Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

A NEW STOCHASTIC APPROXIMATION METHOD FOR GRADIENT-BASED SIMULATED PARAMETER ESTIMATION

Zehao Li[1,2] and Yijie Peng[1,2]

[1]Guanghua School of Management, Peking University, Beijing, CHINA
[2]Xiangjiang Laboratory, Changsha, Hunan, CHINA

ABSTRACT

This paper tackles the challenge of parameter calibration in stochastic models, particularly in scenarios where the likelihood function is unavailable in an analytical form. We introduce a gradient-based simulated parameter estimation framework, which employs a multi-time scale stochastic approximation algorithm. This approach effectively addresses the ratio bias that arises in both maximum likelihood estimation and posterior density estimation problems. The proposed algorithm enhances estimation accuracy and significantly reduces computational costs, as demonstrated through extensive numerical experiments. Our work extends the GSPE framework to handle complex models such as hidden Markov models and variational inference-based problems, offering a robust solution for parameter estimation in challenging stochastic environments.

1 INTRODUCTION

Parameter estimation is a vital aspect in fields like financial risk assessment and medical diagnosis, where it entails calibrating model parameters based on observed data. Frequentist methods treats parameters as unknown constants, whereas the Bayesian perspective infers their posterior distribution. Important inference methods include maximum likelihood estimation (MLE), which provides consistency and asymptotic efficiency (Shao 2003), and posterior density estimation (PDE), which combines observed data with prior knowledge to ensure precise inference. Both techniques are extensively used in statistics and machine learning.

To solve the MLE, one relies on the analytical form of the logarithmic likelihood function. By substituting the observed data and solving for its maximum, the MLE can be derived. For PDE, the classical method involves variational inference (Blei et al. 2017), which similarly requires an analytical form of the logarithmic likelihood. This method assumes a family of posterior distributions and minimizes the Kullback-Leibler divergence (KL divergence) to derive optimal posterior parameters. This paper focuses on stochastic models or simulators, which are characterized by system dynamics rather than explicit likelihood functions. Examples include Lindley's recursion in queuing models, where the likelihood function of the output data does not have an analytical form, posing significant challenges for parameter calibration.

The MLE problem was first introduced and addressed by the gradient-based simulated maximum likelihood estimation (GSMLE) method in Peng et al. (2020). The Robbins-Monro algorithm, a classic stochastic approximation (SA) method (Kushner and Yin 2003), is used to optimize unknown parameters for MLE. Specifically, let Y denote the observed data, $\theta \in \mathbb{R}^d$ represent the parameter of interest, and p stand for the unknown density. The gradient of the logarithmic likelihood function $\sum_{t=1}^{T} \log p(Y_t;\theta)$ with respect to θ can be expressed as a ratio:

$$\nabla_\theta \sum_{t=1}^{T} \log p(Y_t;\theta) = \sum_{t=1}^{T} \frac{\nabla_\theta p(Y_t;\theta)}{p(Y_t;\theta)}. \tag{1}$$

When no analytical form for the likelihood function is available, the generalized likelihood ratio (GLR) method is employed to obtain unbiased estimators for the density and its gradients (Peng et al. 2018). The GLR estimator provides unbiased estimators for "distribution sensitivities" as shown in Lei et al. (2018), achieving a square-root convergence rate (Glynn et al. 2021).

However, the gradient estimator of the logarithmic likelihood function presented in Peng et al. (2020) is biased. Although the GLR estimator is unbiased, meaning we can obtain unbiased estimators $G_1(Y_t,\theta)$ and $G_2(Y_t,\theta)$ for $\nabla_\theta p(Y_t;\theta)$ and $p(Y_t;\theta)$ through the GLR method and Monte Carlo simulation, the ratio of these two unbiased estimators may not be unbiased. Therefore, when this ratio estimator is used in the Robbins-Monro algorithm, the update rule becomes:

$$\theta_{k+1} = \theta_k + \beta_k \sum_{t=1}^{T} \frac{G_1(Y_t,\theta_k)}{G_2(Y_t,\theta_k)}, \tag{2}$$

where the gradient term is biased, introducing a certain bias into the iterative results (β_k is the step-size, satisfying specific conditions). Additionally, the estimator in the denominator may cause numerical instability, resulting in inaccuracies in the MLE.

In the context of PDE, the computation of the log-likelihood function is similarly crucial. When the likelihood function does not have an analytical form, an estimator must be devised. In this simulation-based inference, also referred to as likelihood-free inference, various methods utilize neural networks to estimate likelihoods or posteriors that are otherwise infeasible to calculate (Glöckler et al. 2022; Papamakarios et al. 2019). However, the likelihood functions inferred by neural networks tend to be biased. The integration of neural networks and the associated bias make these algorithms theoretically challenging. To simplify this and enable theoretical analysis, we frame this problem within the SA framework, utilizing unbiased GLR gradient estimators for the likelihood function as in the MLE case. Since the gradient estimator of the posterior density also involves Equation (1), reducing the ratio bias in these stochastic models remains an open problem.

To tackle the ratio bias arising in both MLE and PDE, we propose a gradient-based simulated parameter estimation (GSPE) algorithm based on a multi-time scale (MTS) SA method (Kushner and Yin 2003; Borkar 2009). The core idea is to treat both the parameters and the gradient of the logarithmic likelihood function together as parts of a stochastic root-finding problem. The method then approximates the solution by using two coupled iterations, with one component updated at a faster rate than the other. Specifically, we develop a recursive estimator that replaces the ratio form of the gradient estimator. This approach attempts to approximate the solution by devising two separate but coupled iterations, where one component is updated at a faster pace than the other. Specifically, we find a recursive estimator that substitutes the ratio form of the gradient estimator, thus eliminating the ratio bias throughout the iterative process. ariety of problems, including bilevel optimization (Hong et al. 2023), minmax optimization (Lin et al. 2025), and reinforcement learning scenarios (Khodadadian et al. 2022). They have also been used extensively in quantile optimization (Hu et al. 2022; Hu et al. 2025; Jiang et al. 2022), black-box CoVaR estimation (Cao et al. 2023), and dynamic pricing and replenishment problems (Zheng et al. 2024).

Our approach, however, involves a more complex structure. The PDE problem is formulated as a nested simulation optimization problem within the variational inference framework. Minimizing KL divergence is equivalent to maximizing the evidence lower bound (ELBO), which is expressed as an expectation with respect to the unknown variational distribution. Consequently, the optimization objective is an expectation, and the sample average approximation (SAA) method is applied to obtain an unbiased gradient estimator for the ELBO, forming the outer layer of the simulation. Meanwhile, the intractable likelihood within this expectation is estimated using the inner-layer simulation and unbiased GLR estimators. A nested MTS algorithm is designed to address ratio bias, solving the nested simulation optimization problem.

