Multi-fidelity Monte Carlo: a pseudo-marginal approach

Anonymous Author(s)
Affiliation
Address
email

Abstract

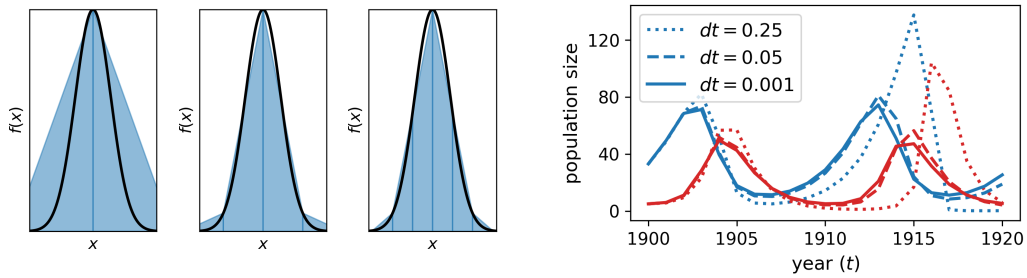
Markov chain Monte Carlo (MCMC) is an established approach for uncertainty quantification and propagation in scientific applications. A key challenge in applying MCMC to scientific domains is computation: the target density of interest is often a function of expensive computations, such as a high-fidelity physical simulation, an intractable integral, or a slowly-converging iterative algorithm. Thus, using an MCMC algorithms with an expensive target density becomes impractical, as these expensive computations need to be evaluated at each iteration of the algorithm. In practice, these computations often approximated via a cheaper, low-fidelity computation, leading to bias in the resulting target density. Multi-fidelity MCMC algorithms combine likelihoods of varying fidelities in order to obtain an approximate target density with lower computational cost. In this paper, we describe a class of asymptotically exact multi-fidelity MCMC algorithms for the setting where a sequence of likelihoods of increasing fidelity can be computed that approximates the high-fidelity likelihood. We take a pseudo-marginal MCMC approach for multi-fidelity inference that utilizes a low-fidelity unbiased estimator of the high-fidelity likelihood constructed via randomized truncation of a telescoping series of the low-fidelity sequence of models. Finally, we discuss and evaluate the proposed multi-fidelity MCMC approach on several applications, including log-Gaussian Cox process modeling, Bayesian ODE system identification, PDE-constrained optimization, and Gaussian process regression parameter inference.

1 Introduction

Simulation and computational modeling play a key role in science, engineering, economics, and many other areas. When they are high-quality and accurate, these models are important for scientific discovery, design, and data-driven decision making. However, the ability to accurately model complex physical phenomena often comes with a significant cost—many models involve expensive computations that then need to be evaluated repeatedly in, for instance, a sampling or optimization algorithm. Examples of model classes with expensive computations include intractable integrals or sums, expensive quantum simulations [39], expensive numerical simulations arising from partial differential equations (PDEs) [34], and large systems of ordinary equations (ODEs).

In many situations, one has the ability to trade off computational cost against *fidelity* or accuracy in the result. Such a tradeoff might arise from the choice of discretization or the number of basis functions when solving a PDE, or the number of quadrature points when estimating an integral. Sometimes it is possible to leverage lower-fidelity models to help accelerate high-quality solutions, e.g., by using multigrid methods [20] for spatial discretizations. More generally, *multi-fidelity* methods combine multiple models of varying cost and fidelity to accelerate computational algorithms and have been applied to solving inverse problems [10, 21, 34], trust region optimization [1, 4, 14, 28, 35], Bayesian optimization [7, 18, 24, 25, 37, 40], Bayesian quadrature [16, 41], and sequential learning [19, 32].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) Trapezoid rule with $2k$ trapezoids ($k = 1, 2, 3$)

(b) Lokta-Volterra ODE solutions, $dt = 1/k$

Figure 1: Examples of low-fidelity sequences of models. **(a)** Sequence trapezoid quadrature estimates I_k, where I_k is the trapzoid rule with $2k$ trapezoids. **(b)** Lokta-Volterra ODE solutions for prey $u(t)$ (blue) and predator $v(t)$ (red) using Euler's method with step size dt.

One critically important tool for scientific and engineering computation is Markov chain Monte Carlo (MCMC), which is widely used for uncertainty quantification, optimization, and integration. MCMC methods are recipes for constructing a Markov chain with some desired target distribution as the limiting distribution. One uses pseudo-random numbers to simulate transitions of the Markov chain in order to produce samples from the target distribution. However, MCMC often becomes impractical for high-fidelity models, where a single step of the Markov chain may, for instance, involve a numerical simulation that takes hours or days to complete. Multi-fidelity methods for MCMC focus on constructing Markov chain transition operators that are sometimes able to use inexpensive low-fidelity evaluations instead of expensive high-fidelity evaluations. The goal is to increase the effective number of samples generated by the algorithm, given a constrained computational budget. A large focus of the multi-fidelity MCMC literature is on two-stage Metropolis-Hastings (MH) methods [9, 13], which use a single low-fidelity model for early rejection of a proposed sample, thereby often short-circuiting the evaluation of the expensive, high-fidelity model.

However, there are several limitations of two-stage multi-fidelity Monte Carlo. First, in many applications, a *hierarchy* of cheaper, low-fidelity models is available; for instance, in the case of integration, k-point quadrature estimates form a hierarchy of low-fidelity models, and in the case of a PDE, varying the discretization. Thus, the two-stage approach does not fully utilize the availability of a hierarchy of fidelities and may be more suitable for settings where the high- and low-fidelity models are not hierarchically related, e.g., semi-empirical methods vs. Hartree-Fock in computational chemistry. In addition, in such applications, there is often a limiting model of interest, such as a continuous function that the low-fidelity discretizations approximate. Two-stage MCMC does not asymptotically sample from this limiting target density and will at best sample from an approximation of the biased, high-fidelity posterior. Finally, the two-stage method is unnatural to generalize to more sophisticated MCMC algorithms such as slice samplers and Hamiltonian Monte Carlo.

