



Beyond Likelihood Ratio Bias: Nested Multi-time Scale Stochastic Approximation for Likelihood-Free Parameter Estimation

Zehao Li¹ · Zhouchen Lin² · Yijie Peng^{3,4}

Received: 11 March 2026 / Revised: 2 August 2026 / Accepted: 20 September 2026

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2026

Abstract

We study parameter inference in simulation-based stochastic models where the analytical form of the likelihood is unknown. The main difficulty is that score evaluation as a ratio of noisy Monte Carlo estimators induces bias and instability, which we overcome with a ratio-free nested multi-time-scale (NMTS) stochastic approximation (SA) method that simultaneously tracks the score and drives the parameter update. We provide a comprehensive theoretical analysis of the proposed NMTS algorithm for solving likelihood-free inference problems, including strong convergence, asymptotic normality, and convergence rates. We show that our algorithm can eliminate the original asymptotic bias $O(\sqrt{\frac{1}{N}})$ and accelerate the convergence rate from $O(\beta_k + \sqrt{\frac{1}{N}})$ to $O(\frac{\beta_k}{\alpha_k} + \sqrt{\frac{\alpha_k}{N}})$, where N is the fixed batch size, α_k and β_k are decreasing step sizes with $\alpha_k, \beta_k, \beta_k/\alpha_k \rightarrow 0$. With proper choice of α_k and β_k , our convergence rates can match the optimal rate in the multi-time-scale SA literature. Numerical experiments demonstrate that our algorithm can improve the estimation accuracy by one to two orders of magnitude at the same computational cost, making it efficient for parameter estimation in stochastic systems.

Keywords Stochastic approximation · Likelihood-free estimation · Ratio bias · Convergence analysis

1 Introduction

In statistical inference, likelihood-free parameter estimation (LFPE) refers to the case where the analytical form of the likelihood function is not available, and how to infer the model parameters based on observed data becomes difficult. Key inference methods include maximum likelihood estimation (MLE), which obtains a point estimate by maximizing the likelihood, and posterior density estimation (PDE), which views inference as density

Editor: Quanming Yao.

Extended author information available on the last page of the article

approximation and typically uses variational inference to select the distribution closest to the target posterior within a chosen variational family.

Traditional density estimation focuses on approximating an unknown distribution $p_{\text{data}}(x)$ with a model $p_{\theta}(x)$ parameterized by θ . Depending on whether the likelihood is tractable, existing approaches include explicit likelihood models such as exponential families and normalizing flows Papamakarios et al. (2021), unnormalized energy models trained via score matching or Stein operators Hyvärinen and Dayan (2005); Anastasiou et al. (2023), and implicit generative models such as GANs and diffusion models Puchkin et al. (2024). Despite methodological diversity, these approaches share a common goal: learning a density over data. In contrast, this paper addresses a different class of inference problems, where the target of learning is not the density of data itself, but the parameters of a stochastic system that implicitly generates observable data.

This paper focuses on stochastic models or simulators characterized by system dynamics rather than explicit likelihood functions. Examples include system dynamics models, complex network models, and Lindley's recursion in queueing models. In such systems, the likelihood function of the observed data $p(Y; \theta)$ is intractable, posing a major obstacle for parameter calibration and statistical inference Gross et al. (2011); Shepherd (2014); Peng et al. (2020). Estimating or optimizing parameters in this setting requires differentiating the implicit likelihood with respect to θ , leading to the score function $\nabla_{\theta} \log p(y; \theta)$. However, when only Monte Carlo estimators of $\nabla_{\theta} p(y; \theta)$ and $p(y; \theta)$ are available, the ratio $\nabla_{\theta} p(y; \theta) / p(y; \theta)$ introduces a systematic ratio bias that hampers convergence and stability.

Existing studies have investigated similar ratio-estimation problems from different perspectives. For instance, density-ratio estimation methods directly learn q/p from samples drawn from two data distributions Sugiyama et al. (2010, 2012a, 2012b); Thomas et al. (2022), and score-based or density-derivative-ratio estimators learn normalized derivatives like $\nabla_x p(x) / p(x)$ to characterize data-space geometry Sasaki et al. (2017, 2018). These approaches rely on access to data-space gradients or two well-defined distributions in x , and the ratio is computed analytically within a deterministic objective. In contrast, our ratio $\nabla_{\theta} p_{\theta}(y) / p_{\theta}(y)$ resides in the parameter domain, and both numerator and denominator are obtained through Monte Carlo simulation. Consequently, existing ratio-estimation techniques cannot be applied, since their unbiasedness and convergence rely on exact evaluations or deterministic gradients, whereas in our case, the two components are stochastic and correlated, making the direct division intrinsically biased. This motivates our development of a ratio-free nested multi-time-scale (NMTS) stochastic approximation (SA) scheme that recursively tracks the desired parameter-space score without explicit division.

Therefore, we cast the inference problem as a stochastic optimization task, jointly performing estimation and optimization Harold et al. (1997); Borkar (2009). The core idea is to treat both the parameters and the gradient of the logarithmic likelihood function jointly as components of a stochastic root-finding problem for a coupled system of nonlinear equations. 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 a gradient estimator. This method continually refines the gradient estimator by averaging all available simulation data, thereby eliminating ratio bias throughout the iterative process.

Building on the rich SA literature on the vanilla MTS methods Borkar (2009); Harold et al. (1997); Doan (2022); Hu et al. (2022); Hong et al. (2023); Hu et al. (2024); Zeng et al.

(2024); Lin et al. (2025); Cao et al. (2025, 2026); Wang et al. (2026), which provides a versatile and well-understood foundation, we develop an NMTS scheme expressly tailored to the structural demands of our likelihood-free inference problem. In our setting, variational inference is used to perform density estimation by maximizing the evidence lower bound (ELBO) Blei et al. (2017), whose unbiased gradient estimator appears as an expectation involving a likelihood ratio. To evaluate this outer expectation, we first adopt the sample average approximation (SAA) to draw a finite set of outer samples. Then, for each outer sample, we design a set of parallel fast-timescale recursions that update local estimates of the likelihood-ratio gradient. These parallel recursions are subsequently coupled into the slow-timescale parameter update through a weighted averaging mechanism. This structure allows the fast-timescale to track the local ratio estimates, while the inner recursions continuously update the parameters by the overall ELBO gradient represented by the combination of parallel fast-timescale recursions. In this way, we effectively extend the classical multi-time-scale SA to a nested and parallel setting, specifically adapted to intractable likelihood simulators.

We also provide a comprehensive theoretical analysis of the proposed NMTS algorithm, including almost-sure convergence, asymptotic normality, and the $\mathbb{L}^1$ convergence rate. The main technical challenge arises from the nested structure of the algorithm, where the inner recursions evolve on different timescales while the outer recursion depends on an SAA-based approximation of the ELBO gradient. In the strong convergence analysis, the introduction of outer samples requires proving that the SAA gradient estimator remains uniformly consistent when the inner parameter λ_k changes during the iteration, a result guaranteed by Donsker's Theorem in Proposition 2. Theorems 4 and 6 respectively establish the uniform convergence of the parallel fast-timescale recursions and the convergence of the slow-timescale updates. Our proof constructs a sequence of continuous time interpolations of the discrete iterates and shows that they are asymptotically governed by a limiting system of ordinary differential equations (ODEs). The stationary point of this ODE system coincides with the almost sure limit of the NMTS iterates, thereby extending convergence results in prior work to the nested and high-dimensional setting considered here. Proposition 7 and 8 further establish that the convergence results hold for the two-layer structure of the algorithm, extending the previous pioneering SA literature Doan (2022); Hong et al. (2023); Zeng et al. (2024); Lin et al. (2025); Hu et al. (2022, 2024); Cao et al. (2025).

For the weak convergence results, we build upon the framework in Mokkadem and Pelletier (2006) to analyze the weak limits of both the inner and outer stochastic processes. Theorem 11 characterizes the weak convergence of the NMTS iterates, while Theorems 13 and 14 provide the weak convergence and corresponding rates for the outer-layer SAA sequence. Finally, the $\mathbb{L}^1$ convergence rate analysis in Theorem 16 shows that, for a fixed batch size N , the NMTS algorithm achieves $O(\frac{\beta_k}{\alpha_k} + \sqrt{\frac{\alpha_k}{N}})$, where α_k and β_k are decreasing step sizes satisfying $\beta_k, \alpha_k, \beta_k/\alpha_k \rightarrow 0$. With an appropriate choice of step-size schedule, the optimal $\mathbb{L}^1$ rate is $O(k^{-1/3})$, which coincides with the convergence rate in the literature Doan (2022); Hong et al. (2023). In contrast, conventional single-timescale algorithms suffer from a persistent asymptotic bias of order $O(\sqrt{1/N})$ due to the ratio bias in gradient estimation. Our NMTS framework removes this bias through the coupled fast-timescale recursive estimator, thereby yielding a provably faster and consistent convergence behavior.

Furthermore, we introduce the concept of different timescales into neural network training to demonstrate the compatibility and scalability of our NMTS framework. For overly complex simulators, we use a neural network as an alternative to estimate the intractable likelihood. Additionally, when the posterior is complex and the simple variational distribution family has limited representational ability, another neural network can serve as the variational distribution. We design an NMTS algorithm that adjusts the update frequency of the two neural networks to improve training outcomes. Under a diminishing-bias assumption on the learned likelihood model, we show how approximation error at the fast timescale propagates to the slow-timescale update. This extension is further illustrated empirically, highlighting the practical compatibility of the NMTS principle with neural approximators. Our approach provides theoretical support for such estimation and optimization algorithms that require updates at different frequencies. More generally, this offers a new SA perspective on neural network training at various scales.

We summarize our main contributions as follows:

- A ratio-free nested multi-time-scale (NMTS) algorithm: We introduce a new NMTS algorithm that jointly addresses MLE and PDE problems when the likelihood function is intractable. The new NMTS SA scheme transforms the ratio estimator into a coupled root-finding system with two interacting time scales. The fast recursions track local score estimates through parallel updates, while the slow recursion aggregates these estimates to optimize the ELBO or likelihood objective. This structure extends classical multi-time-scale SA theory to a nested and parallel regime tailored to likelihood-free simulators.
- Theoretical guarantees for nested stochastic optimization: We establish strong convergence, weak convergence, and $\mathbb{L}^1$ convergence rate results for the NMTS algorithm, proving that the algorithm converges almost surely to the true solution of the intractable likelihood inference problem. The analysis introduces new techniques to ensure uniform convergence of SAA-based gradient estimators along the entire algorithmic trajectory—an essential step absent in prior SA literature—and characterizes the joint asymptotic behavior of both layers.
- Eliminating asymptotic bias and better empirical results: Theoretically, our NMTS algorithm removes the ratio-induced asymptotic bias $O(\sqrt{1/N})$ that persists in single-time-scale methods and achieves a faster $\mathbb{L}^1$ convergence rate of $O\left(\frac{\beta_k}{\alpha_k} + \sqrt{\frac{\alpha_k}{N}}\right)$, with an optimal decay rate of $O(k^{-1/3})$ under proper step-size scheduling. This result significantly sharpens the asymptotic efficiency bounds for simulation-based inference. Consistent with the theory, our experiments show that NMTS delivers lower error than STS under matched computational budgets.

The remainder of the paper is organized as follows. Section 2 reviews the related works. Section 3 provides the necessary background and introduces the NMTS algorithm for both MLE and PDE cases. In Sect. 4, we conduct an in-depth analysis of the algorithm, establishing consistency results and convergence rates. Section 5 extends the NMTS framework to neural network training. Section 6 presents numerical results, and Sect. 7 concludes the paper.

