hat{p}_0 + h}{\hat{p}_0} - 1 - \frac{1}{2}h' \hat{\ell}_{\theta_0}}) \right| \to 0,
\]

\[
\sup_{h \in K} \left| n(\pi_0 - P_{\theta_0})(\sqrt{\frac{\hat{p}_0 + h}{\pi_0} - 1})^2 \right| \to 0.
\]

3. The scaling factor is asymptotically linear in the sense that there exists a sequence of \( \mathbb{R}^d \)-valued random variables \( \hat{c}_{n,\theta_0} \) such that for every compact \( K \subset \mathbb{R}^d \), in outer probability,

\[
\sup_{h \in K} |n(\hat{c}_{n,\theta_0} + h/\sqrt{n} - \hat{c}_{\theta_0}) - \sqrt{n}h' \hat{\ell}_{\theta_0,\theta_0}| \to 0.
\]

**Assumption 4 (1) requires that \( \hat{p}_0 \) estimates the score well and is smoother than once differentiable. If \( \hat{p}_0 \) is twice differentiable in \( \theta_0 \), then it holds with \( O_p(||h||^4) \). Assumption 4 (2) requires that the dependence of \( \pi_0 \) and \( \hat{p}_0 \) be ignored asymptotically. If \( \pi_0 \) and \( \hat{p}_0 \) were independent, it would follow from Chebyshev’s or Markov’s inequality. Assumption 4 (3) requires that the quadratic curvature of the scaling constant vanishes asymptotically. In general, Assumption 4 is not verifiable since the likelihood is not available. To develop intuition behind this assumption, we verify that it holds for a toy normal location-scale model example in Section 6 in the supplementary materials. With Assumption 4, the estimated log-likelihood asymptotes to a quadratic function that has the oracle curvature but a different center.



\[\text{Theorem 4.2. Let } p_0 \text{ and } \hat{p}_0 \text{ satisfy Assumptions 3 and 4 and } \int \sqrt{p_0 - \hat{p}_0}^2 = O_p(\delta_n^3) \text{ for some } \delta_n = o(n^{-1/4}). \text{ Then, for every compact } K \subset \mathbb{R}^d, \text{ in outer probability,}
\]

\[
\sup_{h \in K} \left| n \pi_0 \hat{p}_0 + h/\sqrt{n} - \hat{p}_0 \right| \to 0
\]

**Proof.** Section 3 in the supplementary materials.

**Remark 1.** Recall that the true log-likelihood ratio locally approaches a quadratic curve

\[
\hat{h} = \sqrt{n}h(\hat{\ell}_{\theta_0}, \hat{\ell}_{\theta_0}) \quad \text{in (4.2) shifts the center of the quadratic curve but not the curvature.}
\]

One important implication of Theorem 4.2 is linearity of \( u_0 \).

**Corollary 4.3 (Linear \( u_0 \)).** Under assumptions of Theorem 4.2 we have

\[
u_{\theta_0+h/\sqrt{n}} - u_{\theta_0} = \hat{h}\sqrt{n}(\hat{\ell}_{\theta_0} - \hat{p}_0) + o_p(1).
\]

**Proof.** Follows from Van der Vaart (1998, Theorem 7.2) and Theorem 4.2.

We revisit linearity of \( u_0 \) later in Section 4.2.2 (Example 1, supplementary materials) as one of the sufficient conditions for the Bernstein-von Mises theorem. Corollary 4.3 has a very important consequence regarding the limiting shape of the stationary distribution \( \pi_n(\theta | X^{(n)}) \) for Algorithm 1 defined in (3.4). It shows that \( \pi_n(\theta | X^{(n)}) \) approaches a biased normal distribution with the same variance as the true posterior. In addition, we have seen in Section 3.2 that the stationary distribution \( \pi_n(\theta | X^{(n)}) \) of Algorithm 1 defined in (3.7) is averaged over the bias. Therefore, if \( \mathbb{E}[\hat{\ell}_{\theta_0} - \hat{p}_0 \hat{\ell}_{\theta_0} | X^{(n)}] = 0 \), where the expectation is taken over the latent data \( X^{(n)} \), then the stationary distribution of Algorithm 3 (in Table 2) converges to the correct normal posterior, that is, it has the same limit as the actual posterior \( \pi_n(\theta | X^{(n)}) \). Theorem 4.2 thus provides a theoretical justification for debiasing suggested in Section 3.3.

### 4.2. Posterior Concentration Rates

Having quantified the convergence rate of the posterior residual \( u_0(\theta) \) in Theorem 4.1, we are now ready to explore the convergence rate of the entire stationary distribution without necessarily imposing differentiability assumptions.

#### 4.2.1. Empirical Bayes Lens

Recall that the MHC sampler does not reach \( \pi_n(\theta | X^{(n)}) \) in steady state. Recall that the stationary distribution (using the fixed generator) takes the form

\[
\Pi_n(B | X^{(n)}) = \int_B p_0(\theta) / p_n(\theta) \times e^{u_0} \times \pi(\theta) d\theta / \int_B p_0(\theta) / p_n(\theta) \times e^{u_0} \times \pi(\theta) d\theta.
\]

In the random design, we simply replace \( u_0 \) with \( \tilde{u}_0 \) defined in (3.8). Interestingly, (4.3) can be viewed as an actual posterior under a tilted prior with a density \( \pi^*(\theta) \sim e^{u_0} \pi(\theta) \).
This shifted prior depends on the data \(X^{(n)}\) (through \(u_0(X^{(n)})\)) and thereby (4.3) can be loosely regarded as an empirical Bayes (EB) posterior. While EB uses plug-in estimators of prior hyperparameters, here the data enters the prior in a less straightforward manner.