We propose a class of multi-fidelity MCMC methods designed for applications with a hierarchy of low-fidelity models available. More specifically, we assume access to a sequence of low-fidelity models that converge to a "perfect-fidelity" model in the limit. Within an MCMC algorithm, we can approximate perfect-fidelity target density with an unbiased estimator constructed from a randomized truncation of the infinite telescoping series of low-fidelity target densities. This class of multi-fidelity MCMC is an example of a pseudo-marginal MCMC algorithm—the unbiased estimator essentially guarantees that the algorithm is asymptotically exact in that it eventually generates samples from the perfect-fidelity target distribution. Our approach introduces the fidelity of a model as an auxiliary random variable that is evolved separately from the target variable according to its own conditional target distribution; this technique can be used in conjunction with any suitable MCMC update that leaves the conditional update for the target variable of interest invariant, such as MH, slice sampling, elliptical slice sampling, or Hamiltonian Monte Carlo. We apply pseudo-marginal multi-fidelity MCMC to several problems, including a log-Gaussian Cox process, Bayesian ODE system identification, PDE-constrained optimization, and Gaussian process parameter inference.

Related work. Multi-fidelity MCMC methods are typically applied in a two-stage scenario where the goal is to reduce the computational cost of using a single expensive high-fidelity model by using a cheap low-fidelity model as a low-pass filter for a delayed acceptance/rejection algorithm [9, 10, 13].

Higdon et al. [21] propose coupling a high-fidelity Markov chain with a low-fidelity Markov chain via a product chain. We refer to Peherstorfer et al. [33] for a survey on two-stage multi-fidelity MCMC methods. In constrast, our approach aims to sample from a "perfect-fidelity" target density while reducing computational cost; in our setting, a two-stage MCMC algorithm would be biased with respect to this target density. Dodwell et al. [12] consider consider a hierarchy of multi-fidelity models in a similar sense to our setting and propose coupling multi-level Monte Carlo with an M-H algorithm; while similar in motivation, this method only consider a finite number of levels and is unable to achieve the same asymptotic guarantees as our method.

Our approach applies pseudo-marginal MCMC to multi-fidelity problems. There is a rich literature developing pseudo-marginal MCMC methods [2, 6] for so-called "doubly-intractable" likelihoods, likelihoods when the likelihood is intractable to evaluate. There are several appraoches in the pseudo-marginal MCMC literature are particular relevant to our work. The first is are PM-MCMC approaches introduced by Lyne et al. [27], which describes a class of pseudo-marginal MH methods that use Russian roulette estimators to obtain unbiased estimators of the likelihood. However, this method samples the variable of interest jointly with the auxillary randomness, which can lead to the sticking.

Alternatively, several methods have considered sampling the randomness separately. The idea of clamping random numbers is explored in depth by Andrieu et al. [3] and Murray and Graham [30]; the latter applies to this pseudo-marginal slice sampling. In particular, our approach applies these ideas for the specific setting of multi-fidelity models, where the fidelity is treated as an auxillary variable. Finally, while our approach applies to doubly-intractable problems, we are also motivated by a larger class of multi-fidelity problems studied in the computational sciences that may not even inference problems, such as quantum simulations and PDE-constrained optimization problems.

2 Multi-fidelity MCMC

2.1 Notation and high-level summary

Monte Carlo is a tool for approximating integrals and sums that can be framed as expectations:

$$\int h(\theta)\,\pi(\theta)\,d\theta \approx \frac{1}{T}\sum_{t=1}^{T} h(\theta^{(t)}) \qquad \text{where} \qquad \theta^{(t)} \sim \pi\,. \tag{1}$$

Here $\pi(\theta)$ is the *target density* and $h(\theta)$ is a function whose expectation we wish to compute. The $\theta^{(t)}$ are samples taken from π, which are generated from a Markov chain in the case of interest here. A variety of problems can take this form; Bayesian inference is the most common use case, but essentially any integral can be framed in this way, as can many inverse and optimization problems.

In the multi-fidelity setting we consider here, we imagine that there are a sequence of target densities: $\pi_1(\theta), \pi_2(\theta), \ldots, \pi_K(\theta), \ldots$; as the index K increases the target density becomes higher-fidelity, i.e., more reflective of the underlying phenomena being modeled. Here increasing k could correspond to finer discretizations, increasing numbers of quadrature points, or performing a larger number of iterations in a solver. We assume that what the MCMC user truly wants is to estimate the functional in Eq. 1 in the *perfect-fidelity* limit $\pi_\infty(\theta)$ where there is, e.g., no discretization error. We also take it for granted that higher fidelity (larger K) densities $\pi_K(\theta)$ are more costly to evaluate, increasing in expense super-linearly with K. Conventionally one would choose a K that has a tolerable level of bias for a given computational budget.

Our strategy for multi-fidelity MCMC will be to construct an unbiased estimator of $\pi_\infty(\theta)$ using random choices of k and then include K in the Markov chain as an auxiliary variable. By carefully constructing such a Markov chain it will be possible to asymptotically estimate the functional in Eq.1 as though the samples were taken from the perfect-fidelity model; each step of the Markov chain will nevertheless only require a finite amount of computation.

2.2 Pseudo-marginal MCMC for the multi-fidelity setting

Pseudo-marginal MCMC [2, 6] is a class of MCMC algorithms that replaces the target density $\pi(\theta)$ with an estimator $\hat{\pi}(\theta)$ that is a function of a random variable. If the estimator is nonnegative and unbiased, i.e., $\hat{\pi}(\theta) \geq 0$ and $\mathbb{E}[\hat{\pi}(\theta)] = \pi(\theta)\ \forall\theta$, then MCMC transitions that use the estimator still have $\pi(\theta)$ as their invariant distribution. This property is sometimes referred to as "exact-approximate"

MCMC as the transitions are approximate but the limiting distribution is exact. Estimators can be constructed from a variety of methods, including particle filtering [2]; our approach will be similar to Lin et al. [26], Lyne et al. [27], and Georgoulas et al. [15] and use randomized series truncations.