2 Related Works

The existing literature related to our topic can be organized into two main parts: likelihood-free parameter estimation and the MTS algorithm.

Likelihood-free parameter estimation: The LFPE problem is closely related to the simulation-based inference problem in the literature. As a special case, the MLE problem with such an intractable likelihood was first addressed by the gradient-based simulated maximum likelihood estimation (GSMLE) method Peng et al. (2020, 2016, 2014). The Robbins-Monro algorithm, a classic SA technique Harold et al. (1997); Borkar (2009), is applied to optimize unknown parameters for MLE. In the absence of an analytical form for the likelihood function, 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 the distribution sensitivities in Lei et al. (2018) and achieves a square-root convergence rate Glynn et al. (2021). However, the plug-in gradient estimator for the log-likelihood in the SA literature suffers from a ratio bias, leading to inaccuracies in the optimization process Peng et al. (2020); Li and Peng (2025).

For the PDE problem, traditional methods include approximate Bayesian computation and synthetic likelihood methods Tavaré et al. (1997). Techniques such as variational Bayes synthetic likelihood Ong et al. (2018) and multilevel Monte Carlo variational Bayes He et al. (2022) have been applied to likelihood-free models, such as the g-and-k distribution and the α -stable model Peters et al. (2012), but not to stochastic models. Additionally, these methods often require carefully designed summary statistics and distance functions. Meanwhile, numerous approaches leverage neural networks to estimate likelihoods or posteriors that are otherwise intractable to solve Tran et al. (2017); Papamakarios et al. (2019); Greenberg et al. (2019); Glöckler et al. (2022). However, the likelihood functions inferred through neural networks tend to be biased. The incorporation of neural networks and the presence of such bias present theoretical challenges for these algorithms. To simplify the problem and make theoretical analysis feasible, we adopt an SA perspective, using unbiased gradient estimators for the likelihood function. Unbiased stochastic gradient estimation has also been studied in machine learning and downstream tasks (Li et al., 2026; Ren et al., 2025; Chung et al., 2025; Han et al., 2025).

MTS algorithms: In a different line of recent research, MTS algorithms have been applied to a variety of problems, including bilevel optimization Hong et al. (2023), minmax optimization Lin et al. (2025), and reinforcement learning scenarios, such as actor-critic methods Heusel et al. (2017); Wu et al. (2020); Khodadadian et al. (2022). They have also been used extensively in quantile optimization Hu et al. (2022, 2024); Jiang et al. (2023), black-box CoVaR estimation Cao et al. (2025), and dynamic pricing and replenishment problems Zheng et al. (2024).

When considering theoretical results, the convergence and convergence rates of single-timescale (STS) SA have been studied for many years Tsitsiklis (1994); Borkar (2009); Bhandari et al. (2018); Karimi et al. (2019); Liu et al. (2025). In contrast, the convergence and convergence rates of MTS algorithms are not as well understood, primarily due to the complex interplay between the two step sizes and the iterates Mokkadem and Pelletier (2006); Karmakar and Bhatnagar (2018). Specifically, research on MTS convergence has mostly focused on linear settings Konda and Tsitsiklis (2004); Doan (2021); Kaledin et al. (2020). Recently, theoretical results, including high-probability bounds, finite-sample anal-

ysis, central limit theorem (CLT), and convergence rates under various assumptions, have begun to emerge Doan (2022); Han et al. (2024); Hu and Doshi (2024); Zeng et al. (2024). In particular, Doan (2025) leverages a Polyak–Ruppert-type averaging scheme to obtain an optimal $\mathcal{O}(1/k)$ mean-square rate for nonlinear two-time-scale SA under strong monotonicity. In contrast to this prior literature, the NMTS algorithm presented in our paper applies a parallel structure to perform gradient descent. We establish uniform convergence among the parallel gradient descent components and formulate this nested simulation optimization problem in the PDE case, focusing on its asymptotic analysis.

Table 1 summarizes the most relevant literature compared to our paper.

3 Problem Setting and Algorithm Design

This section introduces the basic LFPE problem setting. As a special case, we eliminate ratio bias in the MLE problem by the NMTS algorithm in Sect. 3.1. Then the PDE problem will be solved by our method in Sect. 3.2.

3.1 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 y , a standard result in probability theory allows the density of Y_t to be expressed in closed form as: $p(y; \theta) = f(g^{-1}(y; \theta)) \left| \frac{d}{dy} g^{-1}(y; \theta) \right|$.

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:

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

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

$$\nabla_{\theta} L_T(\theta) = \sum_{t=1}^T \frac{\nabla_{\theta} p(Y_t; \theta)}{p(Y_t; \theta)}. \quad (2)$$

Table 1 Comparison with representative related works

Representative work	Task	Same setting	Ratio bias	Algorithm structure	Convergence analysis	NN ext.
<i>Likelihood-free parameter estimation/simulation-based inference</i>						
GSMLE line Peng et al. (2020, 2016, 2014)	MLE	Yes	No	STS	Yes	No
VB synthetic likelihood Ong et al. (2018)	PDE	Partial	No	VB	No	No
Unbiased MLMC-VB He et al. (2022)	PDE	Partial	No	MLMC-VB	No	No
Neural SBI methods Tran et al. (2017); Papamakarios et al. (2019); Greenberg et al. (2019); Glöckler et al. (2022)	PDE	Broad	No	Neural SBI	No	Yes
<i>Multi-time-scale stochastic approximation</i>						
Representative MTS Doan (2022); Hong et al. (2023); Zeng et al. (2024)	General	No	No	Vanilla MTS	Yes	No
This paper	MLE+PDE	Yes	Yes	Nested MTS	Yes	Yes

Note “Same setting” refers to parameter inference for stochastic simulators with analytically intractable likelihoods, where Monte Carlo likelihood estimation induces a ratio-bias issue. “Partial” means partially overlapping with this setting (e.g., same PDE/LFPE goal but different model class or inference mechanism), while “Broad” means belonging to the broader simulation-based inference literature without focusing on this specific setting. “Ratio bias” indicates whether the Monte Carlo ratio-bias issue is explicitly addressed. “Convergence analysis” indicates whether the paper provides formal convergence/rate analysis for its main inference algorithm. STS = single-time-scale SA; VB = variational Bayes; Neural SBI = neural simulation-based inference

```

1: Input: data  $\{Y_t\}_{t=1}^T$ , initial iterative values  $\theta_0$ ,  $D_0$ , number of samples  $N$ , iterative steps  $K$ , the step-sizes  $\alpha_k$ ,  $\beta_k$ .
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:
       
$$D_{k+1} = D_k + \alpha_k(G_{1,k}(X, Y, \theta_k) - G_{2,k}(X, Y, \theta_k)D_k),$$

       
$$\theta_{k+1} = \Pi_{\Theta}(\theta_k + \beta_k ED_k).$$

5: end for
6: Output:  $\theta_K$ .
```

Algorithm 1 (NMTS for MLE)

```

1: Input: data  $\{Y_t\}_{t=1}^T$ , prior  $p(\theta)$ , iteration initial value  $\lambda_0$  and  $D_0$ , iteration times  $K$ , number of outer layer samples  $M$ , number of inner layer samples  $N$ , step-sizes  $\alpha_k$ ,  $\beta_k$ .
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:
       
$$D_{k+1,m} = D_{k,m} + \alpha_k(G_{1,k}(X, Y, \theta_{k,m}) - G_{2,k}(X, Y, \theta_{k,m})D_{k,m}).$$

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

7: end for
8: Output:  $\lambda_K$ .
```

Algorithm 2 (NMTS for PDE)

Suppose we have unbiased estimators for $\nabla_{\theta} p(Y_t; \theta)$ and $p(Y_t; \theta)$ for every θ and Y_t . Specifically, let N represent batch size, $G_1(X_i, y, \theta)$ and $G_2(X_i, y, \theta)$ represent unbiased estimators obtained via single-run Monte Carlo samples X_i . For simplicity, we define the estimators for $\nabla_{\theta} p(Y_t; \theta)$ and $p(Y_t; \theta)$ with batch size N as

$$G_{1,2}(Y_t, \theta) = \frac{1}{N} \sum_{i=1}^N G_{1,2}(X_i, Y_t, \theta) \quad (3)$$

where $\mathbb{E}_X[G_1(X, Y_t, \theta)] = \nabla_{\theta} p(Y_t; \theta)$, $\mathbb{E}_X[G_2(X, Y_t, \theta)] = p(Y_t; \theta)$. The forms of G_1 and G_2 can be derived by the GLR estimators Peng et al. (2018), and we also present them in Appendix E for completeness. 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). Then, a natural idea is to construct a plug-in estimator and update the parameter by the STS algorithm, also called Robbins-Monro algorithm, as claimed in Peng et al. (2020):

$$\theta_{k+1} = \theta_k + \beta_k \sum_{t=1}^T \frac{G_1(Y_t, \theta_k)}{G_2(Y_t, \theta_k)}. \quad (4)$$

While these individual estimators are unbiased, the ratio of two unbiased estimators may introduce bias. To address this issue, we adopt an NMTS framework that incorporates the gradient estimator into the iterative process, aiming for more accurate optimization results.

We propose the iteration formulae for the NMTS algorithm as follows:

$$D_{k+1} = D_k + \alpha_k (G_{1,k}(X, Y, \theta_k) - G_{2,k}(X, Y, \theta_k) D_k), \quad (5)$$

$$\theta_{k+1} = \Pi_{\Theta}(\theta_k + \beta_k E D_k), \quad (6)$$

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,

$$\begin{aligned} G_{2,k}(X, Y, \theta_k) &= \text{diag}\{G_2(X, Y_1, \theta_k)I_d, \dots, G_2(X, Y_T, \theta_k)I_d\} \\ &= \text{diag}\{G_2(X, Y_1, \theta_k), \dots, G_2(X, Y_T, \theta_k)\} \otimes I_d, \end{aligned}$$

which is a diagonal matrix with $T \times d$ rows and $T \times d$ columns. Here $\otimes$ stands for the Kronecker product and I_d denotes the d -dimensional identity matrix. The constant matrix $E = [I_d, I_d, \dots, I_d] = e^{\top} \otimes I_d$ is a block matrix with d rows and $T \times d$ columns, where e is a column vector of ones. This matrix reshapes the long vector D_k to match the structure of Equation (2), 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 (4). 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 (5) until it converges to the true gradient, and then use this limit in iteration (6). However, such an approach is computationally inefficient. Instead, these coupled iterations are executed interactively, with iteration (5) running at a faster rate than (6), effectively treating θ as fixed in the second iteration. This timescale separation is achieved by ensuring that the step sizes satisfy: $\frac{\beta_k}{\alpha_k} \rightarrow 0$ as k tends to infinity. 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 (3). Ultimately, ED_k converges to zero, and θ converges to its optimal value. The NMTS framework for MLE is summarized in Algorithm 1.

3.2 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 Blei et al. (2017).