We first assess the quality of the posterior approximation (4.3) through its concentration rate around the true parameter value \(\theta_0\) using the traditional Hellinger semimetric \(d_n(\theta, \theta')\).

The rate depends on the interplay between the concentration of the actual posterior (4.4) \(\Pi_n(\theta | X^{(n)})\) and the rate at which the residual \(u_0(X^{(n)})\) in (2.6) diverges. Recall that the rate of \(u_0(\cdot)/n\) was established earlier in Theorem 4.1. The following Theorem uses assumptions on prior concentration around \(\theta_0\) using the typical Kullback–Leibler neighborhood \(B_n(\theta_0, \epsilon) = \{\theta \in \Theta : K(p^{(n)}_0, \theta_0, p^{(n)}_\theta) \leq n\epsilon^2, \frac{1}{n} \sum_{i=1}^n V_2(p_{\theta_0}(X_i), p_\theta(X_i)) \leq \epsilon^2\}\).

**Theorem 4.4.** Consider the pseudo-posterior distribution \(\Pi^*_n\) defined through (4.3). Suppose that the prior \(\Pi_n(\cdot)\) satisfies conditions (3.2) and (3.4) in Ghosal and van der Vaart (2007) for a sequence \(\epsilon_n \to 0\) such that \(n\epsilon_n^2 \to \infty\). In addition, let \(C_n\) be such that

\[
P_{\theta_0}^{(n)} \left( \sup_{\theta \in \Theta_0} |u_0(X^{(n)})/n| > C_n\epsilon_n^2 \right) = o(1) \tag{4.4}
\]

and assume that for sets \(\Theta_n \subset \Theta\) the prior satisfies

\[
\frac{\Pi_n(\Theta_n | \Theta)}{\Pi_n(B_n(\theta_0, \epsilon_n))} = o(e^{-2(1+C_n)n\epsilon_n^2}). \tag{4.5}
\]

Then we have, for any \(M_n \to \infty\) such that \(C_n = o(M_n)\),

\[
P_{\theta_0}^{(n)} \left[ \Pi_n(\theta : d_n(\theta, \theta_0) > M_n\epsilon_n | X^{(n)}) \right] = o(1) \quad \text{as} \quad n \to \infty.
\]

**Proof.** The proof is a minor modification of Theorem 4 in Ghosal and van der Vaart (2007) and is postponed until Section 4 in the supplementary materials.

Theorem 4.4 shows that the concentration rate of the posterior nearly matches the concentration rate of the original prior \(\epsilon_n\) (this is implied by conditions (3.2), (3.4) and a variant of (4.5) according to Theorem 4 of Ghosal and van der Vaart 2007) up to an inflation factor \(C_n\) which depends on the rate of \(u_0(X^{(n)})/n\). If \(\tilde{C}_n = O(1)\) in (4.4), the rate of the actual posterior and pseudo-posterior will be the same.

**Remark 2 (Random Generator).** Recall that the stationary distribution \(\pi^*_n(\theta | X^{(n)})\) of the random generator MHC version can be written as (4.3) where \(u_0\) is replaced with \(u_\theta\) from (3.8). Theorem 4.4 holds also for the random generator where \(\tilde{C}_n\) is obtained from (4.4) with \(\tilde{u}_0\) instead of \(u_0\). Due to the averaging aspect, we might expect this \(\tilde{C}_n\) to be smaller in the random generator design.

**Remark 3 (Marginal Reference Distribution).** Theorem 4.4 holds also for the marginal contrastive learning Metropolis–Hastings approach proposed in Heermons, Begy, and Louppe (2020).

Indeed, defining \(u_\theta\) in terms of the discriminator \(D^\theta_n(X) = p(X)\pi(\theta) / [p(X)\pi(\theta) + p_n(X)\pi(\theta)]\), the same conclusion holds for the marginal approach under the assumption in (4.4).

**Theorem 4.4** describes the behavior of the pseudo-posterior around the truth \(\theta_0\). We learned that the rate is artificially inflated due a bias inflicted by the likelihood approximation, where \(\Pi^*_n(\cdot | X^{(n)})\) may not shrink around \(\theta_0\) when \(\epsilon_n\) is faster than the rate \(\delta_n\) established in Theorem 4.1. This suggest that the truth may not be the most natural centering point for the posterior to concentrate around. A perhaps more transparent approach is to consider a different (data-dependent) centering which will allow for a more honest reflection of the contraction speed devoid of any implicit bias. We look into model misspecification for guidance about reasonable centering points.

### 4.2.2. Model Misspecification Lens

In Section 4.2.1, we reframed the stationary distribution (3.4) as an empirical Bayes posterior by absorbing the term \(e^{u_0(X^{(n)})}\) inside the prior. This section pursues a different approach, absorbing \(e^{u_0(X^{(n)})}\) inside the likelihood instead. This leads a misspecified model \(\tilde{P}^*_\theta^{(n)}\) prescribed by the following likelihood function

\[
\tilde{P}^*_\theta^{(n)}(X^{(n)}) = \frac{P^{(n)}_\theta(X^{(n)})e^{u_0(X^{(n)})}}{C_\theta} \quad \text{where}
\]

\[
C_\theta = \int_X P^{(n)}_\theta(X^{(n)})e^{u_0(X^{(n)})}dX^{(n)}. \tag{4.6}
\]