To apply pseudo-marginal techniques to the multi-fidelity setting, we consider our estimator to arise from a random choice of the fidelity K, where we take $K \in \{1, 2, \ldots\}$. We construct a distribution $\mu(K)$ and an estimator $\hat{\pi}_K(\theta)$ (here explicitly including K in the estimator), such that

$$\sum_{k=1}^{\infty} \mu(k)\hat{\pi}_k(\theta) = \pi_\infty(\theta)\,. \tag{2}$$

Ideally, our estimator $\hat{\pi}_K(\theta)$ will prefer smaller values of K while having sufficiently low variance as to allow the Markov chain to mix effectively; thus the simulations can be run at inexpensive low-fidelities while the answers will be as though it was the perfect-fidelity model being used.

While the standard approach to pseudo-marginal MCMC would use independent estimators within Metropolis–Hastings accept/reject decisions, this technique can "get stuck" when the estimator is noisy and fail accept new states. Thus, we apply the approach in Murray and Graham [30] that augments the Markov chain to include the randomness of the estimator. A second problem with pseudo-marginal MCMC is that the unbiased estimator can return negative values, but the pseudo-marginal algorithm assumes that $\hat{\pi}_K(\theta)$ is unbiased and nonnegative in order to asymptotically sample from the correct target. Nonnegativity of the estimator is difficult to guarantee in general without additional knowledge about the boundedness of the likelihood cite. One technique considered by Lin et al. [26] and Lyne et al. [27] is to instead sample from the target distribution using the absolute value of the likelihood estimator and applying a sign-correction to the final Monte Carlo estimate in Equation (1), an approach borrowed from the quantum Monte Carlo literature where it is necessary for modelling fermions. This approach has been applied to the M-H algorithm, but we note that this general approach can be applied much more broadly, as we do in this work.

As in Murray and Graham [30], our Markov chain alternately samples between $\hat{\pi}_k(\theta)$ for fixed k and the conditional distribution $p(K \mid \theta)$. In problems where the estimator may return negative values, we sample from the conditional target distributions using the absolute value $|\hat{\pi}_k(\theta)|$, which we denote by $\tilde{p}(K \mid \theta)$ and $\tilde{\pi}_k(\theta)$. The estimate in Equation (1) is then corrected using the signs $\sigma(\theta, k)$,

$$\int h(\theta)\,\pi(\theta)\,d\theta \approx \frac{\sum_{t=1}^{T} h(\theta^{(t)})\sigma(\theta^{(t)}, K^{(t)})}{\sum_{t=1}^{T} \sigma(\theta^{(t)}, K^{(t)})} =: \hat{I}_T, \tag{3}$$

The full multi-fidelity MCMC algorithm with sign correction summarized in Appendix A. We note that while the Markov chain no longer converges to the target π but instead an absolute value target density, the final estimate after sign-correction is unbiased, which is ultimately the downstream goal of interest. While this may seem limiting if one is interested in the posterior itself, useful unbiased posterior summaries may be still be obtained via the functional, such as the posterior mean, variance, quantiles, and histograms that may be used to visualize marginal distributions.

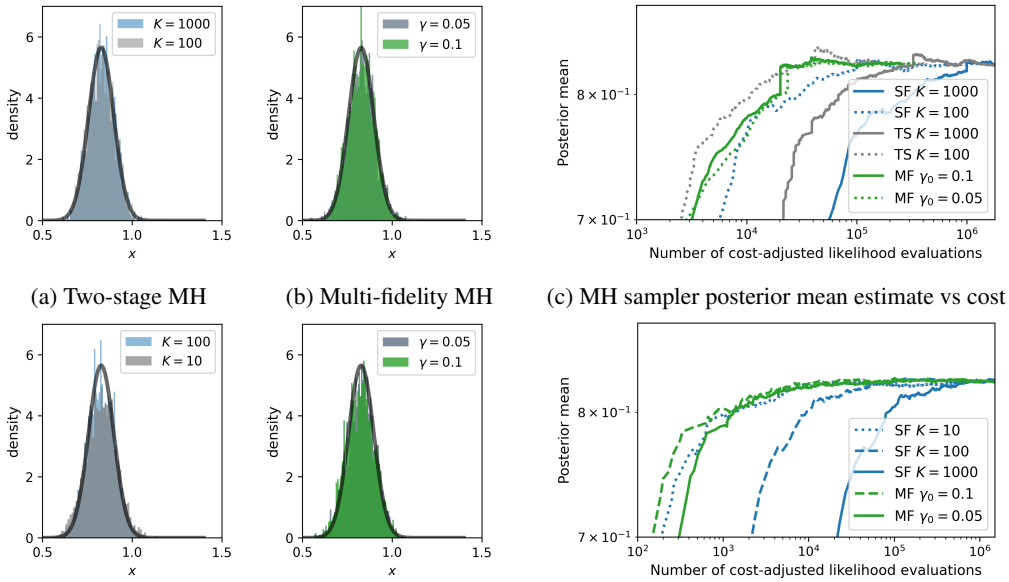
Here we provide a brief sketch of why the sign correction still leads to the correct estimate of functionals. Define $\hat{\pi}_k(\theta) = \sigma(\theta, k)|\hat{\pi}_k(\theta)|$ and let $\tilde{\pi}$ be the density satisfying $\tilde{\pi}(\theta, k) \propto |\hat{\pi}_k(\theta)|\mu(k)$. Expanding the posterior functional into its joint distribution and substituting with the above gives

$$\int h(\theta)\pi(\theta)d\theta = \sum_{k=1}^{\infty} \int h(\theta)\hat{\pi}_k(\theta)\mu(k)d\theta = \frac{\int \sum_{k=1}^{\infty} h(\theta)\sigma(\theta, k)\tilde{\pi}(\theta, k)d\theta}{\int \sum_{k=1}^{\infty} \sigma(\theta, k)\tilde{\pi}(\theta, k)d\theta}.$$

A Monte Carlo estimate of the rightmost ratio gives the estimator $\hat{I}_T$ in Equation (3), where $\theta^{(t)}, K^{(t)} \sim \tilde{\pi}(\theta, K)$. Finally, sampling from the conditional target with the low-fidelity likelihood estimator generates samples from the joint distribution $\tilde{\pi}(\theta, K)$.