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 stochastic optimization Mohamed et al. (2020). 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 terms of the score function method, noting the fact that $\mathbb{E}_{q_\lambda(\theta)}[\nabla_\lambda \log q_\lambda(\theta)] = 0$, we have

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

When the conditional density function $p(y|\theta)$ is given, we can get an unbiased estimator for $\nabla_\lambda L(\lambda)$ naturally by sampling θ from $q_\lambda(\theta)$. However, in this paper, $p(y|\theta)$ is estimated by simulation rather than computed precisely, inducing bias to the $\log p(y|\theta)$ term. Furthermore, the score function 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

$$\begin{aligned}
\nabla_{\lambda} L(\lambda) &= \nabla_{\lambda} \mathbb{E}_{q_{\lambda}(\theta)} [\log p(y|\theta) + \log p(\theta) - \log q_{\lambda}(\theta)] \\
&= \nabla_{\lambda} \mathbb{E}_u [\log p(y|\theta(u; \lambda)) + \log p(\theta(u; \lambda)) - \log q_{\lambda}(\theta(u; \lambda))] \\
&= \mathbb{E}_u [\nabla_{\lambda} \theta(u; \lambda) \cdot (\nabla_{\theta} \log p(y|\theta) + \nabla_{\theta} \log p(\theta) - \nabla_{\theta} \log q_{\lambda}(\theta))].
\end{aligned} \tag{7}$$

In Equation (7), 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 NMTS framework to reduce ratio bias. After calculating the part inside the expectation in Equation (7) 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 (7), 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. This will be analyzed in Sect. 4.1. Unlike Algorithm 1, this approach involves a nested simulation optimization structure, where simulation and optimization are conducted simultaneously.

The NMTS algorithm framework for the PDE problem is shown as follows in Algorithm 2. Here, $G_{1,k}(X, Y, \theta_{k,m})$ and $G_{2,k}(X, Y, \theta_{k,m})$ could be GLR estimators satisfying $\mathbb{E}_X[G_{1,k}(X, Y_t, \theta_{k,m})] = \nabla_{\theta} p(Y_t|\theta_{k,m})$ and $\mathbb{E}_X[G_{2,k}(X, Y_t, \theta_{k,m})] = p(Y_t|\theta_{k,m})$ for every observation t and block m . 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 (7). Due to the nested simulation structure, Algorithm 1 is a special variant of Algorithm 2.

4 Theoretical Results

In this section, we present the convergence results for the proposed NMTS algorithms. The analysis has two nested layers. The *outer layer* concerns the SAA used in the ELBO gradient, while the *inner layer* concerns the SA iterations of the algorithm itself. Because

the notation in this section is necessarily heavier than in the algorithmic sections, we first provide a short roadmap and a compact summary of the main symbols.

The proof strategy proceeds in four steps. First, in Sect. 4.1, we study the outer-layer SAA gradient estimator and show that it converges uniformly to the population ELBO gradient. Second, in Sect. 4.2, we prove that the fast-timescale recursion tracks the likelihood-ratio gradient, and then use this to establish convergence of the slow-timescale update. Third, in Sect. 4.3, we characterize the stochastic fluctuations of the iterates through weak convergence. Finally, in Sect. 4.4, we derive the $\mathbb{L}^1$ convergence rate and compare it with the STS baseline.

Among the notation introduced below, three objects are the most important for following the theory:

- $D_{k,m}$: the fast-timescale tracker for the score corresponding to outer sample u_m ;
- S_k : the aggregated gradient surrogate used in the slow-timescale update;
- $\bar{\lambda}^M$: the optimizer of the outer-layer SAA problem with M fixed outer samples.

The remaining notation is introduced mainly to write the coupled recursions in a compact matrix form.

Suppose that $\theta \in \mathbb{R}^d$ and the feasible domain $\Lambda \subset \mathbb{R}^l$ for the variational parameter $\lambda \in \mathbb{R}^l$ is a convex bounded set defined by inequality constraints. For example, Λ may be a hyper-rectangle or a convex polytope in $\mathbb{R}^l$. We assume that the optimizer $\bar{\lambda}^M$ of the approximate problem lies in the interior of Λ .

Let $(\Omega, \mathcal{F}, P)$ be the probability space generated by the algorithm, and let $\mathcal{F}_k$ denote the natural filtration:

$$\mathcal{F}_k = \sigma\{\{u_m\}_{m=1}^M, \lambda_0, \{D_{0,m}\}_{m=1}^M, \dots, \lambda_k, \{D_{k,m}\}_{m=1}^M\}, \quad k = 0, 1, \dots$$

We use the standard notation $O(\cdot)$, $o(\cdot)$, and $O_p(\cdot)$ throughout.

Recall that $\theta_{k,m} = \theta(u_m; \lambda_k)$ denotes the re-parameterized parameter associated with outer sample u_m at iteration k . For each m , the Monte Carlo estimators of $\nabla_{\theta} p(y \mid \theta_{k,m})$ and $p(y \mid \theta_{k,m})$ are denoted by $G_{1,k,m}$ and $G_{2,k,m}$, respectively. To analyze the M parallel blocks together, we stack them into larger vectors and matrices.

Specifically,

$$G_{1,k} = [G_{1,k,1}^\top, \dots, G_{1,k,M}^\top]^\top \in \mathbb{R}^{MTd},$$

and

$$G_{2,k} = \text{diag}\{G_{2,k,1} \otimes I_d, \dots, G_{2,k,M} \otimes I_d\} \in \mathbb{R}^{MTd \times MTd}.$$

Similarly, we collect all fast-timescale trackers into

$$D_k = [D_{k,1}^\top, \dots, D_{k,M}^\top]^\top \in \mathbb{R}^{MTd}.$$

With this notation, the M parallel fast-timescale recursions can be written compactly as

$$D_{k+1} = D_k + \alpha_k (G_{1,k}(\lambda_k) - G_{2,k}(\lambda_k) D_k). \quad (8)$$

Here the stepsize α_k is the *fast-timescale* stepsize. Intuitively, it controls how quickly the recursion for D_k reacts to new Monte Carlo information and tracks the local likelihood-ratio gradient.

We next introduce three deterministic terms appearing in the slow-timescale update. For each outer sample u_m , define

$$B_m(\lambda) = \nabla_{\theta} \log p(\theta(u_m; \lambda)), \quad C_m(\lambda) = \nabla_{\theta} \log q_{\lambda}(\theta(u_m; \lambda)),$$

and stack them as

$$B(\lambda) = [B_1(\lambda)^{\top}, \dots, B_M(\lambda)^{\top}]^{\top} \in \mathbb{R}^{Md}, \quad C(\lambda) = [C_1(\lambda)^{\top}, \dots, C_M(\lambda)^{\top}]^{\top} \in \mathbb{R}^{Md}.$$

The Jacobian of the re-parameterization map with respect to λ is denoted by

$$A(\lambda) = [A_1(\lambda), \dots, A_M(\lambda)], \quad A_m(\lambda) = \nabla_{\lambda} \theta(u_m; \lambda) \in \mathbb{R}^{d \times l},$$

so $A(\lambda)$ is the matrix that transports score information in the θ -space to the gradient in the λ -space.

To aggregate the T observation-wise score components inside each block, define

$$E^M = \text{diag}\{[I_d, \dots, I_d], \dots, [I_d, \dots, I_d]\} = I_M \otimes E,$$

which is a block-diagonal matrix with Md rows and MTd columns. The role of E^M is purely algebraic: it sums the T score contributions within each outer sample block.

Using these objects, the slow-timescale recursion becomes

$$\lambda_{k+1} = \lambda_k + \beta_k \left(\frac{A(\lambda_k)}{M} (E^M D_k + B(\lambda_k) - C(\lambda_k)) + Z_k \right), \quad (9)$$

where Z_k is the projection correction that keeps λ_k inside Λ . When λ_k lies in the interior of Λ , we have $Z_k = 0$.

Here the stepsize β_k is the *slow-timescale* stepsize. It governs the update of the variational parameter λ_k . The basic two-timescale principle is that α_k is larger than β_k , so the fast recursion for D_k nearly equilibrates relative to the movement of λ_k . This is formalized later by the condition $\beta_k = o(\alpha_k)$.

For convenience, define the aggregated gradient surrogate

$$S_k := \frac{A(\lambda_k)}{M} (E^M D_k + B(\lambda_k) - C(\lambda_k)). \quad (10)$$

The goal of the analysis is to show that S_k tracks the gradient of the approximate ELBO, namely $\nabla_{\lambda} \hat{L}_M(\lambda_k)$.

Table 2 Main notation used in Sect. 4

Symbol	Meaning
$\theta_{k,m}$	Re-parameterized parameter corresponding to outer sample u_m at iteration k
$G_{1,k,m}$	Monte Carlo estimator of $\nabla_{\theta} p(y \theta_{k,m})$
$G_{2,k,m}$	Monte Carlo estimator of $p(y \theta_{k,m})$
$D_{k,m}$	Fast-timescale tracker for $\nabla_{\theta} \log p(y \theta_{k,m})$
D_k	Stacked vector collecting all $\{D_{k,m}\}_{m=1}^M$
$A(\lambda)$	Jacobian matrix of the re-parameterization map $\theta(u; \lambda)$ with respect to λ
$B(\lambda)$	Stacked prior-score term $\nabla_{\theta} \log p(\theta(u_m; \lambda))$
$C(\lambda)$	Stacked variational-score term $\nabla_{\theta} \log q_{\lambda}(\theta(u_m; \lambda))$
E^M	Block matrix aggregating the T observation-wise score components within each outer block
S_k	Aggregated gradient surrogate used in the slow-timescale update
α_k	Fast-timescale stepsize for the recursion of D_k
β_k	Slow-timescale stepsize for the recursion of λ_k
$\hat{L}_M(\lambda)$	Outer-layer SAA approximation of the ELBO
$\tilde{\lambda}^M$	Maximizer of the approximate ELBO $\hat{L}_M(\lambda)$

-
- 1: Input: data $Y: \{Y_t\}_{t=1}^T$, prior $p(\theta)$, iteration rounds K , number of outer layer samples and inner layer samples: M, N .
 - 2: **for** k in $0 : K - 1$ **do**
 - 3: Simulate θ_m from $q_{\lambda_k}(\theta)$ for $m = 1 : M$;
 - 4: Sample $\{X_{m,i}\}$ and calculate the corresponding output $y_{m,i} = g(X_{m,i}; \theta_m)$ for $i = 1 : N$ and $m = 1 : M$;
 - 5: Train $p_{\phi_k}(y|\theta)$ with a faster speed: $\phi_{k+1} = \arg \min_{\phi} -\frac{1}{MN} \sum_{m,i} \log p_{\phi}(y_{m,i}|\theta_m)$.
 - 6: Train $q_{\lambda_k}(\theta)$ with a slower speed: $\lambda_{k+1} = \arg \max_{\lambda} \mathbb{E}_{q_{\lambda}(\theta)} [\log p_{\phi_k}(Y|\theta) + \log p(\theta) - \log q_{\lambda}(\theta)]$.
 - 7: **end for**
 - 8: Output: posterior $q_{\lambda_K}(\theta)$.
-

Algorithm 3 (NMTS for training likelihood and posterior neural networks)

When the number of outer samples M is fixed, we simply write S_k and λ_k ; otherwise, we write S_k^M and λ_k^M to emphasize the dependence on M . To help the reader navigate the notation, Table 2 summarizes the main symbols used in this section.