Defining \(\tilde{\pi}(\theta) \propto \pi(\theta)C_\theta\), we can rewrite (3.4) as a posterior density under a misspecified likelihood and the modified prior \(\tilde{\pi}(\theta)\) as

\[
\tilde{\pi}^*_n(\theta | X^{(n)}) = \frac{P^{(n)}_\theta(X^{(n)})\tilde{\pi}(\theta)}{\int_\Theta P^{(n)}_\theta(X^{(n)})\tilde{\pi}(\theta)d\theta}. \tag{4.7}
\]

Since the model \(\tilde{P}^*_\theta^{(n)}\) is misspecified (i.e., \(\tilde{P}^*_\theta^{(n)}\) is not of the same form as \(\tilde{P}^{(n)}\)) due to the fact that \(D_n, m\) departs from the oracle discriminator, the posterior will concentrate around the point \(\theta^*\) defined as

\[
\theta^* = \arg\min_{\theta \in \Theta} -P_{\theta_0}^{(n)} \log(\tilde{P}_{\theta_0}^{(n)}/P^{(n)_{\theta^*}}) \tag{4.8}
\]

which corresponds to the element \(\tilde{P}^{(n)}_{\theta^*}\) in \(\tilde{P}^{(n)}\) that is closest to \(\tilde{P}_{\theta_0}^{(n)}\) in the KL sense Kleijn and van der Vaart (2006). Unlike in the iid data case studied, for example, in Kleijn and van der Vaart (2006) and De Blasi and Walker (2013), our likelihood (4.6) is not an independent product due to the none separability of the function \(u_\theta(X^{(n)})\). Theorem 5.1 (Section 5 in the supplementary materials) quantifies concentration in terms of a KL neighborhoods around \(\tilde{P}_{\theta^*}^{(n)}\). Beyond the speed of posterior concentration, we provide sufficient conditions for the stationary distribution to converge to a Gaussian distribution (see Section 8 in the supplementary materials).
5. MHC in Action

To whet reader’s appetite, we present MHC performance demonstrations in two examples which we found challenging for pseudo-marginal (PM) approaches and ABC. The first one (the CIR model) exemplifies data arising as discretizations of continuous-time processes for which likelihood inference can be problematic Sorensen (2004). We show that, compared with MCWM, MHC is not only far more straightforward to implement but also more scalable. The second demonstration involves a generative model (Lotka–Volterra) for which no explicit hierarchical model exists, precluding from straightforward application of MH methods Beaumont (2003). We thus compare MHC with ABC, showing that ABC techniques may fall short without a very informative prior and suitable summary statistics. More examples are shown in the supplementary materials where we show bias-variance tradeoffs between fixed/random generators on a toy normal location-scale model (Section 6) and the Ricker model Ricker (1954) (Section 10, supplementary materials). We also present a Bayesian model selection example (Section 11 in the supplementary materials) where ABC faces challenges.

5.1. The CIR Model

The CIR model Cox, Ingersoll, and Ross (1985) is prescribed by the stochastic differential equation

\[ dX_t = \beta(\alpha - X_t)dt + \sigma \sqrt{X_t}dW_t \]

where \( W_t \) is the Brownian motion, \( \alpha > 0 \) is a mean-reverting level, \( \beta > 0 \) is the speed of the process and \( \sigma > 0 \) is the volatility parameter. This process is an integral component of the Heston model Heston (1993) where it is deployed for modeling instantaneous variances. We want to perform Bayesian inference for the parameters \( \theta = (\alpha, \beta, \sigma)^\top \) of this continuous-time Markov process which is observed at discrete time points \( t_j = jT \) for \( j = 1, \ldots, T \). We will assume that there are \( n \) independent observed realizations \( x_i = (x_{i1}, \ldots, x_{iT})^\top \) of this discretized series for \( 1 \leq i \leq n \). It has been acknowledged that if the data are recorded at discrete times, parameteric inference using the likelihood can be difficult, partially due to the fact that the likelihood function is often not available Sorensen (2004). One possible Bayesian inferential platform for such problems is the MH algorithm where the likelihood function can be replaced with its approximation (e.g., using the analytical closed-form likelihood approximations Ait-Sahalia (2002). Stramer and Bognar (2011) perform a delicate Bayesian analysis of this model using the MCWM algorithm (defined in O’Neill et al. (2000) and discussed in Beaumont 2003; Andrieu and Roberts 2009) and the GIMH algorithm Beaumont (2003). Here, we compare MHC with MCWM, referring to Stramer and Bognar (2011) for a detailed analysis of the CIR model using GIMH.

One common approach in the literature for Bayesian estimation of diffusion models is to consider estimation on the basis of discrete measurements as a classic missing-data problem (see Roberts and Stramer (2001) for irreducible diffusion contexts). The idea is to introduce latent observations between every two consecutive data points. The time-step interval \([0, \Delta] \) is thus partitioned into \( M \) subintervals, each of length \( h = \Delta/M \).

The granularity \( M \) should be large enough so that the grid is sufficiently fine to yield more accurate likelihood approximations. With the introduction of latent variables, the pseudo-marginal approach naturally comes to mind as a possible inferential approach. The MCWM variant (described in Sec. 3 of Stramer and Bognar 2011) alternates between simulating \( \theta \), conditionally on the missing data blocks, say \( U \), and updating \( U \), given \( \theta \). We will be using the following enumeration for the missing data \( U = (u_{ik}^j) \): we have a replicade index \( 1 \leq i \leq n \), a discrete time index \( 0 \leq j \leq T \), an index of the intermittent auxiliary series \( 1 \leq m \leq M \) and an index \( 1 \leq k \leq N \) for the number of replications inside MCWM. Given \( \theta \), one can generate the missing data using the Modified Brownian Bridge (MBB) sampler Durham and Gallant (2002). Denote with \( X = [x_1, \ldots, x_n]^\top \) an \( n \times (T + 1) \) matrix of observations where \( x_0 = x_0 \) is the initial condition. The CIR model is an interesting testbed for both MCWM and our MHC approach, because the transition density is actually known (i.e., noncentral \( \chi^2 \) Cox, Ingersoll, and Ross 1985). We can thereby make comparisons with an exact algorithm which constructs the likelihood from the exact transition function.