3 Unbiased low-fidelity estimators via randomized truncations

In this section, we discuss how to construct an unbiased estimator of $\pi_\infty(\theta)$, given a sequence of low-fidelity likelihoods with the property $\pi_k(\theta) \to \pi_\infty(\theta)$ as $k \to \infty$. This estimator has the property that it requires a finite amount of computation with probability one, and it also has a tunable amount of expected computation per estimate, i.e., it uses low-fidelity density evaluations to estimate the



(a) Two-stage MH (b) Multi-fidelity MH (c) MH sampler posterior mean estimate vs cost

(d) Single-fidelity slice (e) Multi-fidelity slice (f) Slice sampler posterior mean estimate vs cost

Figure 2: Conjugate Gaussian model with a Gaussian likelihood sequence. **Top left (a,b):** Histograms for two-stage M-H, where K denotes the high-fidelity model, and multi-fidelity (pseudo-marginal) MH. **Top right (c):** Comparison of posterior mean estiamate vs computation for single-fidelity, multi-fidelity, and two-stage multi-fidelity M-H. Average posterior mean computed over 4 different chains. **Bottom left (d,e):** Histograms for single-fidelity slice sampling and multi-fidelity (pseudo-marginal) M-H. **Bottom right (f):** Comparison of posterior mean estiamate vs computation for single-fidelity and multi-fidelity slice sampling.

perfect-fidelity target density. The central idea of this estimator has been used for decades, going back to John von Neumann and Stanislaw Ulam. More recently it has found use in related work such as Lyne et al. [27], Beatson and Adams [5], Jacob et al. [23], and Glynn and Rhee [17]. First note that we can express the perfect-fidelity model as a telescoping sum: let $\pi_0(\theta) = 0$ and write

$$\pi_\infty(\theta) = \sum_{k=1}^{\infty} \pi_k(\theta) - \pi_{k-1}(\theta). \tag{4}$$

The estimator $\hat{\pi}_K$ is then constructed by taking a random truncation $K \sim \mu$ of the infinite telescoping series. The sampled terms in the sum are then reweighted to ensure the estimator remains unbiased:

$$\hat{\pi}_K(\theta) = \sum_{k=1}^{K} w_{k,K}(\pi_k(\theta) - \pi_{k-1}(\theta)). \tag{5}$$

Two approaches are commonly used to ensure that the resulting estimator is unbiased: weighted single-term estimators and Russian roulette estimators. The single-term estimator [27] is constructed by importance sampling a term from the series in Equation (4): the truncation level is drawn as $K \sim \mu$, and the Kth term is used to form the estimate $\hat{\pi}_K(\theta) = \mu(K)^{-1}(\pi_K(\theta) - \pi_{K-1}(\theta))$. Thus, the weight in Equation (5) is $W_{k,K} = \mu(K)^{-1}\mathbb{1}(n = N)$. In the Russian roulette estimator, the remaining terms in the estimator are reweighted by their survival probabilities, i.e., $W_{k,K} = (1 - \sum_{k'=1}^{k-1} \mu(k'))^{-1}\mathbb{1}(K \geq k)$.

The ability to use cheaper models is a key feature of multi-fidelity inference, and the low-fidelity estimator provides a means to reduce the computational cost of multi-fidelity Monte Carlo. However, these estimators are an example of a class of methods that explores a compute-variance tradeoff: computationally cheaper estimates leads to high variability. The resulting increase in variance slows down the convergence of the MCMC procedure and could lead to an overall less efficient method due to a reduced effective sample size.

4 Summary of the Multi-Fidelity MCMC Recipe

Here we summarize the recipe for constructing a multi-fidelity Markov chain Monte Carlo algorithm.

First, identify a sequence of increasing-fidelity target densities, with the property that their limit is the desired "perfect-fidelity" density. Low-fidelity densities should be cheap with the cost rapidly increasing within the sequence. In the context of Bayesian inference, it may be appropriate to focus the multi-fidelity aspects on the likelihood term and construct the target densities via, e.g., $\pi_k(\theta) \propto \pi(\theta)L_k(\theta\,;\mathcal{D})$. This likelihood-based version is what we use in several of the experiments.

Next, introduce a proposal distribution μ on the positive integers. This proposal distribution should be chosen to balance between expected cost and variance of the resulting estimator; our overall goal is to mostly use cheap low-fidelity densities, but high variance estimators will presumably damage the mixing time and/or the asymptotic variance.

Initialize the Markov chain with a reasonable choice for θ and a draw of K from the distribution μ. Each step of the Markov chain simulation consists of an update to θ given K and an update of K given θ. The update of θ given K can be done using any standard MCMC algorithm, e.g., Metropolis-Hastings, slice sampling, or Hamiltonian Monte Carlo. It is important to remember to use the absolute value of the estimator and keep track of its sign. The update of K given θ is also flexible, but it is reasonable to construct things so that only a few K are considered in each step, as each of those fidelities will need to be evaluated. A simple random walk on the positive integers appears to suffice.

After running a sufficient number of steps of the Markov chain, use the sign corrected-estimator in Equation (3) to compute the expectation of the your function $h(\theta)$.