Furthermore, the MLE in hidden Markov models (HMM) is also a difficult problem. The likelihood function in HMMs is a high-dimensional integral over the hidden states, which does not have a closed form. Estimating the gradient of this likelihood becomes problematic. Sequential Monte Carlo (SMC),

also known as particle filtering, is a standard method for handling HMMs (Wills and Schön 2023; Doucet et al. 2001). We find that the proposed algorithm can be applied to this complex scenario in conjunction with SMC.

In this paper, we introduce a new GSPE framework that can asymptotically eliminate ratio bias in parameter estimation without requiring an analytical likelihood function. The MTS algorithm is employed in the MLE problem to enhance estimation accuracy and reduce computational cost. The GSPE framework also incorporates a nested MTS algorithm to solve the PDE problem in conjunction with variational inference. MLE for HMMs is also addressed as a specific case. This work extends the previous GSPE framework in Li and Peng (2024) to HMMs, with all theoretical results omitted.

The paper is organized as follows: Section 2 provides the necessary background and introduces the GSPE algorithm framework for both MLE and PDE. Section 3 presents numerical results, and Section 4 concludes the paper.

2 PROBLEM SETTING AND ALGORITHM DESIGN

This section outlines the fundamental problem settings within the GSPE framework. To address the issue of ratio bias in the MLE problem, we propose the MTS algorithm in Section 2.1. Additionally, a nested MTS approach is introduced to tackle the PDE problem in Section 2.2. Furthermore, MLE for HMMs is discussed in Section 2.3.

2.1 Maximum Likelihood Estimation

Considering a stochastic model, let X be a random variable with density function $f(x,\theta)$ where $\theta \in \mathbb{R}^d$ is the parameter with feasible domain $\Theta \subset \mathbb{R}^d$. Another random variable Y is defined by the relationship $Y = g(X,\theta)$, where g is known in analytical form. In this model, Y is observable with X being latent. Our objective is to estimate the parameter θ based on the observed data $y := \{Y_t\}_{t=1}^T$.

In a special case where X is one-dimensional with density $f(x)$, and g is invertible with a differentiable inverse with respect to the y, a standard result in probability theory allows the density of Y_t to be expressed in closed form as: $p(y;\theta) = f(g^{-1}(y;\theta))|\frac{d}{dy}g^{-1}(y;\theta)|$. However, the theory developed in this paper does not require such restrictive assumptions. Instead, we only assume that g is differentiable with respect to x and that its gradient is non-zero a.e.

Under this weaker condition, even though the analytical forms of f and g are known, the density of Y may still be unknown. In this case, the likelihood function for Y can only be expressed as:

$$L_T(\theta) := \sum_{t=1}^{T} \log p(Y_t;\theta). \tag{3}$$

To maximize $L_T(\theta)$, we compute the gradient of the log-likelihood:

$$\nabla_\theta L_T(\theta) = \sum_{t=1}^{T} \frac{\nabla_\theta p(Y_t;\theta)}{p(Y_t;\theta)}. \tag{4}$$

Suppose we have unbiased estimators for $\nabla_\theta p(Y_t;\theta)$ and $p(Y_t;\theta)$ for every θ and Y_t. While these individual estimators are unbiased, the ratio of two unbiased estimators may introduce bias. To distinguish between approaches, we refer to the previous algorithm using the plug-in estimator from Equation (2) as the single time scale (STS) algorithm (Peng et al. 2020). To address this issue, we adopt an MTS framework that incorporates the gradient estimator into the iterative process, aiming for more accurate optimization results. Specifically, let $G_1(X,y,\theta)$ and $G_2(X,y,\theta)$ represent unbiased estimators obtained via Monte Carlo simulation:

$$G_1(X,Y_t,\theta) = \frac{1}{N}\sum_{i=1}^{N} G_1(X_i,Y_t,\theta), \quad G_2(X,Y_t,\theta) = \frac{1}{N}\sum_{i=1}^{N} G_2(X_i,Y_t,\theta), \tag{5}$$

such that

$$\mathbb{E}_X[G_1(X,Y_t,\theta)] = \nabla_\theta p(Y_t;\theta), \quad \mathbb{E}_X[G_2(X,Y_t,\theta)] = p(Y_t;\theta).$$

The forms of G_1 and G_2 can be derived by GLR estimators (Peng et al. 2020). Alternative single-run unbiased estimators for G_1 and G_2 can also be obtained via the conditional Monte Carlo method, as described in (Fu et al. 2009). We propose the iteration formulae for the MTS algorithm as follows:

$$D_{k+1} = D_k + \alpha_k(G_{1,k}(X,Y,\theta_k) - G_{2,k}(X,Y,\theta_k)D_k), \tag{6}$$

$$\theta_{k+1} = \Pi_\Theta(\theta_k + \beta_k E D_k), \tag{7}$$

where Π_Θ is the projection operator that maps each iteratively obtained θ_k onto the feasible domain Θ. The algebraic notations are as follows. $G_{1,k}(X,Y,\theta_k)$ represents the combination of all estimators $G_1(X,Y_t,\theta_k)$ under every observation Y_t, forming a column vector with $T \times d$ dimensions. $G_{2,k}(X,Y,\theta_k)$ is also the combination of all estimators $G_2(X,Y_t,\theta_k)$ under every observation Y_t. That is to say, $G_{2,k}(X,Y,\theta_k) = diag\{G_2(X,Y_1,\theta_k)I_d,\cdots,G_2(X,Y_T,\theta_k)I_d\} = diag\{G_2(X,Y_1,\theta_k),\cdots,G_2(X,Y_T,\theta_k)\}\otimes I_d$, which is a diagonal matrix with $T \times d$ rows and $T \times d$ columns. $\otimes$ stands for Kronecker product and I_d denotes the d-dimensional identity matrix. The constant matrix $E = [I_d, I_d, \cdots, I_d] = e^T \otimes I_d$ is a block diagonal matrix with d rows and $T \times d$ column, where e is a column vector of ones. This matrix reshapes the long vector D_k to match the structure of Equation (4), the summation of T d-dimensional vectors.