The likelihood can be, however, stochastically approximated as

\[ \hat{p}_\theta(X) = \prod_{i=1}^n \prod_{j=0}^{T-1} \tilde{p}(x_{j+1} \mid x_j, \theta), \]

where

\[ \tilde{p}(x_{j+1} \mid x_j, \theta) = \frac{1}{N} \sum_{k=1}^N R_M(u_k^j), \]  

(5.1)

where \( u_k^j = (u_{k0}^j, \ldots, u_{kM}^j) \in \mathbb{R}^{M+1} \) is the \( k \)th sample of the Brownian bridge (described in (3) in Stramer and Bognar 2011) stretching from \( u_{k0}^j = x_j \) and \( u_{kM}^j = x_{j+1} \) and where

\[ R_M(u_k^j) = \frac{\Gamma^{M-1}_k \phi(u_{km+1}^j; u_{km}^j + h(\alpha - u_{km}^j) - \sigma\sqrt{h/2}, \mu, \sigma)}{\Gamma_k \phi(u_{km+1}^j + h(\alpha - u_{km}^j), \mu, \sigma)} \]

where \( \phi(x; \mu, \sigma) \) denotes the normal density with a mean \( \mu \) and a standard deviation \( \sigma \). Regarding the choice of \( M \) and \( N \), asymptotic arguments exist for choosing \( N = M^2 \) and Stramer and Bognar (2011) make thorough comparisons for various choices of \( M, N \) and also implement the ("exact" version having the correct stationary distribution) GIMH (see their Section 4) which recycles latent data \( U \). There are some delicate issues regarding dependency between \( \sigma \) and \( U \) in GIMH and we refer the reader to Stramer and Bognar (2011) for further details.

The true data consist of \( n = 100 \) samples generated using the package sde (using the function sde.sim with "rcCIR" initialized at \( x_0 = 0.1 \)) and discussed in Beaumont 2003; Andrieu and Roberts 2009 and the GIMH algorithm Beaumont (2003). Here, we compare MHC with MCWM, referring to Stramer and Bognar (2011) for a detailed analysis of the CIR model using GIMH.

5These values are close to parameter estimates found for FedFunds data analyzed in Stramer and Bognar (2011).
add summary statistics (mean, log-variance, auto-correlations at lag 1 and 2 as well as the first three principal components of \(X\)) yielding the total of 507 predictors (denoted with \(z_i\)). We consider both fixed and random generators where, for the fixed variant, we fix the random seed before generating fake data which essentially corresponds to having a deterministic generative mapping.

We compare the MCWM likelihood approximations obtained in MCWM (using (3.1)) with various choices \(N = M^2\) with the exact one using the explicit transition distribution (top panel in Figure 1). We can see that, even for a small value of \(N = 2\), the likelihood approximation seems to have a correct shape and is peaked close to the true values (marked by vertical dotted lines). The plots show likelihood slices along each parameter, one at a time, fixing the others at their true values. The approximation quality improves for \(M = 5\) and \(N = M^2\). The lower panel in Figure 1 portrays our classification-based log-likelihood (ratio) estimates \(\eta = \sum_{i=1}^n \log[(1 - \hat{D}(z_i))/\hat{D}(z_i)]\) for the fixed and random generators. The curves are nicely wrapped around the true values (perhaps even more so than for MCWM) with no visible systematic bias (even for the fixed generator). While, in the fixed case (solid lines), we would expect entirely smooth curves, recall that our classifier is based on cross-validation which introduces some randomness (thereby the wiggly estimate). The wiggyness can be alleviated by averaging over \(\text{nrep} \in \{1, 5\}\) many fake data replicates (dotted lines). The random generator (dashed lines) yields slightly more variable curves compared to the fixed design, as was expected. These plots indicate that MHC “pseudo-likelihood” contains relevant inferential information.

To implement the exact MH, MCWM and MHC (with \(\text{nrep} \in \{1, 5\}\)), we adopt the same prior settings as in Stramer and Bognar (2011), where \(\pi(\theta) = \mathbb{I}(0,1)(\alpha)\mathbb{I}(0,\infty)(\beta)\sigma^{-1}\mathbb{I}(0,\infty)(\sigma)\). We also use their random walk proposals.\(^6\) All three algorithms are initialized at the same perturbed truth and ran for 10,000 iterations with a burnin period 1000. Smoothed posterior densities obtained by simulation using the exact MH and MHC are in Figure 2 (random generator using \(\text{nrep} \in \{1, 5\}\) where fixed generator is portrayed in Figure 10 in the supplementary materials). The trace-plots of 10,000 iterations are depicted in Figures 11 and 12 in the supplementary materials, where we can see that the random generator variant yields smaller acceptance rates (especially for \(\sigma\)) which masks the fact that the random generator sampler generally yields more spread-out posterior approximations. Smoothing out the likelihood ratio by averaging over \(\text{nrep}\) repetitions reduces variance where fixed and random generators seem to yield qualitatively similar results in this example (this is why we have not used the debiasing variant here). Histograms (together with the demarkation of 95% credible set) are in Figure 14 in the supplementary materials. Compared with the smoothed densities obtained from MCWM (using \(N = M^2\) with \(M \in \{2, 5\}\) in Figure 3) we can see that MHC yields posterior reconstructions that are wrapped more closely around the true values. Increasing \(M\), MCWM yields posterior reconstructions that are getting closer to the actual posterior (not necessarily centered more narrowly around the truth). Recall, however, that MCWM generates Markov chains whose invariant distribution is not necessarily the exact posterior. The posterior summaries (means and 95% credible intervals) are reported in Table 2 (supplementary materials). Interestingly, both MCWM intervals for \(\sigma\) do not include the true value 0.07 and the MCWM computation is