4.1 Outer Layer Gradient Estimator and Its Asymptotic Analysis

To maximize ELBO, we first use SAA to obtain an unbiased gradient estimator. It is an approximation since the outer layer samples $\{u_m\}_{m=1}^M$ are fixed, which is necessary because the fixed point of each inner iteration depends on u_m . To be specific, the problem approximation can be formulated as below. According to the form of ELBO, the approximation function is defined as

$$\hat{L}_M(\lambda) := \hat{L}(\lambda; u_1, \dots, u_M) = \frac{1}{M} \sum_{m=1}^M \left(\log p(y|\theta(u_m; \lambda)) + \log p(\theta(u_m; \lambda)) - \log q_{\lambda}(\theta(u_m; \lambda)) \right),$$

where $\{u_m\}_{m=1}^M$ are sampled from $p_0(u)$, such that θ follows the distribution $q_\lambda(\theta)$. Using the chain rule, the gradient of $\hat{L}_M(\lambda)$ becomes

$$\begin{aligned}\nabla_\lambda \hat{L}_M(\lambda) &= \frac{1}{M} \sum_{m=1}^M \nabla_\lambda \theta(u_m; \lambda) \left(\sum_{t=1}^T \frac{\nabla_\theta p(Y_t | \theta(u_m; \lambda))}{p(Y_t | \theta(u_m; \lambda))} \right. \\ &\quad \left. + \nabla_\theta \log p(\theta(u_m; \lambda)) - \nabla_\theta \log q_\lambda(\theta(u_m; \lambda)) \right) := \frac{1}{M} \sum_{m=1}^M h(u_m; \lambda).\end{aligned}$$

Thus, given the outer layer samples $\{u_m\}_{m=1}^M$, the algorithm solves the surrogate optimization problem

$$\hat{\lambda}^M = \arg \max_{\lambda \in \Lambda} \hat{L}_M(\lambda).$$

Here, M represents the degree of approximation. We now analyze the relationship between this approximate problem and the true problem, including asymptotic results. The gradient estimator's pointwise convergence follows directly from the law of large numbers. For every λ , almost sure convergence holds as M tends to infinity:

$$\nabla \hat{L}_M(\lambda; u_1, \dots, u_M) \xrightarrow{a.s.} \nabla L(\lambda).$$

The distance between $L(\lambda)$ and $\hat{L}_M(\lambda)$ can be measured using the $\mathbb{L}^2$ norm. For every λ ,

$$\mathbb{E} \|\nabla \hat{L}_M(\lambda; u_1, \dots, u_M) - \nabla L(\lambda)\|^2 = \frac{1}{M} \text{Var}_u(h(u; \lambda)).$$

Furthermore, a CLT applies for every λ as M tends to infinity:

$$\sqrt{M}(\nabla \hat{L}_M(\lambda; u_1, \dots, u_M) - \nabla L(\lambda)) \xrightarrow{d} \mathcal{N}(0, \text{Var}_u(h(u; \lambda))).$$

However, since the iterative process in the NMTS algorithm involves a changing λ_k , we require uniform convergence of the gradient estimator with respect to λ . This ensures convergence across both nested layers as k and M approach infinity, and it is established using empirical process theory.

Let $X_1, \dots, X_n$ be random variables drawn from a probability distribution P on a measurable space. Define $\mathbb{P}_n f = \frac{1}{n} \sum_{i=1}^n f(X_i)$, $Pf = \mathbb{E}f(X)$. By the law of large numbers, the sequence $\mathbb{P}_n f$ converges almost surely to Pf for every f such that Pf is defined. Abstract Glivenko-Cantelli theorems extend this result uniformly to f ranging over a class of functions Vaart (1998). A class $\mathcal{C}$ is called P-Glivenko-Cantelli if $\|\mathbb{P}_n f - Pf\|_{\mathcal{C}} = \sup_{f \in \mathcal{C}} |\mathbb{P}_n f - Pf| \xrightarrow{a.s.} 0$.

The empirical process, evaluated at f , is defined as $\mathbb{G}_n f = \sqrt{n}(\mathbb{P}_n f - Pf)$. By the multivariate CLT, given any finite set of measurable functions f_i with $Pf_i^2 < \infty$, $(\mathbb{G}_n f_1, \dots, \mathbb{G}_n f_k) \xrightarrow{d} (\mathbb{G}_P f_1, \dots, \mathbb{G}_P f_k)$, where the vector on the right follows a multivariate normal distribution with mean zero and covariances $\mathbb{E}\mathbb{G}_P f \mathbb{G}_P g = Pfg - PfPg$. Abstract Donsker theorems extend this result uniformly to classes of functions. A class $\mathcal{C}$ is called P-Donsker if the sequence of processes $\{\mathbb{G}_n f : f \in \mathcal{C}\}$ converges in distribution to a tight

limit process. In our case, this conclusion follows from the assumption stated below, with a proof in Appendix B.

Assumption 1 Suppose the feasible region $\Lambda \subset \mathbb{R}^l$ of λ is compact. Additionally, there exists a measurable function $m(x)$ with $\int_{\Lambda} m(u)^2 p_0(u) du < \infty$ such that for every $\lambda_1, \lambda_2 \in \Lambda$,

$$\|h(u; \lambda_1) - h(u; \lambda_2)\| \leq m(u) \|\lambda_1 - \lambda_2\|.$$

Intuitively, because the slow iterate λ_k evolves across the whole feasible set Λ , we must guarantee that the SAA gradient $\nabla \hat{L}_M(\lambda)$ tracks the population gradient $\nabla L(\lambda)$ simultaneously for all $\lambda \in \Lambda$. A standard route is to verify that the relevant function class is P -Donsker, which yields both a uniform law of large numbers and a functional CLT for the SAA process.

Proposition 2 Under Assumption 1, the gradient estimator $\nabla \hat{L}_M(\lambda)$ converges to the true gradient uniformly with respect to λ :

$$\sup_{\lambda \in \Lambda} |\nabla \hat{L}_M(\lambda) - \nabla L(\lambda)| \xrightarrow{a.s.} 0, \quad M \rightarrow \infty.$$

Furthermore, consider $\sqrt{M}(\nabla \hat{L}_M(\lambda) - \nabla L(\lambda))$ as a stochastic process with respect to λ , it converges to a Gaussian process G_P as M tends to infinity:

$$\sqrt{M}(\nabla \hat{L}_M(\cdot) - \nabla L(\cdot)) \xrightarrow{d} G_P(\cdot),$$

where the Gaussian process G_P has mean zero and covariances

$$\mathbb{E} G_P(\lambda_1) G_P(\lambda_2) = \text{Cov}(\nabla \hat{L}_M(\lambda_1), \nabla \hat{L}_M(\lambda_2)).$$

Proposition 2 guarantees that the SAA gradient $\nabla \hat{L}_M(\lambda)$ uniformly tracks the true gradient $\nabla L(\lambda)$ over the entire parameter set, which is essential for ensuring that the slow-timescale updates remain asymptotically correct along the whole algorithmic trajectory. Moreover, the functional CLT quantifies the outer-layer approximation error via a Gaussian process limit, enabling principled uncertainty assessment and yielding the $O(M^{-1/2})$ scaling that underpins our subsequent convergence rate.

4.2 Strong Convergence Results

In the following convergence proof, the following assumptions are made.

Assumption 3 (1)–(4) are conditions on the stochastic estimators.

- (1) There exists a constant $C_1 > 0$ such that $\sup_{k,u} \mathbb{E}[\|G_{1,k}(X, Y, \theta(u; \lambda_k))\|^2 | \mathcal{F}_k] \leq C_1$ w.p.1.
- (2) There exists a constant $\epsilon > 0$ such that $\inf_{k,u,t} \mathbb{E}[G_{2,k}(X, Y_t, \theta(u; \lambda_k)) | \mathcal{F}_k] \geq \epsilon$ w.p.1.
- (3) There exists a constant $C_2 > 0$ such that $\sup_{k,u} \mathbb{E}[\|G_{2,k}(X, Y, \theta(u; \lambda_k))\|^2 | \mathcal{F}_k] \leq C_2$ w.p.1.

(4) $\mathbb{E}[G_{1,k}(X, Y_t, \theta) | \mathcal{F}_k] = \nabla_{\theta} p(Y_t | \theta)$, $\mathbb{E}[G_{2,k}(X, Y_t, \theta) | \mathcal{F}_k] = p(Y_t | \theta)$ for every θ and t .

(5)–(6) are step-size conditions.

(5) (a) $\alpha_k > 0$, $\sum_{k=0}^{\infty} \alpha_k = \infty$, $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$; (b) $\beta_k > 0$, $\sum_{k=0}^{\infty} \beta_k = \infty$,

$\sum_{k=0}^{\infty} \beta_k^2 < \infty$.

(6) $\beta_k = o(\alpha_k)$. (7)–(8) are regularity conditions.

(7) $p(y|\theta)$ is positive and twice continuously differentiable with respect to θ in $\mathbb{R}^d$. $A(\lambda)$, $B(\lambda)$ and $C(\lambda)$ are continuously differentiable with respect to λ in Λ .

(8) $\hat{L}_M(\lambda)$ and $L(\lambda)$ are twice continuously differentiable with respect to λ in Λ . Furthermore, the equation $\nabla_{\lambda} L(\lambda) = 0$ has a unique solution $\bar{\lambda}$ in Λ , and the Hessian matrix $\nabla_{\lambda}^2 L(\lambda)$ is invertible.

Assumptions 3.1–3.8 are standard types of technical conditions in the SA literature, especially in multi-time-scale and simulation-based settings. Assumptions 3.1 and 3.3 impose uniform second-moment bounds on the Monte Carlo estimators $G_{1,k,m}$ and $G_{2,k,m}$. Such square-integrability conditions are standard in SA analyses because they control the cumulative noise level and are used to establish boundedness of the iterates; see, e.g., Harold et al. (1997); Borkar (2009); Hu et al. (2022, 2024); Cao et al. (2025). Assumption 3.2 requires the likelihood estimator to be uniformly bounded away from zero in conditional expectation. This is the analogue of the usual nondegeneracy condition in quantile- and likelihood-based recursive estimation: here it ensures that the drift term in the fast recursion remains contractive and prevents instability caused by vanishing denominators. Related conditions are also commonly imposed in black-box quantile and CoVaR estimation to guarantee boundedness of the iterates and local identifiability of the target equation; see Hu et al. (2022, 2024); Cao et al. (2025). Assumption 3.4 is the unbiasedness condition for the Monte Carlo estimators of $p(y|\theta)$ and $\nabla_{\theta} p(y|\theta)$. In our setting, this assumption is directly motivated by GLR-type estimators, which provide unbiased likelihood and score estimators under suitable regularity conditions; see Peng et al. (2018); Lei et al. (2018); Glynn et al. (2021). This assumption defines the main theoretical regime of the paper and is also the key condition under which the ratio-free NMTS recursion can be compared rigorously with the biased plug-in STS update. Assumption 3.5 gives the classical Robbins–Monro step-size conditions, which are standard in SA and ensure asymptotic averaging while keeping the accumulated noise finite; see Harold et al. (1997); Borkar (2009). Assumption 3.6 is the core time-scale separation condition of the NMTS framework. Conditions of the form $\beta_k = o(\alpha_k)$ are standard in multi-time-scale SA and ensure that the slow recursion is asymptotically frozen when analyzing the fast recursion; see Mokkadem and Pelletier (2006); Doan (2022); Hong et al. (2023); Zeng et al. (2024); Cao et al. (2025). Finally, Assumptions 3.7–3.8 are standard smoothness and local regularity conditions. Such differentiability and curvature-type assumptions are standard in nonlinear SA and stochastic optimization; see Hong et al. (2023); Han et al. (2024); Mokkadem and Pelletier (2006); Cao et al. (2025).

We first establish that the fast-timescale recursion correctly tracks the log-likelihood score. This is the key ingredient that removes the ratio bias and supports all later convergence results for the slow-timescale update. Since D_k is high-dimensional and can be spliced from $\{D_{k,m}\}_{m=1}^M$, we equivalently examine the uniform convergence of $\{D_{k,m}\}_{m=1}^M$. The proofs in this subsection can be found in Appendix B.