In these two coupled iterations, θ_k is the parameter being optimized in the MLE process, as in Equation (2). The additional iteration for D_k tracks the gradient of the log-likelihood function, mitigating ratio bias and numerical instability caused by denominator estimators. These two iterations operate on different time scales, with distinct update rates. Ideally, one would fix θ, run iteration (6) until it converges to the true gradient, and then use this limit in iteration (7). However, such an approach is computationally inefficient. Instead, these coupled iterations are executed interactively, with iteration (6) running at a faster rate than (7), effectively treating θ as fixed in the second iteration. This timescale separation is achieved by ensuring that the step sizes satisfy: $\frac{\beta_k}{\alpha_k} \to 0$ as k tends to infinity, which guarantees the convergence of the algorithm (Li and Peng 2024). This design allows the gradient estimator's bias to average out over the iteration process, enabling accurate results even with a small Monte Carlo sample size N in Equation (5). Ultimately, ED_k converges to zero, and θ converges to its optimal value. The MTS framework for MLE is summarized as follows.

Algorithm 1 (MTS for MLE)

1: Input: data$\{Y_t\}_{t=1}^T$, initial iterative values θ_0, D_0, number of samples N, iterative steps K, the step-sizes α_k, β_k.
2: **for** k in $0:K-1$ **do**
3: For $i = 1:N$, sample X_i and get unbiased estimators $G_{1,k}(X_i,Y,\theta_k)$, $G_{2,k}(X_i,Y,\theta_k)$.
4: Do the iterations: $D_{k+1} = D_k + \alpha_k(G_{1,k}(X,Y,\theta_k) - G_{2,k}(X,Y,\theta_k)D_k)$, $\theta_{k+1} = \Pi_\Theta(\theta_k + \beta_k E D_k)$.
5: **end for**
6: Output: θ_K.

2.2 Posterior Density Estimation

We now turn to the problem of estimating the posterior distribution of the parameter θ in the stochastic model $Y = g(X,\theta)$, where the analytical likelihood is unknown. The posterior distribution is defined as

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{\int p(\theta)p(y|\theta)d\theta},$$

where $p(\theta)$ is the known prior distribution, and $p(y|\theta)$ is the conditional density function that lacks an analytical form but can be estimated using an unbiased estimator. The denominator is a challenging normalization constant to handle and variational inference is a practical approach.

In the variational inference framework, we approximate the posterior distribution $p(\theta|y)$ using a tractable density $q_\lambda(\theta)$ with a variational parameter λ to approximate. The collection $\{q_\lambda(\theta)\}$ is called the variational distribution family, and our goal is to find the optimal λ by minimizing the KL divergence between tractable variational distribution $q_\lambda(\theta)$ and the true posterior $p(\theta|y)$:

$$KL(\lambda) = KL(q_\lambda(\theta)\|p(\theta|y)) = \mathbb{E}_{q_\lambda(\theta)}[\log q_\lambda(\theta) - \log p(\theta|y)].$$

It is well known that minimizing KL divergence is equivalent to maximizing the ELBO, an expectation with respect to variational distribution $q_\lambda(\theta)$:

$$L(\lambda) = \log p(y) - KL(\lambda) = \mathbb{E}_{q_\lambda(\theta)}[\log p(y|\theta) + \log p(\theta) - \log q_\lambda(\theta)].$$

The problem is then reformulated as

$$\lambda^* = \arg\max_{\lambda\in\Lambda} L(\lambda),$$

where Λ is the feasible region of λ. It is essential to estimate the gradient of ELBO, which is an important problem in the field of machine learning and also falls under the umbrella of simulation optimization. Common methods for deriving gradient estimators include the score function method (Ranganath et al. 2014) and the re-parameterization trick (Kingma and Welling 2013; Rezende et al. 2014). In the simulation literature, these methods are also referred to as the likelihood ratio (LR) method and infinitesimal perturbation analysis (IPA) method, respectively (Fu 2006).

In this paper, $p(y|\theta)$ is estimated by simulation rather than computed precisely, inducing bias to the $\log p(y|\theta)$ term in LR method. Furthermore, the LR method is prone to high variance (Rezende et al. 2014), making the re-parameterization trick a preferred choice.

Assume a variable substitution involving λ, such that $\theta = \theta(u;\lambda) \sim q_\lambda(\theta)$, where u is a random variable independent of λ with density $p_0(u)$. This represents a re-parameterization of θ, where the stochastic component is incorporated into u, while the parameter λ is isolated. Allowing the interchange of differentiation and expectation (Glasserman 1990), we obtain

$$\begin{aligned}\nabla_\lambda L(\lambda) =& \nabla_\lambda \mathbb{E}_{q_\lambda(\theta)}[\log p(y|\theta) + \log p(\theta) - \log q_\lambda(\theta)] \\ =& \nabla_\lambda \mathbb{E}_u[\log p(y|\theta(u;\lambda)) + \log p(\theta(u;\lambda)) - \log q_\lambda(\theta(u;\lambda))] \\ =& \mathbb{E}_u[\nabla_\lambda \theta(u;\lambda)\cdot(\nabla_\theta \log p(y|\theta) + \nabla_\theta \log p(\theta) - \nabla_\theta \log q_\lambda(\theta))].\end{aligned} \tag{8}$$

In Equation (8), the Jacobi term $\nabla_\lambda \theta(u;\lambda)$, prior term $\log p(\theta)$ and variational distribution term $\log q_\lambda(\theta)$ are known. Therefore, the focus is on the term involving the intractable likelihood function. Similar to the MLE case, the term $\nabla_\theta \log p(y|\theta) = \frac{\nabla_\theta p(y|\theta)}{p(y|\theta)}$ contains the ratio of two estimators, which introduces bias.

The problem differs in two aspects. First, the algorithm no longer iterates over the parameter θ to be estimated but over the variational parameter λ, which defines the posterior distribution. This shifts the focus from point estimation to function approximation, aiming to identify the best approximation of the true posterior from the variational family $q_\lambda(\theta)$. Second, this becomes a nested simulation problem because the objective is ELBO, an expectation over a random variable u. Estimating its gradient requires an additional outer-layer simulation using SAA. In the outer layer simulation, we sample u to get the different θ, representing various scenarios. For each θ, the likelihood function and its gradient are estimated using the GLR method as in the MLE case, incorporating the MTS framework to reduce ratio bias. After calculating the part inside the expectation in Equation (8) for every sample u, we average the results with respect to u to get the estimator of the gradient of ELBO.

Note that the inner layer simulation for term $\nabla_\theta \log p(y|\theta) = \frac{\nabla_\theta p(y|\theta)}{p(y|\theta)}$ depends on u, so we need to fix outer layer samples $\{u_m\}_{m=1}^M$ at the beginning of the algorithm. Similar to the MLE case, M parallel

gradient iteration processes are defined as blocks $\{D_{k,m}\}_{m=1}^{M}$, where $D_{k,m}$ tracks the gradient of the likelihood function $\nabla_\theta \log p(y|\theta(u_m;\lambda_k))$ for every outer layer sample u_m. The optimization process of λ depends on the gradient of ELBO in Equation (8), which is estimated by averaging over these M blocks. An additional error arises between the true gradient of ELBO and its estimator due to outer-layer simulation. Unlike Algorithm 1, this approach involves a nested simulation optimization structure, where simulation and optimization are conducted simultaneously.