\(^6\) With probability 2/3 propose a joint move \((\alpha^*, \beta^*)\) by generating \(\alpha^* \sim U(\alpha - 0.01, \alpha + 0.01)\) and \(\beta^* \sim U(\beta - 0.01, \beta + 0.01)\) and with probability 1/3 propose \(\sigma^* \sim U(\sigma - 0.01, \sigma + 0.01)\). To increase the acceptance rate of the exact MH algorithm, we change the window from 0.01 to 0.005.
Figure 2. Smoothed posterior densities obtained for the CIR model by simulation using exact MH and MHC using $n_{rep} = 1$ (green) and $n_{rep} = 5$ (blue). Vertical lines are the true values.

Figure 3. Smoothed posterior densities obtained by simulation using MCWM (with $N = M^2$) for $M = 2$ (MCWM1 green) and $M = 5$ (MCWM2 blue). Vertical lines are the true values.

5.2. Lotka–Volterra Model

The Lotka–Volterra (LV) predator–prey model Wilkinson (2011) describes population evolutions in ecosystems where predators interact with prey. The model is deterministically prescribed via a system of first-order nonlinear ordinary differential equations with four parameters $\theta = (\theta_1, \ldots, \theta_4)'$ controlling (a) the rate $r_1 = \theta_1 X_t Y_t$ of a predator being born, (b) the rate $r_2 = \theta_2 X_t$ of a predator dying, (c) the rate $r_3 = \theta_3 Y_t$ of a prey being born and (d) the rate $r_4 = \theta_4 X_t Y_t$ of a prey dying. Given the initial population sizes $X_0$ (predators) and $Y_0$ (prey) at time $t = 0$, the process can be simulated from exactly using the Gillespie algorithm Gillespie (1977). In particular, this algorithm samples times to an event from an exponential distribution (with a rate $\sum_{j=1}^{4} r_j$) and then picks one of the four reactions with probabilities proportional to their individual rates $r_j$. Despite being easy to sample from, the likelihood for this model is unavailable which makes this model a natural candidate for ABC Prangle (2017) and other likelihood-free methods Papamakarios and Murray (2016) and Meeds and Welling (2015). It is not entirely obvious, however, how to implement the pseudo-marginal approach since there is no explicit hierarchical model structure with a conditional likelihood, given latent data, which could be marginalized through simulation to obtain a likelihood estimate.

In our experiments, each simulation is started at $X_0 = 50$ and $Y_0 = 100$ simulated over 20 time units and recorded observations every 0.1 time units, resulting in a series of $T = 201$ observations each. We plot $n = 20$ time series realizations for three particular choices of $\theta$ in Figure 4 which differ in the second argument $\theta_2$ with larger values accentuating the cyclical behavior. Slight shifts in parameters result in (often) dramatically different trajectories. Typical behaviors include (a) predators quickly eating all the prey and then slowly decaying (as in Figure 4(b)), (b) predators quickly dying out and then the prey population sky-rocketing. For certain carefully tuned values $\theta$, the two populations exhibit oscillatory behavior. For example, in Figure 4(a) and (c) we can see how the value $\theta_2$ determines the frequency of the population renewal cycle. We rely on the ability of the discriminator to tell such different shapes apart. The real data ($n = 20$) is generated under the scenario (a) with $\theta^0 = (0.01, 0.5, 1, 0.01)'$.

ABC analyses of this model reported in the literature have relied on various summary statistics including the mean, log-variance, autocorrelation (at lag 1 and 2) of each series as well
as their cross-correlation Papamakarios and Murray (2016). These summary statistics seem to be able to capture the oscillatory behavior (at different frequencies) and distinguish it from exploding population growth (see Figure 19 in Section 13.3 of the supplementary materials). This creates hope that ABC based on these summary statistics has the capacity to provide a reliable posterior reconstruction. In a similar vein, we plotted the estimated log-likelihood ratio estimator ahead of the MCMC trains the log-likelihood ratio estimator (using the R package glmnet and randomForest) on \( m = n \) fake data observations \( \tilde{x}_i = (\tilde{X}_1^i, \ldots, \tilde{X}_m^i, \tilde{Y}_1^i, \ldots, \tilde{Y}_m^i)' \) for \( 1 \leq i \leq m \).

See heat-map plots of the estimated likelihood \( \eta \) as a function of \( (\theta_2, \theta_3)' \) (Figure 5(a)) and as a function of \( (\theta_1, \theta_2)' \) (Figure 5(b) for glmnet and Figure 5(c) for randomForest), keeping the remaining parameters at the truth. Figure 5(b) reveals a sharp spike (approximating a point-mass) around the true value at \( \theta_1 = \theta_2 = 0.01 \) in a otherwise vastly flat landscape. This peculiar likelihood property may require a very careful consideration of initializations and proposal densities for MH and the prior domain for ABC. The random forest classifier, however, did not yield as spiky likelihood estimators (Figure 5(c)), suggesting that it will be less sensitive to MHC initialization. We also inspected estimated log-likelihoods using \( (a) \) the fixed prior as for the ABC method. We use the random generator \( \texttt{withaproposalstandarddeviation}0.05 \) and deploy the same random forest classifier.