Theorem 4 Assuming that Assumptions 1 and 3.1-3.7 hold, the iterative sequence $\{D_{k,m}\}$ generated by iteration (8) converges to the gradient $\nabla_{\theta} \log p(y|\theta(u;\lambda))|_{(u;\lambda)=(u;\lambda_k)}$, uniformly for every outer layer sample u_m , i.e.,

$$\lim_{k \rightarrow \infty} \sup_m \left\| D_{k,m} - \nabla_{\theta} \log p(y|\theta(u_m; \lambda_k)) \right\| = 0, \quad \text{w.p.1.}$$

where $\nabla_{\theta} \log p(y|\theta(u_m; \lambda_k))$ is also a long vector with $T \times d$ dimensions describing every component of observations, which is defined as $[\nabla_{\theta} \log p(Y_1|\theta(u_m; \lambda_k))^{\top}, \dots, \nabla_{\theta} \log p(Y_T|\theta(u_m; \lambda_k))^{\top}]^{\top}$.

Theorem 4 establishes that the fast-timescale recursion $\{D_{k,m}\}$ uniformly in m tracks the parameter-space score $\nabla_{\theta} \log p(y | \theta(u_m; \lambda_k))$ almost surely. This result is pivotal for the NMTS framework: it validates that the fast layer supplies the slow layer with a ratio-free, asymptotically correct gradient surrogate of the log-likelihood (or ELBO component), thereby eliminating the instability and bias caused by directly dividing two Monte Carlo estimators. Uniformity over all outer samples u_m is crucial for coupling the M parallel fast recursions into a single slow update, and for later arguments that pass from the approximate (SAA) objective to the true objective. In short, Theorem 4 provides the consistency backbone that turns the nested, ratio-free construction into a sound SA for likelihood-free inference.

The proof proceeds in three steps. (i) *Uniform boundedness on sample paths.* Lemmas 2–3 show that the second moments of $D_{k,m}$ are uniformly bounded and, in fact, $\sup_{k,m} \|D_{k,m}\| < \infty$ w.p.1. These bounds rely on: (a) the step-size conditions, (b) the positive lower bound of the estimated density in Assumption 3.2 to control the contraction part $I - \alpha_k G_{2,k,m}$, and (c) square-integrability of the Monte Carlo estimators (Assumptions 3.1 and 3.3). A martingale-square function argument shows that the noise accumulates at the $O(\sum \alpha_k^2)$ scale, hence remains controlled. (ii) *ODE method with two timescales.* We embed the discrete dynamics into piecewise-constant interpolations $\{D_m^n(\cdot), \lambda^n(\cdot)\}$ and decompose the increment into a deterministic drift $H(u_m, D, \lambda)$ plus three perturbations: the Riemann-sum mismatch $\rho_m^n(t)$ and two martingale terms $V_m^n(t), W_m^n(t)$. Lemmas 4–5 show that these perturbations vanish uniformly on every finite horizon. The projection-induced term in the slow recursion is handled via the normal-cone inequality, and Lemma 7 uses the scale separation $\beta_k = o(\alpha_k)$ to freeze λ on the fast-timescale, i.e., $\lambda^n(\cdot) \rightarrow \lambda(0)$ uniformly on compact sets. Passing to the limit yields the decoupled ODE $\dot{D}_m(t) = H(u_m, D_m(t), \lambda^*)$ with $\dot{\lambda}(t) = 0$. (iii) *Global asymptotic stability of the score.* For fixed λ^* , the limiting ODE is linear in D_m with equilibrium $D_m^* = p(y | \theta(u_m; \lambda^*))^{-1} \nabla_{\theta} p(y | \theta(u_m; \lambda^*)) = \nabla_{\theta} \log p(y | \theta(u_m; \lambda^*))$. A Lyapunov function built from the residual $\nabla_{\theta} p - p D_m$ shows global asymptotic stability. By the Arzelà–Ascoli theorem, uniform boundedness and equicontinuity justify taking limits and converging uniformly in m .

The next proposition shows that once the fast-timescale trackers are accurate, the aggregated quantity S_k becomes an asymptotically correct surrogate of the approximate ELBO gradient. This bridges the ratio-free estimator recursion and the optimization, removing ratio bias in the driving gradient.

Proposition 5 Assuming that Assumptions 1 and 3.1-3.7 hold and M is fixed, the sequence $\{S_k\}$ defined by Eq.(10) converges to the gradient of the approximate ELBO:

$$S_k - \nabla_{\lambda} \hat{L}_M(\lambda_k) \xrightarrow{a.s.} 0, \quad k \rightarrow \infty.$$

We then use the previous tracking result to prove convergence of the slow-timescale recursion itself. The following theorem establishes the strong convergence of the slow-timescale recursion, showing that the parameter sequence $\{\lambda_k\}$ driven by the ratio-free gradient estimator indeed converges to the stationary point of the approximate ELBO problem. This result guarantees that the outer layer of the NMTS algorithm correctly tracks the deterministic ODE dynamics associated with $\nabla_{\lambda} \hat{L}_M(\lambda)$ and ultimately stabilizes at $\bar{\lambda}^M$.

Theorem 6 *Assuming that Assumptions 1 and 3.1-3.7 hold, the iterative sequence $\{\lambda_k\}$ generated by iteration (9) converges to a limit point of the following ODE:*

$$\dot{\lambda}(t) = \nabla_{\lambda} \hat{L}_M(\lambda)|_{\lambda=\lambda(t)} + Z(t), \quad w.p.1,$$

where $Z(t)$ is the minimum force applied to prevent $\lambda(t)$ from leaving the feasible domain. The limit point is $\bar{\lambda}^M$.

This theorem confirms the overall stability of the NMTS framework: the slow-timescale iterates λ_k converge to the optimum of the sample-based ELBO, completing the link between inner unbiased gradient estimation and outer parameter optimization. It provides the foundation for subsequent weak convergence and rate analyses, demonstrating that the proposed nested scheme preserves the almost sure convergence property of classical single-timescale SA despite its multi-layer coupling.

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

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

$$\begin{aligned} \lambda_{k+1} = \Pi_{\Lambda} \Big(\lambda_k + \beta_k \frac{1}{M} \sum_{m=1}^M \Big(\nabla_{\lambda} \theta(u_m; \lambda_k) \Big(\sum_{t=1}^T \frac{G_1(X, Y_t, \theta_{k,m})}{G_2(X, Y_t, \theta_{k,m})} \\ + \nabla_{\theta} \log p(\theta_{k,m}) - \nabla_{\theta} \log q_{\lambda}(\theta_{k,m}) \Big) \Big) \Big). \end{aligned} \quad (11)$$

In this single-timescale formulation, the gradient is obtained by directly taking the ratio of two Monte Carlo estimators, rather than introducing an auxiliary variable D_k to track the gradient recursively. Such a ratio can be substantially biased when the sample size N is limited, and the stochastic denominator often leads to numerical instability. Consequently, the estimated gradient direction is imprecise, which deteriorates the optimization accuracy and stability. Both the theoretical analysis in Sect. 4.4 and the empirical evidence in Sect. 6 confirm that the proposed NMTS algorithm consistently outperforms the STS baseline in terms of bias reduction and convergence behavior.

The following proposition connects the inner recursion and the outer approximation. It shows that the recursive estimator S_k^M , which aggregates the outputs of the fast-timescale updates, asymptotically tracks the true gradient of the ELBO as both the iteration number and the number of outer samples grow. This result bridges the SA dynamics of the algorithm with the statistical consistency for SAA.

Proposition 7 *Assuming that Assumptions 1 and 3.1-3.7 hold, the sequence $\{S_k\}$ defined by Eq.(10) converges to the gradient of the true optimization function:*

$$\lim_{M \rightarrow \infty} \lim_{k \rightarrow \infty} \|S_k^M - \nabla_{\lambda} L(\lambda_k)\| = 0, \quad w.p.1.$$

Proposition 7 thus ensures that the fast-timescale recursion delivers an asymptotically unbiased estimate of the true ELBO gradient, which is essential for guaranteeing the correctness of the subsequent slow-timescale parameter updates.

The next proposition establishes the final layer of convergence. It proves that the stationary point of the approximate optimization problem based on M outer samples converges to the true optimum as M increases, thereby closing the loop of the nested convergence analysis.

Proposition 8 *Assuming that Assumptions 1 and 3.1-3.8 hold, then*

$$\lim_{M \rightarrow \infty} \bar{\lambda}^M = \bar{\lambda}, \quad w.p.1.$$

Proposition 8 formally guarantees that the nested simulation optimization algorithm is statistically consistent: the limit of the algorithmic iterates coincides with the true maximum of the expected ELBO as the number of outer samples tends to infinity.

4.3 Weak Convergence

Having established strong convergence in the previous subsection, we now turn to the characterization of the limiting distribution and convergence rate of the NMTS algorithm. While strong convergence guarantees that the iterates $\{\lambda_k\}$ and $\{D_k\}$ asymptotically approach their deterministic limits, it does not quantify how the stochastic noise introduced by finite samples propagates through the coupled recursions. For example, the fixed sample size N used in the estimation of G_1 and G_2 in each iteration controls the variance of the stochastic gradients. To address this, we study the weak convergence and asymptotic normality of the NMTS iterates, which describe their second-order stochastic behavior and allow us to derive convergence rates in distribution.

The weak convergence analysis requires additional regularity on the curvature of the objective and on the step-size sequences. The following assumption, standard in SA literature Bottou et al. (2018); Hu et al. (2024); Cao et al. (2025), ensures that the algorithm operates within a locally stable regime where the limiting distribution exists and is asymptotically normal.

Assumption 9 (1) Let $H_M(\lambda) = \nabla_{\lambda}^2 \hat{L}_M(\lambda)$, and denote its largest eigenvalue by $K_M(\lambda)$. There exists a constant $K_L > 0$, such that $K_M(\lambda) < -K_L$ for every $\lambda \in \Lambda$.

(2) The step-size of the NMTS algorithm take the forms $\alpha_k = \frac{\alpha_0}{k^a}$, $\beta_k = \frac{\beta_0}{k^b}$, where $\frac{1}{2} < a < b \leq 1$ and α_0 and β_0 are positive constants.

We next present the asymptotic normality of the fast and slow iterates. This result formalizes the idea that, after appropriate rescaling, the deviations of λ_k and D_k around their equilib-

rium points converge in distribution to independent Gaussian limits. Their covariance matrices characterize the asymptotic variability of the algorithm induced by stochastic gradient noise at the corresponding time scales. The proof is based on the general weak convergence framework for multi-time-scale SA developed by Mokkadem and Pelletier (2006). Detailed proofs are provided in Appendix C.

Proposition 10 *If Assumptions 1, 3.1-3.8, and 9.1-9.2 hold, then we have*

$$\left(\frac{\sqrt{\beta_k^{-1}}(\lambda_k - \bar{\lambda}^M)}{\sqrt{\alpha_k^{-1}}(D_k - \bar{D})} \right) \xrightarrow{d} \mathcal{N}\left(0, \begin{pmatrix} \Sigma_\lambda & 0 \\ 0 & \Sigma_D \end{pmatrix}\right), \quad k \rightarrow \infty, \quad (12)$$

where M is fixed and $\bar{\lambda}^M$ and $\bar{D}$ are the convergence points of iterations (8) and (9), respectively. The covariance matrices Σ_λ and Σ_D are defined in Equation (C10) in Appendix C.