The nested MTS algorithm framework for the PDE problem is shown as Algorithm 2. $G_{1,k}(X,Y,\theta_{k,m})$ and $G_{2,k}(X,Y,\theta_{k,m})$ could be unbiased GLR estimators. The matrix dimensions are consistent with those in the MLE case. The iteration for $D_{k,m}$ resembles the MLE case, except for the parallel blocks. The iteration for λ_k corresponds to the gradient $\nabla_\lambda L(\lambda)$ in Equation (8).

Algorithm 2 (Nested MTS for PDE)

1: Input: data $\{Y_t\}_{t=1}^{T}$, prior $p(\theta)$, iteration initial value λ_0 and D_0, iteration times K, number of outer layer samples M, number of inner layer samples N, step-sizes α_k, β_k.
2: Sample $\{u_m\}_{m=1}^{M}$ from $p_0(u)$ as outer layer samples.
3: **for** k in $0:K-1$ **do**
4: $\theta_{k,m} = \theta(u_m;\lambda_k)$, for $m = 1:M$;
5: Sample $\{X_i\}_{i=1}^{N}$ and get the inner unbiased layer estimators $G_{1,k}(X,Y,\theta_{k,m})$, $G_{2,k}(X,Y,\theta_{k,m})$, for $i = 1:N$ and $m = 1:M$;
6: Do the iterations:

$$D_{k+1,m} = D_{k,m} + \alpha_k(G_{1,k}(X,Y,\theta_{k,m}) - G_{2,k}(X,Y,\theta_{k,m})D_{k,m}).$$

$$\lambda_{k+1} = \Pi_\Lambda\left(\lambda_k + \beta_k \frac{1}{M}\sum_{m=1}^{M}\left(\nabla_\lambda \theta(u;\lambda)\Big|_{(u;\lambda)=(u_m;\lambda_k)}\left(ED_{k,m} + \nabla_\theta \log p(\theta_{k,m}) - \nabla_\theta \log q_\lambda(\theta_{k,m})\right)\right)\right).$$

7: **end for**
8: Output: λ_K.

The following remark highlights the advantage of the MTS algorithm compared to the STS algorithm.

Remark 1 In the PDE case, the corresponding iterative process of STS is as below:

$$\lambda_{k+1} = \Pi_\lambda\left(\lambda_k + \beta_k \frac{1}{M}\sum_{m=1}^{M}\left(\nabla_\lambda \theta(u_m;\lambda_k)\left(\sum_{t=1}^{T}\frac{G_1(X,Y_t,\theta_{k,m})}{G_2(X,Y_t,\theta_{k,m})} + \nabla_\theta \log p(\theta_{k,m}) - \nabla_\theta \log q_\lambda(\theta_{k,m})\right)\right)\right). \tag{9}$$

In this previous way, we do not use D_k to track the gradient but plug in the ratio of two estimators whose bias may not be negligible if N is not large enough. Moreover, the estimator in the denominator makes the algorithm numerically unstable. Therefore, the gradient estimated in this algorithm is not precise so the optimization process is impacted. In Section 3, we will find that the STS algorithm does not perform as well as MTS.

2.3 MLE for the Hidden Markov Models

Generally, an HMM can be specified by the following general state space model: for $t = 1,\cdots,T$,

$$Y_t = g(W_t;S_t,\theta), \quad S_t = h(V_t;S_{t-1},\theta),$$

where $\{V_t\}_{t=1}^{T}$ are i.i.d. random variables driving the hidden underlying Markov chain $\{S_t\}_{t=1}^{T}$ with initial state S_0. The model dynamics is governed by some parameter θ belonging to some parameter space Θ. $\{W_t\}_{t=0}^{T}$ are i.i.d. random variables introducing interference to the unobservable state S_t of the Markov chain. Only $\{Y_t\}_{t=1}^{T}$ are observable. For given observation data $\{Y_t\}_{t=1}^{T}$, the log-likelihood of observations

following an HMM is given by

$$L_T(\theta) \doteq \log \mathbb{E}\left[\prod_{t=1}^{T} p_\theta(Y_t; S_t)\right], \tag{10}$$

where $p_\theta(\cdot; S_t)$ is the conditional density of observation Y_t on hidden state S_t and the expectation is taken w.r.t S_t. The asymptotic properties of the MLE for an HMM are similar to the i.i.d. case and can be found in Cappé et al. (2005), chap. 6.

To derive the gradient estimator $\nabla_\theta L_T(\theta)$ and reduce its variance, we need to construct a consecutive update of the prior distribution by incorporating information from observations sequentially and sample from the posterior. We decompose the log-likelihood into a sum of log conditional expectations:

$$L_T(\theta) = \sum_{t=0}^{T-1} \log\left(\mathbb{E}\left[\prod_{l=1}^{t+1} p(Y_l; S_l, \theta)\right] \Big/ \mathbb{E}\left[\prod_{l=1}^{t} p(Y_l; S_l, \theta)\right]\right) \doteq \sum_{t=0}^{T-1} \log \pi_{t+1|t}(p_\theta(S_{t+1}; Y_{t+1})),$$

where $\prod_l^0 \doteq 1$ and $\pi_{t+1|t}(p_\theta(S_{t+1}; Y_{t+1})) = \mathbb{E}[p_\theta(S_{t+1}; Y_{t+1})|Y_{1:t}]$. The gradient of the log-likelihood $L_T(\theta)$ becomes

$$\nabla_\theta L_T(\theta) = \sum_{t=0}^{T-1} \frac{\nabla \pi_{t+1|t}(p_\theta(S_{t+1}; Y_{t+1}))}{\pi_{t+1|t}(p_\theta(S_{t+1}; Y_{t+1}))}. \tag{11}$$

Take the derivative and we can get

$$\nabla \pi_{t+1|t}(p_\theta(S_{t+1}; Y_{t+1})) = \frac{\nabla \mathbb{E}[p_\theta(S_{t+1}; Y_{t+1}) \prod_{k=0}^{t} p_\theta(S_k; Y_k)]}{\mathbb{E}[\prod_{k=0}^{t} p_\theta(S_k; Y_k)]} - \pi_{t+1|t}(p_\theta(S_{t+1}; Y_{t+1})) \frac{\nabla \mathbb{E}[\prod_{k=0}^{t} p_\theta(S_k; Y_k)]}{\mathbb{E}[\prod_{k=0}^{t} p_\theta(S_k; Y_k)]},$$

where

$$\nabla \mathbb{E}\left[\prod_{k=0}^{t+1} p_\theta(S_k; Y_k)\right] = \mathbb{E}\left[\left(\nabla p_\theta(S_{t+1}; Y_{t+1}) + p_\theta(S_{t+1}; Y_{t+1}) \sum_{k=0}^{t} \frac{\nabla p_\theta(S_k; Y_k)}{p_\theta(S_k; Y_k)}\right) \prod_{k=0}^{T} p_\theta(S_k; Y_k)\right].$$