In order to facilitate ABC analysis, we have used an informative uniform prior \( \theta \sim U(\Xi) \) with a restricted domain \( \Xi = [0, 0.1] \times [0, 1] \times [0, 2] \times [0, 0.1] \) so that the procedure does not waste time sampling from unrealistic parameter values. These values were chosen based on a visual inspection of simulated evolutions, where we have seen only a limited range of values to yield periodic behavior. In a pilot ABC run, we rank \( M = 10,000 \) ABC samples based on \( r \) in an ascending manner and report the histogram of the first \( r = 100 \) samples (Figure 23 in the supplementary materials, the upper panel). We can see that ABC was able to narrow down the region of interest for \( (\theta_1, \theta_2) \), but is still largely uninformative about parameters \( (\theta_2, \theta_3) \) with histograms stretching from the boundaries of the prior domain. Given how narrow the range of likely parameter values is (according to Figure 5), the likelihood of encountering such values even under the restricted uniform prior is still quite negligible. We thereby tried many more ABC samples \( M = 100,000 \) which took 47.46 hr only to find out that the histograms (top \( r = 1000 \) samples) did not improve much (Figure 23 in the supplementary materials, the lower panel).

The hostile likelihood landscape will create problems not only for ABC but also for Metropolis–Hastings. Indeed, initializations that are too far from the likelihood domain may result in Markov chains wandering aimlessly in the vast plateaus for a long time. Rather than competing with ABC, a perhaps more productive strategy is to combine the strengths of both. We have thereby used the pilot ABC run (the closest 100 samples out of \( M = 10,000 \) which took roughly 4 hr) to obtain ABC approximated posterior means \( \tilde{\theta} = (0.015, 0.55, 1.31, 0.012)' \). We use these to initialize\(^8\) all MH procedures to accelerate convergence (i.e., prevent painfully long burn-in). To implement MHC, we define a Gaussian random walk proposal for log-parameter values with a proposal standard deviation 0.05 and deploy the same prior as for the ABC method. We use the random generator variable here, where the fixed one can be implemented (for example) by fixing the random seed prior generating the fake data. We compare our approach with the Classification Metropolis–Hastings of Pham, Nott, and Chaudhuri (2014) and the marginal

\[^8\]This is a valid concern for the glmnet classifier.
\[^9\]MHC with random forests did not seem as sensitive to initialization compared to glmnet.
Table 3. Posterior summaries using ABC1 ($M = 10,000$ and $r = 100$), ABC2 ($M = 100,000$ and $r = 1000$) and MH variants ($M = 10,000$ with burnin 1000).

| Method          | $\theta_0 = 0.01$ | $\theta_0 = 0.5$ | $\theta_3 = 1$ | $\theta_4 = 0.01$ | Time (hr) |
|-----------------|-------------------|-------------------|-----------------|-------------------|-----------|
| ABC1            | 0.015             | 0.003             | 0.038           | 1.315             | 0.012     |
| ABC2            | 0.016             | 0.003             | 0.042           | 1.063             | 0.01      |
| MHC (rf)        | 0.01              | 0.008             | 0.011           | 0.514             |           |
| MHC (glmnet)    | 0.01              | 0.009             | 0.015           | 0.490             |           |
| ALR MH ($m = 10,000$) | 0.006          | 0.002             | 0.012           | 0.477             | 0.006     |
| ALR MH ($m = 50,000$) | 0.008          | 0.005             | 0.013           | 0.527             | 0.008     |
| Classif MH ($m = 20$) | 0.01           | 0.008             | 0.012           | 0.5                | 0.01      |
| Classif MH ($m = 100$) | 0.01           | 0.009             | 0.011           | 0.501             | 0.015     |

NOTE: $\bar{\theta}$ is the posterior mean, $l$ and $u$ denote the lower and upper boundaries of 95% credible intervals. MHC variants are implemented with random forests and glmnet classifiers. ALR MH is the amortized likelihood ratio MH of Heermans, Begy, and Louppe (2020) (using random forests). Classif MH is the classifier MCMC of Pham, Nott, and Chaudhuri (2014) (using random forests). $m$ is the fake data sample size.

Figure 5. Lotka–Volterra model. Estimated log-likelihood for a grid of parameters.

This article develops an approximate Metropolis–Hastings (MH) posterior sampling method for when the likelihood is not tractable. By deploying a Generator and a Classifier (similarly as in Generative Adversarial Networks Goodfellow et al. 2014), likelihood ratio estimators are obtained which are then plugged into the MH sampling routine. One of the main distinguishing features of our work is that we consider two variants: (a) a fixed generator design yielding biased samples,
and (b) a random generator yielding more dispersed samples. Compared to related existing approaches Pham, Nott, and Chaudhuri (2014) and Heermans, Begy, and Louppe (2020), our approach uses observed data as the contrasting dataset. This ultimately poses limitations on the classifier when the sample size $n$ is small in which case the approaches Pham, Nott, and Chaudhuri (2014) and Heermans, Begy, and Louppe (2020) are more appropriate. We provide a thorough frequentist analysis of the stationary distribution including convergence rates and asymptotic normality. Under suitable differentiability assumptions, we conclude that correct shape and location can be recovered by deploying a debiasing combination of the fixed and random generator variants. We demonstrate a very satisfactory performance on nontrivial time series examples which render existing techniques (such as PM or ABC) less practical.