Proposition 10 provides a probabilistic refinement of the strong convergence result. It shows that, beyond almost sure convergence, the properly normalized iterates follow a joint Gaussian law with block-diagonal covariance, indicating that the slow and fast components are asymptotically independent. This characterization quantifies the stochastic variability of the NMTS algorithm.

By Theorem 4 and Theorem 6, $\bar{D}$ can be expressed as $\nabla_\theta \log p(y|\theta(u; \bar{\lambda}^M))$, which is a long vector with $T \times d \times M$ dimensions defined as the combination of $\{\nabla_\theta \log p(y|u_m; \bar{\lambda}^M)\}_{m=1}^M$. Having established almost-sure tracking on both time scales, we next quantify the fluctuation behavior: how the inner estimator S_k (the gradient surrogate for slow-timescale) deviates from its limit at the correct scaling, and how this depends on the parallelization level M . The following theorem gives a central-limit characterization for S_k under fixed M , which will serve as an input to the outer-layer convergence rate analysis.

Theorem 11 *If Assumptions 1, 3.1-3.8, and 9.1-9.2 hold and M is fixed, we have*

$$\sqrt{\alpha_k^{-1}}(S_k - \nabla_\lambda \hat{L}_M(\bar{\lambda}^M)) = \sqrt{\alpha_k^{-1}}S_k \xrightarrow{d} \mathcal{N}(0, \Sigma_s^M), \quad k \rightarrow \infty,$$

where $\Sigma_s^M = \frac{1}{M^2} A(\bar{\lambda}^M) E^M \Sigma_D (E^M)^\top A(\bar{\lambda}^M)^\top$.

Theorem 11 formalizes that the inner gradient surrogate attains a $\sqrt{\alpha_k}$ -scale Gaussian fluctuation limit. This CLT is the key ingredient to propagate fast-timescale noise into the gradient surrogate update and derive weak convergence rates for the composite estimator in what follows.

Next, we analyze how the inner Monte Carlo batch size N and the outer parallelization level M enter the fluctuation sizes. The following lemma isolates the order in N and M of the asymptotic covariance matrices in Proposition 10.

Lemma 1 *Under the conditions in Proposition 10, Σ_D is a covariance matrix with $T \times d \times M$ dimensions and its elements have an order of $O(N^{-1})$. While Σ_λ is a covariance matrix with l dimensions, and its element also has an order of $O(N^{-1})$.*

Lemma 1 indicates that the fundamental noise level is governed by the inner Monte Carlo batch size N . In particular, as $N \rightarrow \infty$ the inner-layer fluctuations vanish and the algorithm becomes deterministic. Similarly, an infinite number of outer-layer samples M allows the ELBO function to be estimated exactly. In this scenario, the algorithm operates with infinitely many parallel faster scale iterations and one slower scale iteration, resulting in the asymptotic variance of constant order with respect to M . This is intuitive, as the number of outer-layer samples does not affect the asymptotic variance of the inner iterations.

Building on the CLT for S_k and the covariance orders above, we now provide an explicit weak convergence rate for the estimation error of the slow-timescale gradient itself, as a function of the iteration index k , inner sample size N , and the number of outer samples M . This bound separates the contribution of inner Monte Carlo variability and the outer SAA error.

Theorem 12 *If Assumptions 1, 3.1-3.8, and 9.1-9.2 hold, then we have*

$$S_k^M - \nabla_{\lambda} L(\lambda_k) = O_p\left(\frac{\alpha_k^{\frac{1}{2}}}{N^{\frac{1}{2}}}\right) + O_p(M^{-\frac{1}{2}}).$$

Theorem 12 shows that the gradient-surrogate error decomposes additively into an inner-layer fluctuation term of order $\sqrt{\alpha_k/N}$ and an outer SAA term of order $M^{-1/2}$. This clean separation will allow us to combine inner and outer central limit behaviors to obtain the weak rate for the decision iterate λ_k^M .

We have shown that the iterative sequence λ_k^M weakly converges to $\bar{\lambda}^M$. It is then natural to quantify the outer-layer statistical error due solely to the SAA, i.e., the gap between the M -sample optimizer $\bar{\lambda}^M$ and the true optimizer $\bar{\lambda}$.

Theorem 13 *If Assumptions 1, 3.1-3.8, and 9.1-9.2 hold, then as $M \rightarrow \infty$, we have*

$$\sqrt{M}(\bar{\lambda}^M - \bar{\lambda}) \xrightarrow{d} \mathcal{N}(0, \nabla^2 L(\bar{\lambda})^{-1} \text{Var}_u(h(u; \bar{\lambda})) \nabla^2 L(\bar{\lambda})^{-\top}).$$

Theorem 13 is a classical SAA CLT in our setting: it asserts that the outer layer optimizer concentrates at rate $M^{-1/2}$ around the true optimizer with a covariance determined by curvature and the variability of $h(u; \lambda)$ in u . This result will be coupled with the inner-layer fluctuation to yield the composite weak rate for λ_k^M .

Finally, combining the inner fluctuation of the slow-timescale iterate around $\bar{\lambda}^M$ and the outer SAA fluctuation of $\bar{\lambda}^M$ around $\bar{\lambda}$, we obtain the overall weak convergence rate for the NMTS decision iterate.

Theorem 14 *If Assumptions 1, 3.1-3.8, and 9.1-9.2 hold, then we have*

$$\lambda_k^M - \bar{\lambda} = O_p\left(\frac{\beta_k^{\frac{1}{2}}}{N^{\frac{1}{2}}}\right) + O_p(M^{-\frac{1}{2}}).$$

Theorem 14 shows that the total weak error decomposes into two orthogonal sources: the inner Monte Carlo noise propagated through the slow-timescale dynamics at order $\sqrt{\beta_k/N}$, and the outer SAA error at order $M^{-1/2}$.

4.4 $\mathbb{L}^1$ Convergence Rate

Beyond asymptotic normality, we quantify the mean absolute error (MAE) of NMITS iterates. As a special case, the MLE setting corresponds to $M = 1$. So we first fix M and expose how the timescale choice and inner Monte Carlo variance co-determine the $\mathbb{L}^1$ rate. This separates a deterministic tracking term due to timescale mismatch from a stochastic fluctuation term driven by batch size N . While unbiasedness guarantees convergence of the algorithm, the convergence rate depends on the variance. It complements strong convergence by providing a characterization at the level of convergence rates, and it complements weak convergence: the former gives the order of the mean error, while the latter gives the limiting distribution and asymptotic variance of the random fluctuations.

The next theorem bounds the mean tracking error of the fast recursion in Equation (8). It shows that the mismatch between the fast and slow iterates contributes an $O(\beta_k/\alpha_k)$ term, while inner Monte Carlo noise contributes an $O(\sqrt{\alpha_k/N})$ term.

Theorem 15 *If M is fixed, Assumptions 1, 3.1–3.8 and 9.1–9.2 hold, the sequence D_k generated by recursion (8) satisfies*

$$\mathbb{E}[|D_k - \nabla_{\theta} \log p(y|\theta(u; \lambda_k))|] = O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right). \quad (13)$$

This result isolates the bias–variance tradeoff on the fast scale: shrinking β_k/α_k improves tracking, while increasing N reduces stochastic variation. It provides the key input for the slow recursion rate.

We now transfer the previous bound to the parameter update for Equation (9). The following theorem shows that the slow sequence achieves the same $\mathbb{L}^1$ rate, hence removing the ratio-induced $O(N^{-1/2})$ bias that persists under single-timescale schemes.

Theorem 16 (Faster convergence) *If M is fixed, Assumptions 1, 3.1–3.8 and 9.1–9.2 hold, the sequence λ_k generated by recursion (9) satisfies*

$$\mathbb{E}[|\lambda_k - \bar{\lambda}^M|] = O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right). \quad (14)$$

Under polynomial stepsizes $\alpha_k = k^{-a}$, $\beta_k = k^{-b}$ with $\frac{1}{2} < a < b \leq 1$, the right-hand side is minimized when $b - a = \frac{a}{2}$, i.e., $a = \frac{2}{3}$ and $b = 1$. With this choice, both terms scale as $k^{-1/3}$, so the $\mathbb{L}^1$ error decays as $k^{-1/3}$; equivalently, the mean square error (MSE) decays as $k^{-2/3}$, which matches the optimal rate in the two-timescale SA literature Doan (2022); Hong et al. (2023). Intuitively, the fast tracker averages noise quickly enough via α_k while the slow update moves cautiously enough via β_k so their errors shrink.

For comparison, we record the $\mathbb{L}^1$ rate of the ratio-based single-timescale (STS) update, which exhibits a nonvanishing $O(N^{-1/2})$ term that cannot be annealed over iterations. The asymptotic bias of stochastic gradient descent can also be referenced to Doucet and Tadic (2017).

Proposition 17 *If M is fixed, Assumptions 1, 3.1-3.8 and 9.1-9.2 hold, the sequence λ_k generated by recursion (11) satisfies*

$$\mathbb{E}[\|\lambda_k - \bar{\lambda}^M\|] = O(\beta_k) + O\left(\sqrt{\frac{1}{N}}\right), \quad (15)$$

Comparing Theorem 16 with Proposition 17 highlights the advantage of NMTS: the fixed $O(N^{-1/2})$ asymptotic bias in STS is replaced by a vanishing $O(\sqrt{\alpha_k/N})$ term due to the decreasing stepsize. The stepsize dependence changes from $O(\beta_k)$ to $O(\beta_k/\alpha_k)$, which also converge to 0 due to the stepsize condition. This means NMTS gains accuracy with iterations rather than larger inner batchsize, delivering strictly sharper rates and better long-run accuracy.

We next incorporate the approximation error from the outer SAA layer. The following theorem combines inner tracking, inner Monte Carlo noise, and outer sampling error to bound the distance to the true optimizer $\bar{\lambda}$.

Theorem 18 *In the NMTS algorithm, if Assumptions 1, 3.1-3.8 and 9.1-9.2 hold, the sequence λ_k^M generated by recursion (9) satisfies*

$$\mathbb{E}[\|\lambda_k^M - \bar{\lambda}\|] = O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right) + O\left(\sqrt{\frac{1}{M}}\right). \quad (16)$$

This decomposition cleanly attributes error to three sources: (i) timescale mismatch, (ii) inner Monte Carlo variance, and (iii) outer SAA variance. Increasing M reduces the outer error at the nominal $M^{-1/2}$ SAA rate, while the NMTS structure attenuates inner noise through the $\sqrt{\alpha_k}$ factor.

For completeness, we state the counterpart bound for STS with outer SAA, which retains the nonvanishing $O(N^{-1/2})$ term even after averaging over M outer samples.

Proposition 19 *In the STS algorithm, if Assumptions 1, 3.1-3.8 9.1-9.2 hold, the sequence λ_k^M generated by recursion (11) satisfies*

$$\mathbb{E}[\|\lambda_k^M - \bar{\lambda}\|] = O(\beta_k) + O\left(\sqrt{\frac{1}{N}}\right) + O\left(\sqrt{\frac{1}{M}}\right). \quad (17)$$

Taken together, the above theorems and propositions establish that NMTS removes the ratio-induced asymptotic bias and yields strictly sharper $\mathbb{L}^1$ rates. Under the balanced choice $\alpha_k = k^{-2/3}$ and $\beta_k = k^{-1}$, the MAE decays as $k^{-1/3}$ and the MSE attains the optimal $k^{-2/3}$ order, offering a clear recipe for stepsize in practice. The proofs in this subsection can be found in Appendix D. Furthermore, the convergence of the variational parameter λ_k^M

induces the uniform convergence of the approximate posterior $q_{\lambda_k^M}(\theta)$. These results are detailed in Appendix A.