5 Experiments

In this section, we consider a number of applications of MF-MCMC. In all experiments, we use a random-walk MH update to sample from the conditional $K|\theta$. For the truncation distribution μ, we use a geometric distribution with parameter γ_0. Additional experiments and details are in Appendix F.

5.1 Toy conjugate Gaussian models

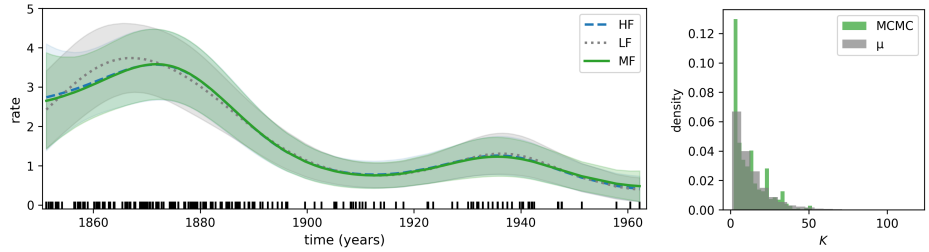
We first examine a toy example that does not involve any high- or low-fidelity computation (i.e., the computational cost does not increase with k): consider a high-fidelity model $L_\infty(\theta) = \mathcal{N}(x;\theta,\sigma_\infty)$ and a low-fidelity model $L_k(\theta) = \mathcal{N}(x;\theta,\sigma_k)$, where $\sigma_k^2 \to \sigma_\infty^2$. Here we we consider the sequence $\sigma_k^2 = 1+2/k^2$. In Figure 2 we compare the performance of single-fidelity and multi-fidelity M-H and slice sampling. We also compare to the two-stage M-H algorithm summarized in Appendix C.4; here we consider 2 two-stage setups of $k = \{1000, 10\}$ and $k = \{100, 5\}$. The histograms show the bias of each method after simulating 10,000 samples, and the solid gray curve denotes the exact posterior density. The number of cost-adjusted likelihoods was computed by multiplying each likelihood evaluation by the fidelity. Here we see that the multi-fidelity methods converge to the same value as the high-fidelity but in fewer cost-adjusted likelihood evaluations.

5.2 Log-Gaussian Cox processes

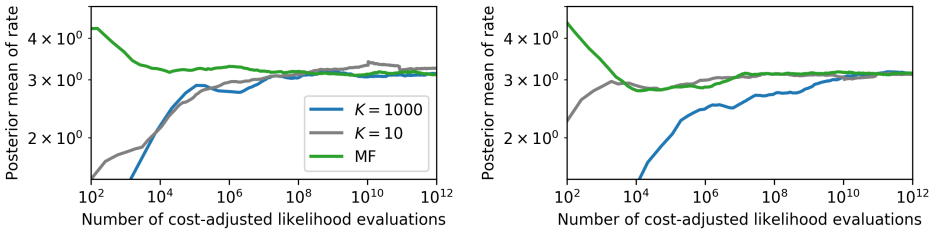
We now examine an application of MF-MCMC to the log Gaussian Cox process (LGCP) model [29], where the perfect-fidelity model involves and integral and the lower-fidelity sequence of models arise from k-point quadrature estimates. Let $\log f \sim \mathrm{GP}(0, k_\theta)$, where $k(x,x') = \exp\left(-\frac{1}{2\ell^2}\|x-x'\|_2^2\right)$, where ℓ is a lengthscale hyperparameter. Consider an inhomogenous Poisson process on $\mathbb{X} \subseteq \mathbb{R}^D$ with intensity $\lambda(x) = e^{f(x)}$. Given a random set of points $\{X_1, \ldots, X_N\}$, where $X_n \in \mathbb{X}$,

$$L_\infty(f) = \exp\left(\int_{\mathbb{X}} (1 - e^{f(x)})dx\right) \prod_{n=1}^{N} e^{f(X_n)}. \tag{6}$$

Typically, inference in the LGCP uses a grid-based approximation of Equation (6), where the points are binned into counts and each bin has a Poisson likelihood [11, 31, 38]. This approximation leads to a biased posterior. We perform inference for f using the elliptical slice sampling (ESS) algorithm; see Appendix C.3 for a summary of the method. We approximate the integral in Equation (6) with a trapezoidal quadrature rule I_k, where the number of quadrature points is a linear function of k.



(a) Posterior mean and marginal μ



(b) Posterior mean of rate at $t = 1860$ (left) and $t = 1880$ (right) vs computation

Figure 3: Coal mining disasters 1850–1963. The single-fidelity models used $K^{HF} = 1000$, $K^{LF} = 10$. **Top:** Posterior mean estimates of the rate function at the observed data point times (left) and the marginal distribution of fidelities from the MF method (right). **Bottom:** Posterior mean estimates of the rate function at two time points vs computation.

We apply multi-fidelity and single-fidelity ESS algorithms to a coal mining disasters data set Carlin et al. [8]. The data contain the dates of 191 coal mine explosions that killed ten or more men in Britain between 15 March 1851 and 22 March 1962. Figure 3a shows the estimated mean intensity on coal mining disasters data between one run of multi-fidelity ESS and two single-fidelity ESS runs on a high-fidelity ($k = 1000$ quadrature points) and low-fidelity ($k = 10$ quadrature points) setting. We also computed the cost-adjusted number of likelihood evaluations performed in each iteration of MF-ESS and SF-ESS. The posterior mean estimates were computed after discarding 1000 burnin samples and then collecting every third sample. Figure 3b shows the average estimated mean intensity at two time steps ($t = 1860, 1880$) on the three models plotted against the average cost-adjusted number of likelihood evaluations per iteration. Here we can see observe that the multi-fidelity and high-fidelity estimates are quite close after many iterations of sampling, but that the multi-fidelity estimate converges with less computation.