**Supplementary Materials**

Supplemental Materials include: (a) Proofs of main theorems (Section 2, 3 and 4), (b) explanations of the model-misspecification perspective on tilted likelihoods (Section 5 and 8), (c) mixing properties of the Markov chains (Section 7), (d) additional examples (Section 6, 10, 11, 12 and 13), (d) comparisons with alternatives (Section 9).

**Funding**

Tetsuya Kaji gratefully acknowledges the support from the Richard N. Rosett Faculty Fellowship and the Liew Family Faculty Fellowship at the University of Chicago Booth School of Business. Veronika Rockova gratefully acknowledges the support from James S. Kemper Faculty Scholarship and the National Science Foundation (DMS: 1944740).

**References**

Ait-Sahalia, Y. (2002), “Maximum Likelihood Estimation of Discretely Sampled Diffusions: A Closed-form Approximation Approach,” *Econometrica*, 70, 223–262. [2541]

Andrieu, C., Doucet, A., and Holenstein, R. (2009), “Particle Markov chain Monte Carlo Methods,” *Journal of the Royal Statistical Society*, Series B, 72, 1–33. [2536]

Andrieu, C., and Roberts, G. O. (2009), “The Pseudo-Marginal Approach for Efficient Monte Carlo Computations,” *The Annals of Statistics*, 37, 697–725. [2534,2535,2536,2541]

Andrieu, C., and Vihola, M. (2015), “Convergence Properties of Pseudo-Marginal Markov chain Monte Carlo Algorithms,” *The Annals of Applied Probability*, 25, 1030–1077. [2535]

Beaumont, M. A. (2003), “Estimation of Population Growth or Decline in Genetically Monitored Populations,” *Genetics*, 164, 1139–1160. [2534,2535,2541]

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002), “Approximate Bayesian Computation in Population Genetics,” *Genetics*, 162, 2025–2035. [2534,2535,2536]

Bernton, E., Jacob, P., Gerber, M., and Robert, C. (2019), “Approximate Bayesian Computation with the Wasserstein Distance,” *Journal of the Royal Statistical Society*, Series B, 81, 235–269. [2533]

Blum, M. (2010), “Approximate Bayesian Computation: A Nonparametric Perspective,” *Journal of the American Statistical Association*, 105, 1178–1187. [2533]

Blum, M., Nunes, M., Prangle, D., and Sisson, S. (2013), “A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation,” *Statistical Science*, 28, 189–208. [2533]

Clarke, G., Robert, C., Ryder, R., and Stoehr, J. (2021), “Componentwise Approximate Bayesian Computation via Gibbs-like Steps,” *Biometrika*, 108, 591–607. [2545]

Cox, J., Ingersoll, J., and Ross, S. (1985), “A Theory of the Term Structure of Interest Rates,” *Econometrica*, 53, 385–407. [2534,2541]

Cranmer, K., Pavez, J., and Louppe, G. (2016), “Approximating Likelihood Ratios with Calibrated Discriminative Classifiers,” arXiv, 1–30. [2534,2535]

De Blasi, P., and Walker, S. (2013), “Bayesian Estimation of the Discrepancy with Misspecified Parametric Models,” *Bayesian Analysis*, 8, 781–800. [2540]

Deligiannidis, G., Doucet, A., and Pitt, M. K. (2018), “The Correlated Pseudomarginal Method,” *Journal of the Royal Statistical Society*, Series B, 80, 839–870. [2534,2536]

Diggle, P. J., and Gratton, R. J. (1984), “Monte Carlo Methods of Inference for Implicit Statistical Models,” *Journal of the Royal Statistical Society*, Series B, 46, 193–212. [2533]

Doucet, A., Pitt, M., Deligiannidis, G., and Kohn, R. (2014), “Efficient Implementation of Markov chain Monte Carlo When Using an Unbiased Likelihood Estimator,” arXiv, 1–40. [2536]

Durham, G., and Gallant, A. (2002), “Numerical Techniques for Maximum Likelihood Estimation of Continuous-Time Diffusion Processes,” *Journal of Business & Economic Statistics*, 20, 297–338. [2541]

Durkan, C., Murray, I., and Papamakarios, G. (2020), “On Contrastive Learning for Likelihood-Free Inference,” *Proceedings of the 37th International Conference on Machine Learning*, 119, 297–338. [2533,2534,2535]

Fearnhead, P., and Prangle, D. (2011), “Constructing ABC Summary Statistics: Semi-automatic ABC,” *Nature Precedings*, 74, 1–10. [2533,2534]

Frazier, D., and Drovandi, C. (2021), “Robust Approximate Bayesian Inference with Synthetic Likelihood,” *Journal of Computational and Graphical Statistics*, 30, 958–976. [2533]

Frazier, D., Martin, G., Robert, C., and Rousseau, J. (2018), “Asymptotic Properties of Approximate Bayesian Computation,” *Biometrika*, 105, 593–607. [2534]

Frazier, D., Robert, C., and Rousseau, J. (2020), “Model Misspecification in Approximate Bayesian Computation: Consequences and Diagnostics,” *Journal of the Royal Statistical Society*, Series B, 82, 421–444. [2533]

Ghosal, S., Ghosh, J. K., and van der Vaart, A. W. (2000), “Convergence Rates of Posterior Distributions,” *The Annals of Statistics*, 28, 500–531. [2538]