5 Extension: Training Two Neural Networks at Different Time Scales

In previous sections, we proposed the NMTS algorithm framework and established its asymptotic properties. The main idea involves using two coupled iterations to update parameters and eliminate ratio bias. Estimation and optimization are performed simultaneously through these coupled iterations: a faster iteration approximates the gradient of the log-likelihood function, while a slower iteration updates the variational parameter λ in $q_\lambda(\theta)$. Additionally, the likelihood function and its gradient are estimated using unbiased estimators. However, when the simulator is sufficiently complex and unbiased estimators are challenging to obtain, more powerful tools are needed to approximate the likelihood function. Similarly, a more expressive variational distribution family $\{q_\lambda(\theta)\}$ may be required to better represent the true posterior when it is complex.

To address the first challenge, a natural approach is to use a neural network to approximate the intractable likelihood function as an alternative to the GLR method Papamakarios et al. (2019). The GLR method is advantageous due to its unbiasedness and simplicity, but relies on relatively strict regularity conditions Peng et al. (2020). A neural network offers a flexible alternative when these conditions are not satisfied, though it provides a biased estimate of the likelihood function. Hence, we train a deep neural density estimator $p_\phi(y|\theta)$ by minimizing the forward KL divergence between $p_\phi(y|\theta)$ and the true conditional density $p(y|\theta)$, which is defined as

$$KL(p(y|\theta)||p_\phi(y|\theta)) = \mathbb{E}_{\theta \sim q_\lambda(\theta), y \sim p(y|\theta)} \left[\log \left(\frac{p(y|\theta)}{p_\phi(y|\theta)} \right) \right].$$

This optimization minimizes the divergence between the unknown conditional density $p(y|\theta)$ and the network $p_\phi(y|\theta)$ using samples (θ, y) generated from the simulator. The loss function for the neural network at each iteration is

$$L_{faster}(\phi) = -\frac{1}{MN} \sum_{m=1}^M \sum_{i=1}^N \log p_\phi(y_{m,i}|\theta_m),$$

where $\theta_m \sim q_\lambda(\theta)$, $y_{m,i} \sim p(y|\theta_m)$, and $p_\phi(y|\theta)$ acts as a conditional density estimator. This network learns the true conditional density $p(y|\theta)$ by generating many samples from the simulator. While this process serves the same purpose as the GLR method—approximating the intractable likelihood—the estimation method differs. Here, L_{faster} denotes that the neural network operates at a faster time scale, with a larger step-size. As in previous cases, while fixing λ and iterating until convergence would provide accurate estimates, such an approach is computationally expensive. Thus, the coupled iterations are performed simultaneously, with the faster iteration preceding the slower one.

To address the second challenge, another neural network $q_\lambda(\theta)$ can be employed to construct a more expressive posterior distribution. The loss function is the ELBO, as in Algorithm 2:

$$L_{\text{slower}}(\lambda) = \mathbb{E}_{q_\lambda(\theta)}[\log p_\phi(Y|\theta) + \log p(\theta) - \log q_\lambda(\theta)].$$

Here Y is the observed data and $p_\phi(Y|\theta)$ is the likelihood network trained at the faster time scale. Unlike Algorithm 2, the convergence of ϕ is independent of the realization of θ , so fixing the outer-layer samples is unnecessary.

The choice of the variational distribution family $q_\lambda(\theta)$ is an important step. Our NMTS framework places no restrictions on the choice of the variational distribution family, which also implies its scalability and compatibility. Beyond simple choices such as the normal distribution, more sophisticated methods for selecting posterior distributions with good representational power have been studied. These include normalizing flows, such as planar flows, Masked Autoregressive Flow (MAF), Inverse Autoregressive Flow (IAF), and others Rezende and Mohamed (2015); Papamakarios et al. (2017); Dinh et al. (2016); Kingma et al. (2016). Normalizing flows are a powerful technique used to model complex probability distributions by mapping them from simpler, more tractable ones. This is achieved through a learned transformation, which acts as a bijective function. These flows are highly advantageous due to their flexibility in approximating a wide array of distribution shapes. Additionally, the re-parameterization trick is employed to ensure low-variance stochastic gradient estimation.

Thus, there are two networks here. The faster scale network $p_\phi(y|\theta)$ is used to update the parameters ϕ to track the intractable likelihood function $p(y|\theta)$, while the slower scale network $q_\lambda(\theta)$ is used to approximate the posterior by updating the variational parameter λ . Optimization and estimation are alternately updated by two coupled neural networks, respectively. These are two coupled iterations with each updated at two different scales, which are contained in our NMTS framework. The specific algorithm is given as follows.

Because the likelihood estimator is now biased and the joint neural-network training dynamics are substantially more complex, our goal here is to present a conditional extension of the NMTS idea, together with a theoretical statement and supporting empirical evidence in Sect. 6.3. To keep the analysis tractable, we do not attempt here to establish a full end-to-end convergence theory for the joint training dynamics of the two neural networks. Instead, we isolate the role of approximation bias in the learned likelihood model. Specifically, we assume that the bias of the neural likelihood estimator and its gradient decays along the fast timescale, and then study how this shrinking approximation error affects the slow-timescale posterior update.

Assumption

20 $\mathbb{E}[p_{\phi_k}(Y_t|\theta)|\mathcal{F}_k] - p(Y_t|\theta) = O(\gamma_k^{(1)}) \rightarrow 0$, $\mathbb{E}[\nabla_\theta p_{\phi_k}(Y_t|\theta)|\mathcal{F}_k] - \nabla_\theta p(Y_t|\theta) = O(\gamma_k^{(2)}) \rightarrow 0$ for every θ and $t = 1, 2, \dots, T$.

This assumption implies that at the first time scale, the bias in the neural network $p_{\phi_k}(Y_t|\theta)$ and its gradient diminishes at rates $O(\gamma_k^{(1)})$ and $O(\gamma_k^{(2)})$, respectively. These rates depend on the training settings and the network's properties, which may not be directly accessible. Under this assumption, the following proposition demonstrates that the shrinking bias at the faster time scale induces a corresponding bias reduction at the slower time scale. It shows that if the approximation bias of the learned likelihood model decreases along the fast tim-

escale, then the resulting error in the slow-timescale posterior update is correspondingly controlled.

Proposition 21 *If M is fixed, Assumptions 3.1-3.3, 3.5-3.8, 9.1-9.2, and 20 hold, the sequence λ_k satisfies*

$$\mathbb{E}[\|\lambda_k - \bar{\lambda}^M\|] = O(\beta_k) + O(\gamma_k^{(1)}) + O(\gamma_k^{(2)}).$$

6 Numerical Experiments

In this section, we demonstrate the application of the NMTS algorithm framework, comprising three specific algorithms, to various cases. Algorithms 1, 2, and 3 are implemented sequentially. Section 6.1 addresses the MLE case, while Sect. 6.2 focuses on the PDE case. In Sect. 6.3, we showcase the application of our framework through an example of a food production system.

6.1 MLE Case

We evaluate the proposed NMTS framework in the MLE setting on a latent-variable model. 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 a 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: $\hat{\theta} = \sqrt{\frac{1}{T} \sum_{t=1}^T Y_t^2 - 1}$, which serves as a ground-truth target for accuracy assessment.

The true value θ is set to be 1. The faster and slower step-size is chosen as $\frac{20}{(k \log(k+1))^{2/3}}$ and $\frac{0.1}{k \log(k+1)}$, respectively, which satisfy the step-size condition of the NMTS 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 NMTS 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, we do not need such heavy computational costs while reaching a more accurate estimation. All comparisons between NMTS and STS are conducted under matched experimental budgets. In particular, within each experiment, the two methods use the same observed data, the same initialization, the same batch size N , the same number of iterations, and the same number of independent repetitions.

Under the same simulation budget, NMTS improves estimation accuracy relative to STS, Fig. 1(a) exhibits the convergence results of NMTS and STS with $N = 10^4$ simulated samples based on 100 independent experiments. Compared to the true MLE, NMTS 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 NMTS and STS is 0.7s and 0.72s, respectively, indicating the gains come from the update rule rather than extra computation. Figure 1(b) depicts the convergence result with

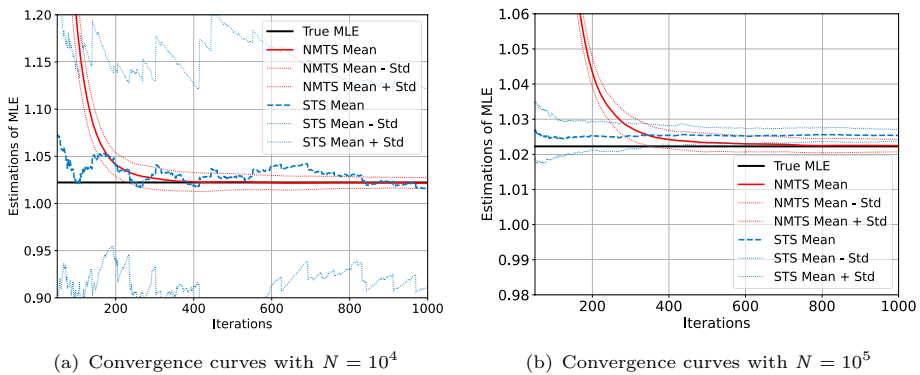


Fig. 1 Trajectories of NMTS and STS with different sample sizes based on 100 independent experiments

10^5 simulated samples based on 100 independent experiments. Even with a large number of simulated samples, NMTS outperforms STS since it suffers from the asymptotic bias.

Table 3 records the MAE for the two methods based on 100 independent experiments. Across all batch sizes, NMTS demonstrates significantly higher estimation accuracy than STS. These trends align with our theory: by removing the noisy denominator, NMTS suppresses the ratio bias and reduces variance under the same budget.

Figure 2 depicts the log-log plot of the MAE of the estimators versus the iteration number k . For each of the 100 settings, we independently sample observations and run NMTS and STS once. The $\log(\text{accuracy})$ is defined as $\log \mathbb{E}[|\theta_k - \hat{\theta}|]$. The observed convergence rates align with Theorem 16 for NMTS. On the contrary, STS suffers from an asymptotic bias caused by the ratio gradient estimator determined by N , which aligns with Proposition 17. NMTS achieves higher accuracy sooner and continues to improve with k , whereas STS saturates due to ratio bias. These results confirm the superior performance of NMTS over STS. Additional non-Gaussian MLE experiments with exponential and uniform latent noise are reported in Appendix A.2, and the results show the same qualitative advantage of NMTS over STS.