We define $Z_t = \frac{\partial S_t}{\partial \theta}$ and set the augmented Markov chain $(S_t, Z_t, W_t)_{t \geq 0}$ by the following recursive relationship:

$$S_{t+1} = h(V_{t+1}; S_t, \theta),\ Z_{t+1} = \frac{\partial S_{t+1}}{\partial \theta} = \frac{\partial h}{\partial \theta} + \frac{\partial h}{\partial S_t} Z_t,$$

$$W_{t+1} = W_t + \frac{\frac{\partial}{\partial S_t} p_\theta(S_{t+1}; Y_{t+1}) Z_{t+1} + \frac{\partial}{\partial \theta} p_\theta(S_{t+1}; Y_{t+1})}{p_\theta(S_{t+1}; Y_{t+1})}. \tag{12}$$

Then based on the SMC method, $\pi_{t+1|t}(p_\theta(S_{t+1}; Y_{t+1}))$ can be estimated by the consistent estimator $\frac{1}{J}\sum_{j=1}^{J} p_\theta(\hat{S}_{t+1}^j; Y_{t+1})$. And $\nabla \pi_{t+1|t}(p_\theta(S_{t+1}; Y_{t+1}))$ can be estimated by the consistent estimator $\sum_{j=1}^{J} \nabla_\theta p_\theta(\hat{S}_t^j; Y_t) + p_\theta(\hat{S}_t^j; Y_t)(W_{t-1}^j - \frac{1}{J}\sum_{j'} W_{t-1}^{j'})$. We deduce the IPA estimator of $\nabla L_T(\theta)$:

$$\sum_{t=1}^{T} \frac{\sum_{j=1}^{J} \nabla_\theta p_\theta(\hat{S}_t^j; Y_t) + p_\theta(\hat{S}_t^j; Y_t)(W_{t-1}^j - \frac{1}{J}\sum_{j'} W_{t-1}^{j'})}{\sum_{j=1}^{J} p_\theta(\hat{S}_t^j; Y_t)}, \tag{13}$$

where $(\hat{S}_t^j, Z_t^j, W_{t-1}^j)$ are particles derived by using a SMC algorithm on the augmented Markov chain.

Noting that Equation (13) contains the ratio of two estimators, we apply the GSPE algorithm and design an additional iteration $\{D_k\}$ to track this gradient in Equation (11). The algorithm framework we propose is shown in Algorithm 3. In the simulation, we sample different hidden states $\hat{S}_t^j$ for $j = 1, \cdots, J$ by transition density $p(S_t^j | S_{t-1}^j, \theta)$ to obtain different particles for every observation $t = 1, \cdots, T$. Then

we use the observation density $p(Y_t|S_t^j,\theta)$ to calculate the likelihood function and its derivatives in the corresponding hidden states, and assign different weights to different particles by comparing them with the real observations. Afterward, resampling is performed to prevent particle degradation caused by uneven weights. Then the numerator and the denominator of Equation (11) can be approximated by the weighted average of all the J particles through Equation (13) separately, denoted as $G_{1,k}$ and $G_{2,k}$.

These two iterations operate on different time scales, with distinct update rates: $\frac{\beta_k}{\alpha_k} \to 0$ as k tends to infinity. The update rule for $\{D_k\}$ is based on a fixed-point principle, mitigating ratio bias and numerical instability caused by denominator estimators. This design allows the gradient estimator's bias to average out over the iteration process, enabling accurate results even with a small particle number J. The estimation of the gradient of the likelihood function is finally obtained by the gradient ascent in the second time scale. The algorithm framework we propose is as follows.

Algorithm 3 (MTS for the MLE in HMMs)

1: Input: data $\{Y_t\}_{t=1}^T$, initial iterative values θ_0, D_0, number of particles J, iterative steps K, the step-size α_k, β_k.
2: initialization: $D_0 = 0$, $w_0^j = 1/J$, for every $j = 1,\cdots,J$.
3: **for** k in $1:K$ **do**
4: **for** t in $1:T$ **do**
5: sample V_t^j and get new state by $S_t^j = h(V_t^j; S_{t-1}^j, \theta_k)$, $j = 1,\cdots,J$;
6: calculate the conditional density of every particle and their derivative: for $j = 1,\cdots,J$,

$$\Phi_{1,t,k}^j(Y_t,S_t^j,\theta_k) = p(Y_t|S_t^j,\theta_k), \quad \Phi_{2,t,k}^j(Y_t,S_t^j,\theta_k) = \frac{\partial p(Y_t|s,\theta)}{\partial\theta}\bigg|_{s=S_t^j,\theta=\theta_k},$$

$$\Phi_{3,t,k}^j(Y_t,S_t^j,\theta_k) = \frac{\partial p(Y_t|s,\theta)}{\partial s}\bigg|_{s=S_t^j,\theta=\theta_k}, \quad \Phi_{4,t,k}^j = \frac{\partial h(V_t^j;s,\theta)}{\partial\theta}\bigg|_{s=S_{t-1}^j,\theta=\theta_k}.$$

7: calculate the estimator of the numerator and denominator of the SMC:

$$G_{1,k}(Y,\theta_k)(t) = \sum_{j=1}^J \Phi_{1,t,k}^j \left(\sum_{l=1}^t \frac{\Phi_{2,l,k}^j + \Phi_{3,l,k}^j \Phi_{4,l,k}^j}{\Phi_{1,l,k}^j} - \frac{1}{J} \sum_{j'=1}^J \sum_{l=1}^{t-1} \frac{\Phi_{2,l,k}^{j'} + \Phi_{3,l,k}^{j'} \Phi_{4,l,k}^{j'}}{\Phi_{1,l,k}^{j'}} w_t^{j'} \right) w_t^j, \quad G_{2,k}(Y,\theta_k)(t) = \sum_{j=1}^J \Phi_{1,t,k}^j w_t^j.$$

8: If $ESS := \left(\sum_{j=1}^J (w_t^j)^2 \right)^{-1} < J/3$, calculate the importance weight of every particle and update it:

$$w_t^j = \frac{\Phi_{1,t,k}^j w_t^j}{\sum_{j=1}^J \Phi_{1,t,k}^j w_t^j}, \quad j = 1,\cdots,J,$$

9: else: using polynomial resampling to resample S_t^j with probability of w_t^j, i.e. sample $S_t^j = S_t^{\xi_j}$ with $\xi_j = i \in 1,\cdots,J$ w.p. w_t^j. Then reset the $w_t^j = 1/J$.
10: **end for**
11: Do the iterations:

$$D_{k+1} = D_k + \alpha_k (G_{1,k}(Y,\theta_k) - G_{2,k}(Y,\theta_k) \cdot D_k), \quad \theta_{k+1} = \Pi_\Theta(\theta_k + \beta_k E D_k).$$

12: **end for**
13: Output: θ_{K+1}.