Ghosal, S., and van der Vaart, A. W. (2007), “Convergence Rates of Posterior Distributions for Noniid Observations,” *The Annals of Statistics*, 35, 192–223. [2540]

Gillespie, D. (1977), “Exact Stochastic Simulation of Coupled Chemical Reactions,” *The Journal of Physical Chemistry*, 81, 2340–2361. [2534]

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., and Ozair, S. (2014), “Generative Adversarial Nets,” *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2, 2672–2680. [2534,2535,2545]

Gutmann, M., and Hyvärinen, A. (2012), “Noise-Contrastive Estimation of Unnormalized Statistical Models with Applications to Natural Image Statistics,” *Journal of Machine Learning Research*, 13, 307–361. [2534,2535]

Gutmann, M. U., and Corander, J. (2016), “Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models,” *Journal of Machine Learning Research*, 17, 4256–4302. [2533,2534]

Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. (2018), “Likelihood-Free Inference via Classification,” *Statistics and Computing*, 28, 411–425. [2534,2535,2536]

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning*, Springer Series in Statistics, New York: Springer. [2534,2535]

Heermans, J., Begy, V., and Louppe, G. (2020), “Likelihood-Free MCMC with Amortized Approximate Ratio Estimators,” *Proceedings of the 37th International Conference on Machine Learning*, 119, 15112–15117. [2534,2535,2544,2545,2546]

Heston, S. (1993), “A Closed-form Solution for Options with Stochastic Volatility with Applications to Bonds and Currency Options,” *Review of Financial Studies*, 6, 327–343. [2534,2541]

Jiang, B., Wu, T., Zheng, C., and Wong, W. (2017), “Learning Summary Statistic for Approximate Bayesian Computation via Deep Neural Network,” *Statistica Sinica*, 27, 1595–1618. [2533]

Kaji, T., Manresa, E., and Pouliot, G. (2020), “An Adversarial Approach to Structural Estimation,” arXiv. [2538]
Kleijn, B., and van der Vaart, A. (2006), “Misspecification in Infinite-Dimensional Bayesian Statistics,” *The Annals of Statistics*, 34, 837–877. [2540]

Marin, J., Pillai, N., Robert, C., and Rousseau, J. (2014), “Relevant Statistics for Bayesian Model Choice,” *Journal of the Royal Statistical Society*, Series B, 76, 833–859. [2533]

Meeds, T., and Welling, M. (2015), “Optimization Monte Carlo: Efficient and Embarrassingly Parallel Likelihood-Free Inference,” *Advances in Neural Information Processing Systems*, 28, 289–309. [2543]

Mohamed, S., and Lakshminarayanan, B. (2017), “Learning in Implicit Generative Models,” arXiv, 1–10. [2535]

O'Neill, P. D., Balding, D. J., Becker, N. G., Eerola, M., and Mollison, D. (2000), “Analyses of Infectious Disease Data from Household Outbreaks by Markov chain Monte Carlo Methods,” *Journal of the Royal Statistical Society*, Series C, 49, 517–542. [2534,2536,2541]

O’Ryan, C., Harley, E. H., Bruford, M. W., Beaumont, M., Wayne, R. K., and Cherry, M. I. (1998), “Microsatellite Analysis of Genetic Diversity in Fragmented South African Buffalo Populations,” *Animal Conservation*, 1, 85–94. [2534]

Papamakarios, G., and Murray, I. (2016), “Fast $\epsilon$-Free Inference of Simulation Models with Bayesian Conditional Density Estimation,” *Advances in Neural Information Processing Systems*, 29, 289–309. [2533,2535,2543,2544]

Robert, C. P., and Casella, G. (2004), *Monte Carlo Statistical Methods* (2nd ed.), New York: Springer. [2536]

Robert, C. P., Cornuet, J.-M., Marin, J.-M., and Pillai, N. S. (2011), “Lack of Confidence in Approximate Bayesian Computation Model Choice,” *Proceedings of the National Academy of Sciences*, 108, 15112–15117. [2533]

Ricker, W. E. (1954), “Stock and Recruitment,” *Journal of the Fisheries Board of Canada*, 11, 559–623. [2534,2541]

Ripley, B. D. (2004), *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press. [2534]

Robert, C. P., and Stramer, O. (2001), “On Inference for Partially Observed Nonlinear Diffusion Models Using the Metropolis-Hastings Algorithms,” *Biometrika*, 88, 603–621. [2541]

Sorensen, H. (2004), “Parametric Inference for Diffusion Processes Observed at Discrete Points in Time: A Survey,” *International Statistical Review*, 72, 337–354. [2541]

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012), *Density Ratio Estimation in Machine Learning*, Cambridge: Cambridge University Press. [2534]

Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. (1997), “Inferring Coalescence Times from DNA Sequence Data,” *Genetics*, 145, 505–518. [2533]

Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. (2021), “Likelihood-Free Inference by Ratio Estimation,” *Bayesian Analysis*, 1, 1–31. [2534,2535]

Tsvetkov, D., Hristov, L., and Angelova-Slavova, R. (2017), “On the Convergence of the Metropolis-Hastings Markov Chains,” *Serdica Mathematical Journal*, 43, 93–110. [2537]

Wilkinson, D. (2011), *Stochastic Modelling for Systems Biology* (2nd ed.), Boca Raton, FL: Chapman and Hall/CRC. [2543]

Wood, S. N. (2010), “Statistical Inference for Noisy Nonlinear Ecological Dynamic Systems,” *Nature*, 466, 1102–1104. [2533,2534]