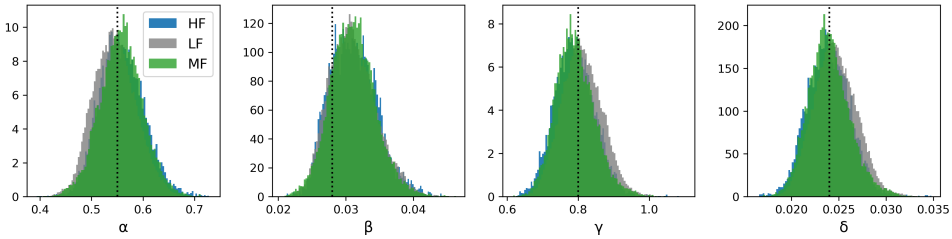
5.3 Bayesian ODE system identification

In this section, we demonstrate the usage of MF-MCMC for a simple Bayesian system identification problem. Consider the classical Lotka-Volterra ODE problem. Let $u(t) \geq 0$ represent the population size of prey species at time t, and $v(t) \geq 0$ represent the population size of the predator species. The dynamics of the two species given parameters $\alpha, \beta, \gamma, \delta \geq 0$ are given by a pair of first-order ODEs:

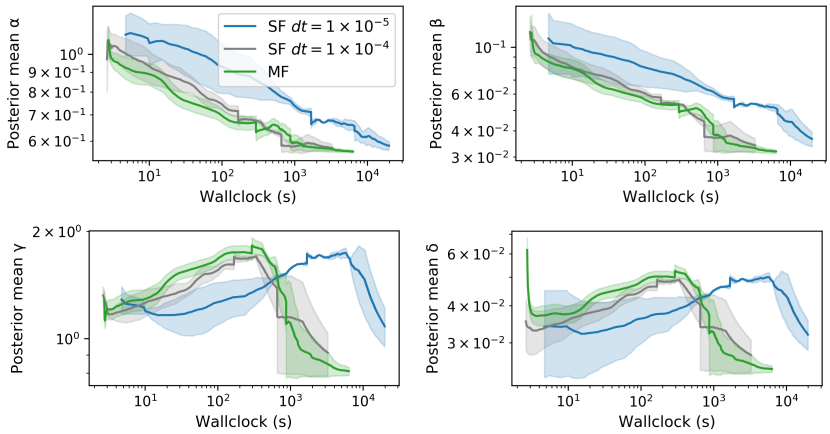
$$\frac{d}{dt}u = (\alpha - \beta v)u = \alpha u - \beta uv, \qquad \frac{d}{dt}v = (-\gamma - \delta u)v = -\gamma v - \delta uv. \tag{7}$$

System identification solves the inverse problem by identifying the parameters of the ODE system $\theta = (\alpha, \beta, \gamma, \delta)$. In a Bayesian approach, we can specify a noise model for the observed data and priors on the parameters and use MCMC to infer the solution. For simplicity, we assume that the initial conditions are known and fix $\sigma = 0.25$.

Define $z_n := (u(t_n), v(t_n))$ and let $z_1(\theta), \ldots, z_N(\theta)$ be the solutions to the Lotka-Volterra differential equations given θ at times $t_1, \ldots, t_N$ given the initial conditions and the system parameters $\theta = (\alpha, \beta, \gamma, \delta)$. Suppose we have observations arising from $\log(y_n) = \log(z_n) + \epsilon_n$, where $\epsilon \sim N(0, \sigma^2 I)$. In our experiments, we generate data with initial conditions $z_0 = [1.0, 1.0]$, system parameters $\alpha = 1.5, \beta = 1.0, \gamma = 3.0, \delta = 1.0$, and noise parameter $\sigma = 0.8$ at a grid of N solution values. For the ODE solver, we considered both an Euler solver and an 4th-order Runge-Kutta solver. Results for the Euler solver are in Figure 4, and results for the Runge-Kutta sovler are in Figure F5.



(a) Marginal densities of system parameters



(b) Posterior mean estimate vs computational cost

Figure 4: Lokta-Volterra system parameter identification. The fidelity represents (a function of) the step size of the ODE solver. **Top:** Marginal distributions of system parameters. Black dotted line represents point estimates reported by Howard [22]. **Bottom:** Posterior mean estimates of the parameters vs wallclock.

5.4 PDE-constrained optimzation

We now consider global optimization of a PDE-constrained objective, where an expensive physical simulation is run repeatedly in an outer loop problem. A common approach for global optimization is simulated annealing, which has been applied to constrained global optimization [36]. Consider a model for heat flow in a thin rod of length L with spatial coordinates $x \in [0, L]$. Let $u(x, t)$ represent the temperature in the rod at position x and time t, and let $\bar{u}$ represent a desired target temperature. The goal is to minimize a loss function subject to u satisfying a linear heat equation. This objective along with an initial condition and homogenous boundary conditions can be summarized as:

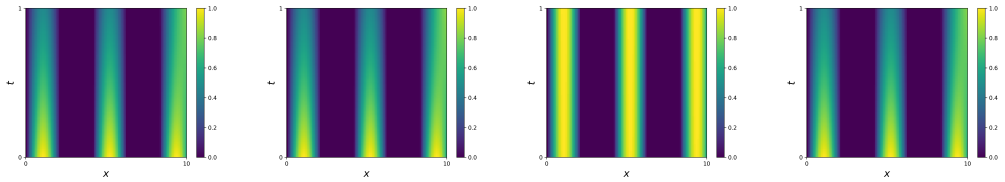
$$\begin{aligned} &\text{minimize}_u \quad \|u - \bar{u}\|_2^2 \\ &\text{subject to} \quad \frac{\partial u}{\partial t} = \alpha \cdot \frac{\partial^2 u}{\partial x^2} + 2\beta \cdot u \\ &\qquad u(x,0) = \sin\left(\pi x/2\right), \quad u(0,t) = u(L,t) = 0, \qquad x \in [0,L], t \in [0,T]. \end{aligned} \tag{8}$$

where $\alpha, \beta > 0$ are the system parameters. The goal is to find $\theta = (\alpha, \beta)$ that minimizes the objective and satisfies the constraints. To solve the PDE, we discretize the domain into a grid of size Δx and represent the second derivative using the central difference formula. This gives us a system of ODEs that we solve numerically using a Tsitouras 5/4 Runge-Kutta method, setting setting $\Delta t = 0.4\Delta x^2$ so as to satisfy a CFL stability condition. Here the fidelity of the problem is given by the size of the spatial discretization Δx, which in turn controls the discretization of Δt. The results are in Figure 5.