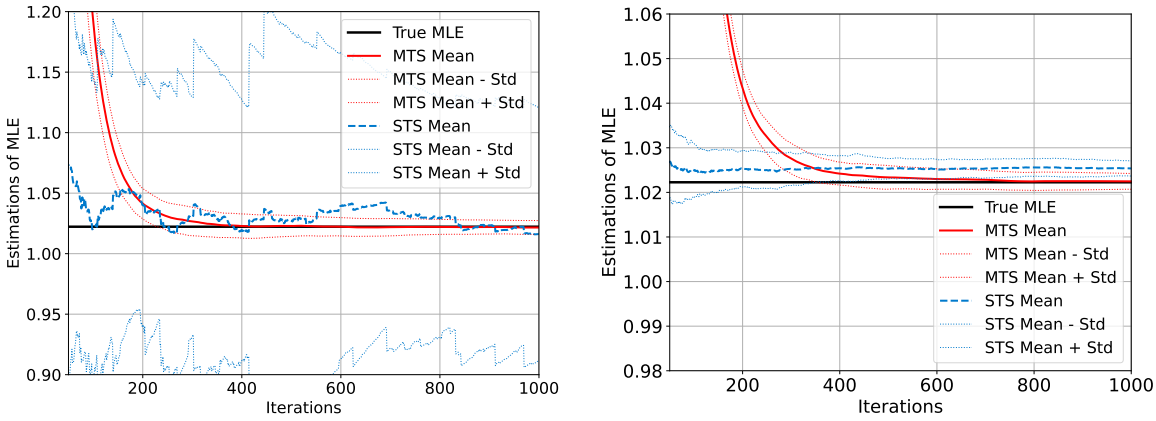
3 NUMERICAL EXPERIMENTS

In this section, we demonstrate the application of the GSPE algorithm framework, comprising two specific algorithms, to various cases. Section 3.1 addresses the MLE case, while Section 3.2 focuses on the PDE case. Section 3.3 is a simple HMM case.

3.1 MLE Case

We apply Algorithm 1 to evaluate the MTS framework in the MLE setting. Consider i.i.d. observations generated by the data-generating process $Y_t = g(X_t;\theta) = X_{1,t} + \theta X_{2,t}$, where $X_{1,t}, X_{2,t} \sim N(0,1)$ are independent. Y_t is observable but X_t is latent variable. The goal is to estimate θ based on observation $\{Y_t\}_{t=1}^T$. For this example, the MLE has an analytical form: $\hat{\theta} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} Y_t^2 - 1}$.

The true value θ is set to be 1. The faster and slower step-size is chosen as $\frac{10}{k^{0.55}}$ and $\frac{0.5}{k}$, respectively, which satisfies the step-size condition of the MTS algorithm. We set $T = 100$ observations, the feasible region $\Theta = [0.5, 2]$, and the initial value $\theta_0 = 0.8$. The samples of $X_t = (X_{1,t}, X_{2,t})$ are simulated to estimate the likelihood function and its gradient at each iteration. We compare our MTS algorithm with the STS method. In previous works, a large number of simulated samples per iteration (e.g., 10^5) is required to ensure a negligible ratio bias from the log-likelihood gradient estimator. By employing our method, computational costs are reduced while improving estimation accuracy. Figure 3.1(a) exhibits the convergence results of MTS and STS with $N = 10^4$ simulated samples based on 100 independent experiments. Compared to the true MLE, MTS achieves lower bias and standard error than STS. The convergence curve is also more stable due to the elimination of the denominator estimator. The average CPU time per experiment for MTS and STS is 0.7s and 0.72s, respectively, indicating comparable computational costs. Figure 3.1(b) depicts the convergence result with 10^5 simulated samples based on 100 independent experiments. Even with a large number of simulated samples, MTS outperforms STS. Table 1 records the absolute bias for the two estimators under their respective optimal allocation policies and the choice of step-size (Li and Peng 2024), based on 100 independent experiments. N is the batchsize, K is the iteration size, and Γ is the total budget. Across all budget levels, MTS demonstrates significantly higher estimation accuracy than STS.



(a) Convergence curves with $N = 10^4$

(b) Convergence curves with $N = 10^5$

Figure 1: Trajectories of MTS and STS with different sample sizes based on 100 independent experiments.

3.2 PDE Case

We apply Algorithm 2 to test the nested MTS framework in the PDE setting. Let the prior distribution of the parameter θ be the standard normal $N(0,1)$. The stochastic model is $Y_t = X_t + \theta$ with latent variable $X_t \sim N(0,1)$. Given the observation $y = \{Y_t\}_{t=1}^T$, the goal is to compute the posterior distribution for θ. It is straightforward to derive that the analytical posterior is $p(\theta|y) \sim N(\frac{n}{1+n}\bar{y}, \frac{1}{1+n})$.

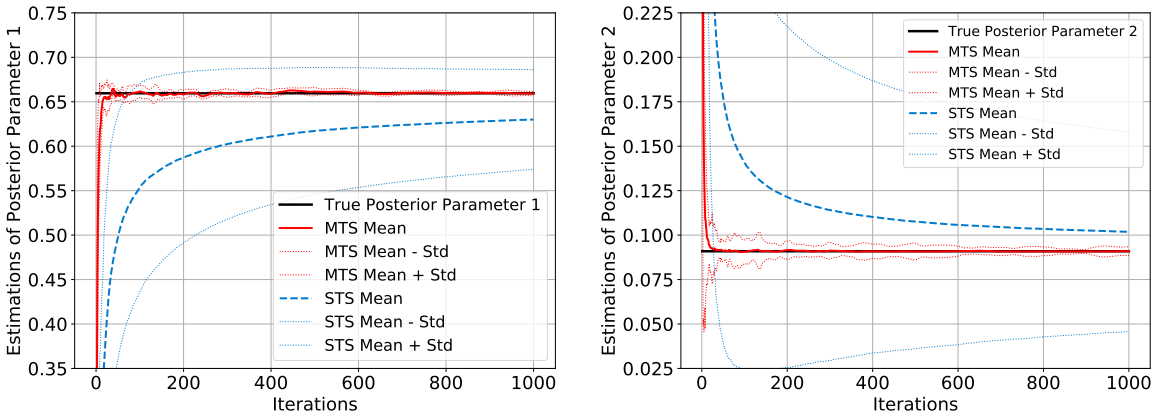
Let the posterior parameter λ be (μ, σ^2). We want to use normal distribution $q_\lambda(\theta)$ to approximate the posterior of θ, i.e., $q_\lambda(\theta) \sim N(\mu, \sigma^2)$. Applying the re-parameterization technique, we can sample u

Table 1: The absolute bias of the two estimators and true MLE, based on 100 independent experiments.