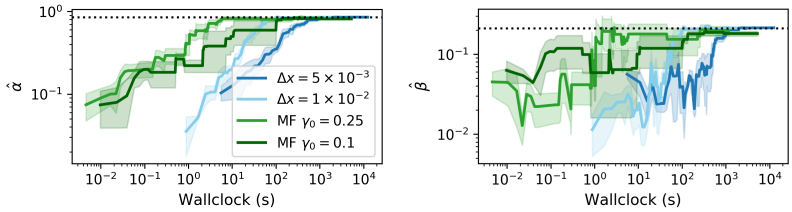
5.5 Gaussian process regression parameter inference

Let $X \in \mathbb{R}^{N \times D}$ and consider a Gaussian process regression model with a squared exponential kernel:

$$f \sim \mathrm{GP}(0, k_\theta), \quad y = f(X) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_0^2), \quad k_\theta(x, x') = \exp\left(-\frac{1}{2\theta^2}\|x - x'\|_2^2\right). \tag{9}$$

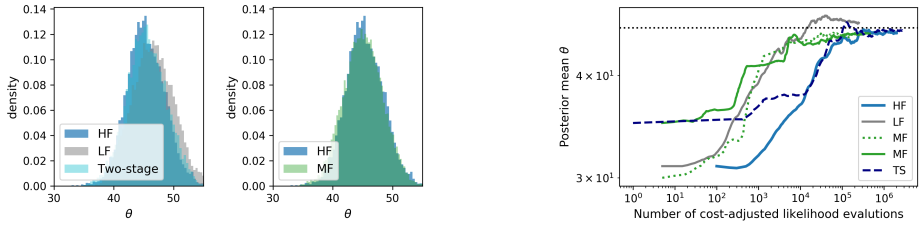


(a) Target temperature $\bar{u}$ (b) Multi-fidelity solution (c) Low-fidelity solution (d) High-fidelity solution



(e) Error for α (left) and β (right) vs computation

Figure 5: PDE-constrained optimization with a linear heat equation



(a) Histograms of high, low, two-stage, and multi-fidelity (b) Posterior mean estimate vs computation

Figure 6: Parameter inference in a Gaussian process regression model.

For simplicity, we assume σ_0^2 is known. Let $\Sigma_\theta := [k_\theta(x_i, x_j)]_{i\in[N],j\in[N]}$. In many applications of GPs, e.g., computing the predictive distribution, one is interested in integrating out the parameters θ using MCMC. Computing the posterior $\pi(\theta|X, y)$ is expensive because each evaluation of the likelihood $p(y|X, \theta) = \mathcal{N}(y \mid 0, \Sigma_\theta + \sigma_0^2 I)$ involves computing a determinant and solving a linear system with respect to the matrix $\Sigma_\theta + \sigma_0^2 I$, which has an $O(N^3)$ computational cost associated with standard methods (e.g., Cholesky decomposition). Additional derivations and details are in Appendix F.3. We generate synthetic data from the GP model in Equation (9), with $\sigma_0^2 = 1$ and lengthscale $\theta_0 = 25$. For the GP model, we use the prior $\theta \sim \log \mathcal{N}(0, 1)$. We compare several likelihoods: a high-fidelity likelihood ($K = N$), low-fidelity likelihood ($K = K_N \ll N$), and the multi-fidelity approach we describe. In Figure 6, we compare these approaches.

6 Discussion and future work

In this work, we introduced a class of multi-fidelity MCMC that uses a low-fidelity unbiased estimator to reduce the computational cost of sampling while still maintaining the desired limiting target distribution of the Markov chain. In particular, we have demonstrated the use of our framework on other MCMC algorithms besides Metropolis-Hastings, such as slice sampling, and to other settings such as optimization. Many future directions remain. First, applying MF-MCMC to large-scale expensive applications has many computational challenges. Thus, making the method more robust to specialized problems is important, especially if the estimator is heavy-tailed. In addition, we have thusfar focused on target densities where there is a single computation whose fidelity is varied. However, in many settings, there may be target densities with multiple computations that converge at different rates, for example, if the likelihood includes both an integral and a solution of a linear system. Our framework can be extended to that setting, and it is useful to understand how these rates impact the convergence properties of the sampler.

References

[1] N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon. A trust-region framework for managing the use of approximation models in optimization. *Structural optimization*, 15(1): 16–23, 1998.

[2] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[3] C. Andrieu, A. Doucet, and R. Holenstein. Discussion of "Particle Markov chain Monte Carlo methods". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342, 2010.

[4] E. Arian, M. Fahl, and E. W. Sachs. Trust-region proper orthogonal decomposition for flow control. Technical report, 2000.

[5] A. Beatson and R. P. Adams. Efficient optimization of loops and limits with randomized telescoping sums. In *International Conference on Machine Learning*, pages 534–543. PMLR, 2019.

[6] M. A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003.

[7] L. Brevault, M. Balesdent, and A. Hebbal. Overview of Gaussian process based multi-fidelity techniques with variable relationship between fidelities. *arXiv preprint arXiv:2006.16728*, 2020.