Γ	N (K for STS)	K (N for STS)	**Absolute Bias $\pm$ std**	
			MTS	STS
10^4	86	116	$1.9\times10^{-2}\pm2.2\times10^{-1}$	$1.5\times10^{-1}\pm3.9\times10^{-1}$
3×10^4	124	241	$1\times10^{-2}\pm8\times10^{-2}$	$1\times10^{-1}\pm3.8\times10^{-1}$
10^5	186	539	$2.3\times10^{-3}\pm8\times10^{-2}$	$6.4\times10^{-2}\pm3.6\times10^{-1}$
3×10^5	268	1120	$1.5\times10^{-3}\pm3.9\times10^{-2}$	$2.9\times10^{-2}\pm2.8\times10^{-1}$
10^6	400	2500	$4.8\times10^{-4}\pm2.2\times10^{-2}$	$7.3\times10^{-3}\pm2.8\times10^{-1}$
3×10^6	577	5200	$3\times10^{-4}\pm1.3\times10^{-2}$	$3.4\times10^{-3}\pm2.6\times10^{-1}$
10^7	862	11604	$2\times10^{-4}\pm8.1\times10^{-3}$	$2.1\times10^{-3}\pm2.9\times10^{-1}$
3×10^7	1243	24137	$1.6\times10^{-4}\pm4.7\times10^{-3}$	$1.9\times10^{-3}\pm1.8\times10^{-1}$
10^8	1857	53861	$5.9\times10^{-5}\pm2.1\times10^{-3}$	$1\times10^{-3}\pm1.2\times10^{-1}$

from normal distribution $N(0,1)$ and set $\theta(u;\lambda)=\mu+\sigma u\sim N(\mu,\sigma^2)$. Here is just an illustrative example of normal distribution, re-parameterization technique can be applied to other more general distributions (Figurnov et al. 2018; Ruiz et al. 2016).

In the PDE case, we can incorporate the data into prior over and over again. Suppose there are only 10 independent observations for one batch. Set feasible region $\Lambda=[-1,10]\times[0.01,2]$ and initial value $\lambda_0=(0,1)$. First, we set $M=10$ outer layer samples u_m and compare the MTS algorithm with the analytical posterior and STS method. The faster and slower step-size is chosen as $\frac{10}{k^{0.55}}$ and $\frac{1}{k}$, respectively. Figure 3.2 displays the trajectories of MTS and STS with sample size 10^4 based on 100 independent experiments. Specifically, Figure 3.2(a) exhibits the convergence for the posterior mean μ and Figure 3.2(b) exhibits the convergence for the posterior variance σ^2. MTS achieves lower bias and standard error than STS when compared to the true posterior parameters.



(a) Estimations of posterior mean

(b) Estimations of posterior variance

Figure 2: Trajectories of MTS and STS with sample size 10^4 based on 100 independent experiments.

Table 2 records the absolute error for both estimators under their respective optimal allocation policies, based on 100 independent experiments. Across all budget levels, MTS consistently outperforms STS in estimation accuracy.

3.3 MLE for the HMM

We illustrate our approach through the following example. The initial value of the hidden state is set as $S_0=0$. The transition kernel is defined as $S_t=S_{t-1}+\theta+V_t$, and the observation kernel is given by

Table 2: The absolute bias of the two estimators, based on 100 independent experiments.

Γ	M	N (K for STS)	K (N for STS)	**Posterior Mean**		**Posterior Variance**	
				MTS	STS	MTS	STS
10^5	4	214	106	2.3×10^{-3}	8.3×10^{-2}	5.5×10^{-3}	5.9×10^{-2}
3×10^5	5	281	183	1.1×10^{-3}	4.5×10^{-2}	2.8×10^{-3}	2×10^{-2}
10^6	7	380	334	5.2×10^{-4}	1.7×10^{-2}	6.8×10^{-4}	6.2×10^{-3}
3×10^6	10	500	578	3.0×10^{-4}	1.2×10^{-2}	1.0×10^{-4}	2.3×10^{-3}
10^7	14	675	1055	1.9×10^{-4}	5×10^{-3}	9.4×10^{-5}	6×10^{-4}
3×10^7	18	889	1826	5×10^{-5}	4.1×10^{-3}	4.9×10^{-5}	3.7×10^{-4}
10^8	25	1200	3334	4.2×10^{-5}	2.3×10^{-3}	1.5×10^{-5}	2.7×10^{-4}
3×10^8	32	1580	5774	1.3×10^{-5}	1.4×10^{-3}	5.2×10^{-6}	8.1×10^{-5}
10^9	44	2134	10561	1×10^{-5}	8.1×10^{-4}	6.6×10^{-6}	4.2×10^{-5}

$Y_t = S_t + W_t$, where S_t denotes the hidden state, while W_t and V_t represent the observation error and transition error, respectively. Here, $W_t \sim N(0,1)$ and $V_t \sim N(0,1)$, and both W_t and V_t are mutually independent and identically distributed (i.i.d.). Our goal is to estimate the parameter θ using the observations $\{Y_t\}_{t=1}^T$.

In this example, we set $T = 100$ and use $J = 10^3$ or 10^4 particles. The step sizes are chosen as $\frac{100}{K^{0.8}}$ and $\frac{0.1}{K}$, respectively. We conduct 20 independent experiments, replacing the GLR estimators with direct computation of particle weights using the observation kernel. By comparing the results presented in Table 3, we conclude that our method significantly reduces bias in the HMM.

Table 3: The mean absolute error of the two estimators based on 20 independent experiments.

Particle Numbers	**Mean Absolute Error**	
	MTS	STS
100	0.0307	0.0427
1000	0.0104	0.0145

4 CONCLUSION

This paper presents a comprehensive study addressing the challenges of parameter estimation where the likelihood function is estimated by simulations. Our GSPE approach, grounded in the MTS algorithm, handles the ratio bias problem, enhances the accuracy of parameter estimation, and saves computational costs. In the realm of PDE, we have explored a nested simulation optimization structure, which is both theoretically sound and empirically effective.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 72325007, 72250065, and 72022001.

REFERENCES

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe. 2017. "Variational Inference: A Review for Statisticians". *Journal of the American Statistical Association* 112(518):859–877.

Borkar, V. S. 2009. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Berlin: Springer.

Cao, H., J. Hu, and J. Hu. 2023. "Black-Box CoVaR and Its Gradient Estimation". *SSRN Electronic Journal*.

Cappé, O., É. Moulines, and T. Rydén. 2005. *Inference in Hidden Markov Models*. Berlin: Springer.

Doucet, A., N. De Freitas, and N. J. Gordon. 2001. *Sequential Monte Carlo methods in practice*. Berlin: Springer.

Figurnov, M., S. Mohamed, and A. Mnih. 2018. "Implicit Reparameterization Gradients". In *Advances in Neural Information Processing Systems*. December 2nd-8th, Montréal, Canada, 439–450.

Fu, M. C. 2006. "Gradient Estimation". *Handbooks in Operations Research and Management Science* 13:575–616.
Fu, M. C., L. J. Hong, and J.-Q. Hu. 2009. "Conditional Monte Carlo Estimation of Quantile Sensitivities". *Management Science* 55(12):2019–2027.
Glasserman, P. 1990. *Gradient Estimation via Perturbation Analysis*. Berlin: Springer Science & Business Media.
Glöckler, M., M. Deistler, and J. H. Macke. 2022. "Variational Methods for Simulation-Based Inference". *arXiv preprint arXiv:2203.04176*.
Glynn, P. W., Y. Peng, M. C. Fu, and J.-Q. Hu. 2021. "Computing Sensitivities for Distortion Risk Measures". *INFORMS Journal on Computing* 33(4):1520–1532.
Hong, M., H.-T. Wai, Z. Wang, and Z. Yang. 2023. "A Two-Timescale Stochastic Algorithm Framework for Bilevel Optimization: Complexity Analysis and Application to Actor-critic". *SIAM Journal on Optimization* 33(1):147–180.
Hu, J., Y. Peng, G. Zhang, and Q. Zhang. 2022. "A Stochastic Approximation Method for Simulation-Based Quantile Optimization". *INFORMS Journal on Computing* 34(6):2889–2907.
Hu, J., M. Song, and M. C. Fu. 2025. "Quantile Optimization via Multiple-Timescale Local Search for Black-Box Functions". *Operations Research* 73(3):1535–1557.
Jiang, J., Y. Peng, and J. Hu. 2022. "Quantile-Based Policy Optimization for Reinforcement Learning". In *2022 Winter Simulation Conference (WSC)*, 2712–2723 https://doi.org/10.1109/WSC57314.2022.10015456.
Khodadadian, S., T. T. Doan, J. Romberg, and S. T. Maguluri. 2022. "Finite-Sample Analysis of Two-Time-Scale Natural Actor–critic Algorithm". *IEEE Transactions on Automatic Control* 68(6):3273–3284.
Kingma, D. P., and M. Welling. 2013. "Auto-Encoding Variational Bayes". *arXiv preprint arXiv:1312.6114*.
Kushner, H. J., and G. G. Yin. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. Berlin: Springer.
Lei, L., Y. Peng, M. C. Fu, and J. Hu. 2018. "Applications of Generalized Likelihood Ratio Method to Distribution Sensitivities and Steady-state Simulation". *Discrete Event Dynamic Systems* 28:109–125.
Li, Z., and Y. Peng. 2024. "Eliminating Ratio Bias for Gradient-Based Simulated Parameter Estimation". *arXiv preprint arXiv:2411.12995*.
Lin, T., C. Jin, and M. I. Jordan. 2025. "Two-Timescale Gradient Descent Ascent Algorithms for Nonconvex Minimax Optimization". *Journal of Machine Learning Research* 26(11):1–45.
Papamakarios, G., D. Sterratt, and I. Murray. 2019. "Sequential Neural Likelihood: Fast Likelihood-Free Inference with Autoregressive Flows". In *International Conference on Artificial Intelligence and Statistics*. April 16th–18th, Okinawa, Japan, 837–848.
Peng, Y., M. C. Fu, B. F. Heidergott, and H. Lam. 2020. "Maximum Likelihood Estimation by Monte Carlo Simulation: Toward Data-Driven Stochastic Modeling". *Operations Research* 68:1896–1912.
Peng, Y., M. C. Fu, J.-Q. Hu, and B. Heidergott. 2018. "A New Unbiased Stochastic Derivative Estimator for Discontinuous Sample Performances with Structural Parameters". *Operations Research* 66(2):487–499.
Ranganath, R., S. Gerrish, and D. M. Blei. 2014. "Black Box Variational Inference". In *Artificial Intelligence and Statistics*. April 22nd–25th, Reykjavik, Iceland, 814–822.
Rezende, D. J., S. Mohamed, and D. Wierstra. 2014. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In *International Conference on Machine Learning*. June 21st–26th, Beijing, China, 1278–1286.
Ruiz, F. R., M. K. Titsias, and D. M. Blei. 2016. "The Generalized Reparameterization Gradient". In *Advances in Neural Information Processing Systems*. December 5th–10th, Barcelona, Spain, 460–468.
Shao, J. 2003. *Mathematical Statistics*. Berlin: Springer Science & Business Media.
Wills, A. G., and T. B. Schön. 2023. "Sequential Monte Carlo: A Unified Review". *Annual Review of Control, Robotics, and Autonomous Systems* 6(1):159–182.
Zheng, Y., Z. Li, P. Jiang, and Y. Peng. 2024. "Dual-Agent Deep Reinforcement Learning for Dynamic Pricing and Replenishment". *arXiv preprint arXiv:2410.21109*.

AUTHOR BIOGRAPHIES

ZEHAO LI is a PhD candidate in the Department of Management Science and Information Systems in Guanghua School of Management at Peking University, Beijing, China. He received the BS degree in School of Mathematical Sciences, Fudan University in 2023. His research interests include simulation optimization and machine learning. His email address is zehaoli@stu.pku.edu.cn.

YIJIE PENG is an associate professor in Guanghua School of Management at Peking University. His research interests include stochastic modeling and analysis, simulation optimization, machine learning, data analytics, and healthcare. He is a member of INFORMS and IEEE and serves as an Associate Editor of the Asia-Pacific Journal of Operational Research and the Conference Editorial Board of the IEEE Control Systems Society. His email address is pengyijie@pku.edu.cn.