[8] B. P. Carlin, A. E. Gelfand, and A. F. Smith. Hierarchical Bayesian analysis of changepoint problems. *Journal of the royal statistical society: series C (applied statistics)*, 41(2):389–405, 1992.

[9] J. A. Christen and C. Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005.

[10] T. Cui, Y. M. Marzouk, and K. E. Willcox. Data-driven model reduction for the Bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102 (5):966–990, 2015.

[11] P. J. Diggle, P. Moraga, B. Rowlingson, and B. M. Taylor. Spatial and spatio-temporal log-Gaussian Cox processes: extending the geostatistical paradigm. *Statistical Science*, 28(4): 542–563, 2013.

[12] T. J. Dodwell, C. Ketelsen, R. Scheichl, and A. L. Teckentrup. A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1075–1108, 2015.

[13] Y. Efendiev, T. Hou, and W. Luo. Preconditioning Markov chain Monte Carlo simulations using coarse-scale models. *SIAM Journal on Scientific Computing*, 28(2):776–803, 2006.

[14] M. Fahl and E. W. Sachs. Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition. In *Large-scale PDE-constrained optimization*, pages 268–280. Springer, 2003.

[15] A. Georgoulas, J. Hillston, and G. Sanguinetti. Unbiased Bayesian inference for population Markov jump processes via random truncations. *Statistics and computing*, 27(4):991–1002, 2017.

[16] A. Gessner, J. Gonzalez, and M. Mahsereci. Active multi-information source Bayesian quadrature. In *Uncertainty in Artificial Intelligence*, pages 712–721. PMLR, 2020.

[17] P. W. Glynn and C.-h. Rhee. Exact estimation for Markov chain equilibrium expectations. *Journal of Applied Probability*, 51(A):377–389, 2014.

[18] R. B. Gramacy and H. K. Lee. Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–145, 2009.

[19] G. W. Gundersen, D. Cai, C. Zhou, B. E. Engelhardt, and R. P. Adams. Active multi-fidelity Bayesian online changepoint detection. In *Uncertainty in Artificial Intelligence*, pages 1916–1926. PMLR, 2021.

[20] W. Hackbusch. *Multi-grid methods and applications*, volume 4. Springer Science & Business Media, 2013.

[21] D. Higdon, H. Lee, and Z. Bi. A Bayesian approach to characterizing uncertainty in inverse problems using coarse and fine-scale information. *IEEE Transactions on Signal Processing*, 50 (2):389–399, 2002.

[22] P. Howard. Modeling basics. *Lecture Notes for Math*, 442, 2009.

[23] P. E. Jacob, J. O'Leary, and Y. F. Atchadé. Unbiased Markov chain Monte Carlo methods with couplings. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(3): 543–600, 2020.

[24] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[25] S. Li, W. Xing, R. Kirby, and S. Zhe. Multi-fidelity Bayesian optimization via deep neural networks. *Advances in Neural Information Processing Systems*, 33:8521–8531, 2020.

[26] L. Lin, K. Liu, and J. Sloan. A noisy Monte Carlo algorithm. *Physical Review D*, 61(7):074505, 2000.

[27] A.-M. Lyne, M. Girolami, Y. Atchadé, H. Strathmann, and D. Simpson. On Russian roulette estimates for Bayesian inference with doubly-intractable likelihoods. *Statistical science*, 30(4): 443–467, 2015.

[28] A. March and K. Willcox. Constrained multifidelity optimization using model calibration. *Structural and Multidisciplinary Optimization*, 46(1):93–109, 2012.

[29] J. Møller, A. R. Syversveen, and R. P. Waagepetersen. Log Gaussian Cox processes. *Scandinavian journal of statistics*, 25(3):451–482, 1998.

[30] I. Murray and M. Graham. Pseudo-marginal slice sampling. In *Artificial Intelligence and Statistics*, pages 911–919. PMLR, 2016.

[31] I. Murray, R. Adams, and D. MacKay. Elliptical slice sampling. In *Artificial Intelligence and Statistics*, pages 541–548. PMLR, 2010.

[32] A. Palizhati, S. B. Torrisi, M. Aykol, S. K. Suram, J. S. Hummelshøj, and J. H. Montoya. Agents for sequential learning using multiple-fidelity data. *Scientific reports*, 12(1):1–13, 2022.

[33] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.

[34] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Inferring solutions of differential equations using noisy multi-fidelity data. *Journal of Computational Physics*, 335:736–746, 2017.

[35] T. Robinson, M. S. Eldred, K. E. Willcox, and R. Haimes. Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. *AIAA journal*, 46(11):2814–2822, 2008.

[36] H. E. Romeijn and R. L. Smith. Simulated annealing for constrained global optimization. *Journal of Global Optimization*, 5(2):101–126, 1994.

[37] J. Song, Y. Chen, and Y. Yue. A general framework for multi-fidelity Bayesian optimization with gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3158–3167. PMLR, 2019.

[38] M. Teng, F. Nathoo, and T. D. Johnson. Bayesian computation for Log-Gaussian Cox processes: A comparative analysis of methods. *Journal of Statistical Computation and Simulation*, 87(11): 2227–2252, 2017.

[39] M. Troyer and U.-J. Wiese. Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations. *Physical review letters*, 94(17):170201, 2005.

[40] J. Wu, S. Toscano-Palmerin, P. I. Frazier, and A. G. Wilson. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In *Uncertainty in Artificial Intelligence*, pages 788–798. PMLR, 2020.

[41] X. Xi, F.-X. Briol, and M. Girolami. Bayesian quadrature for multiple related integrals. In *International Conference on Machine Learning*, pages 5373–5382. PMLR, 2018.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discuss this in the discussion section.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We include a section in Appendix G
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We provide details in Appendix E
   (b) Did you include complete proofs of all theoretical results? [Yes] We provide details in Appendix E
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will include a URL for camera ready.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Yes, see Appendix F for full details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] The plots are generated over multiple random seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix F
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]