6.2 PDE Case

We apply Algorithm 2 to test the NMTS 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 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; the 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 the prior over and over again. Suppose there are only 10 independent observations for one batch. Set feasible region

Table 3 The MAE of the NMTS and STS methods, based on 100 independent experiments after 10000 iterations

Batch size	Absolute Bias $\pm$ std	
	NMTS	STS
1	$2.24 \times 10^{-1} \pm 2.7 \times 10^{-1}$	$3.72 \times 10^{-1} \pm 5.55 \times 10^{-1}$
10	$5.94 \times 10^{-2} \pm 7.3 \times 10^{-2}$	$3.96 \times 10^{-1} \pm 4.4 \times 10^{-1}$
10^2	$1.78 \times 10^{-2} \pm 2.2 \times 10^{-2}$	$3.59 \times 10^{-1} \pm 3.9 \times 10^{-1}$
10^3	$6.69 \times 10^{-3} \pm 8 \times 10^{-3}$	$1.36 \times 10^{-1} \pm 2 \times 10^{-1}$
10^4	$1.78 \times 10^{-3} \pm 2.2 \times 10^{-3}$	$6.56 \times 10^{-2} \pm 1.2 \times 10^{-1}$
10^5	$3.95 \times 10^{-4} \pm 7.2 \times 10^{-4}$	$2.4 \times 10^{-3} \pm 2.7 \times 10^{-3}$

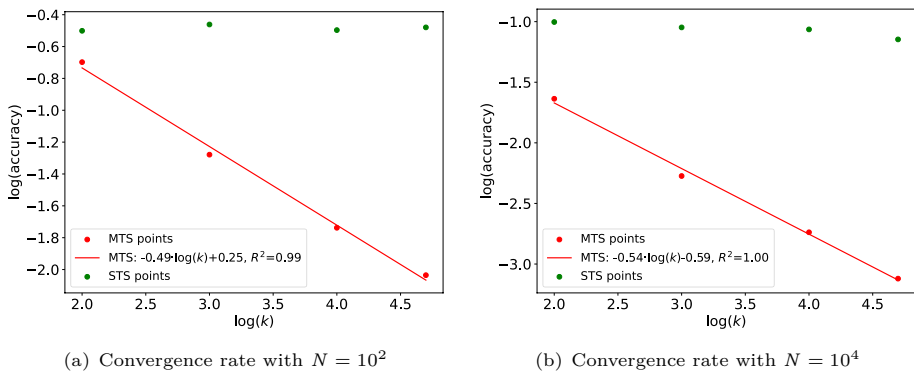


Fig. 2 Log-log plot of the MAE of the estimators versus the iteration step k of NMTS and STS algorithm based on 100 independent experiments

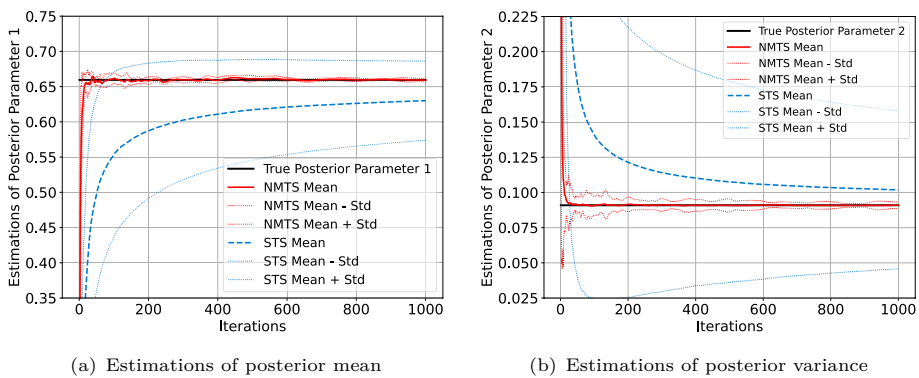


Fig. 3 Trajectories of NMTS and STS with sample size 10^4 based on 100 independent experiments

$\Lambda = [-1, 10] \times [0.01, 2]$ and initial value $\lambda_0 = (0, 1)$. First, we set $M = 10$ outer layer samples u_m and compare the NMTS algorithm with the analytical posterior and STS method. The faster and slower step-size is chosen as $\frac{10}{(k \log(k+1))^{2/3}}$ and $\frac{1}{k \log(k+1)}$, respectively. As in the MLE case, the NMTS and STS methods are compared under the same experimental budget: they use the same observations, the same initialization, the same batch size N , the same number of outer-layer samples M , the same number of iterations, and the same number of independent repetitions. Figure 3 displays the trajectories of NMTS and STS with sample size 10^4 based on 100 independent experiments. Specifically, Fig. 3(a) exhibits the convergence for the posterior mean μ and Fig. 3(b) exhibits the convergence for the posterior variance σ^2 . NMTS achieves lower bias and standard error than STS when compared to the true posterior parameters.

Table 4 records the absolute error for both estimators based on 100 independent experiments. Across all batch sizes, NMTS consistently outperforms STS in estimation accuracy. The accuracy is improved by one to two orders of magnitude.

Figure 4 presents log-log MAE curves versus k in 100 independent experiments when batch size $N = 10^3$. The observed slopes for NMTS track the k -dependence predicted by

Table 4 The MAE of the NMTS and STS methods, based on 100 independent experiments after 50000 iterations

Batch size	Posterior Mean		Posterior Variance	
	NMTS	STS	NMTS	STS
10	1.19×10^{-1}	9.89×10^{-1}	7.95×10^{-2}	4.15×10^{-1}
10^2	2.57×10^{-3}	1.69×10^{-1}	4.18×10^{-4}	1.47×10^{-1}
10^3	8.38×10^{-4}	7.5×10^{-3}	1.40×10^{-4}	1.13×10^{-3}
10^4	1.19×10^{-4}	8.56×10^{-4}	5.83×10^{-5}	7.49×10^{-4}
10^5	6.30×10^{-5}	3.71×10^{-4}	2.78×10^{-5}	1.18×10^{-4}

Theorem 18, while STS displays an apparent floor consistent with ratio-induced bias by Proposition 19. Together with the MLE case, the PDE experiments confirm that the ratio-free design of NMTS translates into tangible accuracy and stability gains in practice. A further non-Gaussian PDE experiment with logistic latent noise is provided in Appendix E.3, where NMTS again achieves lower posterior-parameter error than STS under matched budgets.

6.3 MTS for Training Likelihood and Posterior Neural Networks

This example is intended to mimic a realistic simulator-calibration problem in operations, where only system-level output trajectories are observed and the goal is to infer latent service-time parameters of the underlying stochastic process. In cases where the true posterior is unknown, direct comparisons between algorithms become challenging. Thus, Sect. 6.3.1 illustrates the advantages of the NMTS framework using a toy example, while Sect. 6.3.2 describes its application to a complex simulator where analytical likelihood is infeasible.

6.3.1 A Toy Example

We use the same problem setting as in 6.2 and apply Algorithm 3. MAF method and IAF method Papamakarios et al. (2017); Kingma et al. (2016) are applied to build a conditional likelihood estimator $p_\phi(y|\theta)$ and variational distribution family $q_\lambda(\theta)$, respectively based on their specific nature. Details of the MAF and IAF setups are provided in Appendix E.4. To make a fair comparison, NMTS and STS use the same likelihood-network architecture, the same posterior-network architecture, the same generated training samples, and the same number of training rounds; the difference lies in the update schedule between the two networks.

The results demonstrate the superior accuracy of the NMTS algorithm compared to the corresponding STS algorithm. Figure 5 shows that the posterior estimated by NMTS closely matches the true posterior, whereas the posterior estimated by STS exhibits noticeable deviation. Notably, NMTS achieves this improvement without additional computational burden, as the primary adjustment lies in the training speeds of the two neural networks.

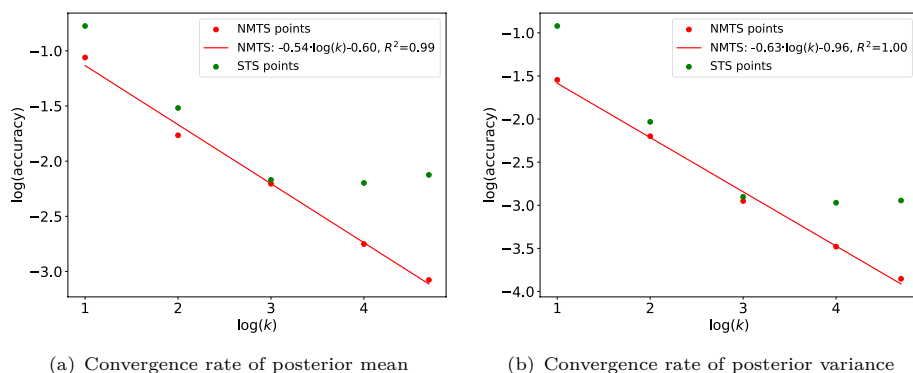
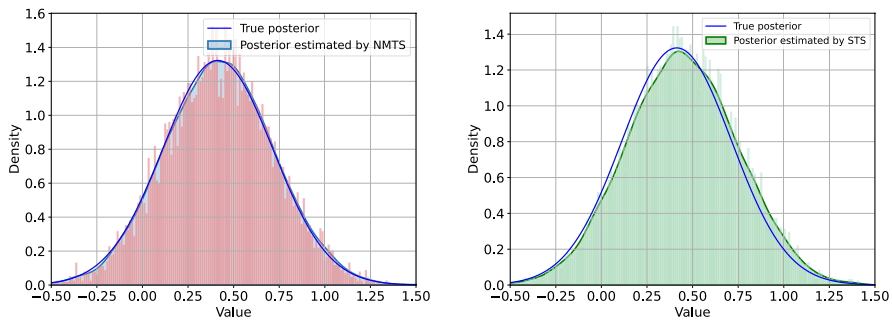


Fig. 4 Log-log plot of the MAE of the estimators versus the iteration step k of NMTS and STS algorithm based on 100 independent experiments when $N = 10^3$



(a) Posterior estimated by NMTS through neural networks (b) Posterior estimated by STS through neural networks

Fig. 5 Posterior estimated by NMTS and STS through neural networks

6.3.2 Parameter Estimation in Food Preparation Process

In this section, we build a stochastic model as a simulator $Y(X; \theta)$, which portrays the food production process in a restaurant. Here Y is the output, X characterizes the stochasticity of the model, and θ comprises the parameters whose posterior distribution we aim to estimate. In this case, the analytical likelihood $p(Y|\theta)$ is absent and the joint posterior of parameters could be complex. We need a general variational parameter class, a neural network, to represent the posterior better, rather than a normal distribution with only two variational parameters in Sect. 6.2.

First, we introduce the setting of the simulator. Assume that order arrival follows a Poisson distribution with parameter 2. The food preparation process comprises three stages. At first, one clerk is checking and processing the order, and the processing time follows a Gamma distribution with shape parameter 3 and inverse scale parameter 2. Next, three cooks are preparing the food, where the preparation time is the first parameter θ_1 whose posterior we want to estimate. After the food is prepared, one clerk is responsible for packing the food, and the packing time is the second parameter θ_2 we want to estimate. Each procedure can be modeled as a single server or three servers queue with a buffer of unlimited capacity, where each job is served based on the first-in/first-out discipline. The final observation is the time series of the completion time of the food orders. This process is illustrated in Fig. 12 in Appendix E.4. To obtain the observations, we sample $\theta = (\theta_1, \theta_2)$ ten times from independent Gamma distribution $(\Gamma(4, 2), \Gamma(1, 1))$. Then, by realizing the stochastic part X and plugging them into the model, we can obtain a realization of the 10-dimensional output $\hat{Y}(X; \theta)$ as our observation. The posterior is estimated based on this observation.

The prior of θ is set to be a uniform distribution: $\theta \sim \mathcal{U}(0, 15)$. MAF and IAF methods are also applied to build $p_\phi(y|\theta)$ and $q_\lambda(\theta)$ in setting the same as Sect. 6.3.1. The details for training can be found in Appendix E.4. Figure 6(a) demonstrates the posterior estimated by NMTS, with the light blue region on the edge representing the marginal distribution. Due to the complexity of the joint density, employing a neural network as a general variational class is necessary. For the output performance measure, we generate another replicated output using parameters sampled from the posterior. Figure 6(b) illustrates that the resulting sequence closely matches the original observations, despite the prior being far from the posterior. This consistency suggests that the learned posterior concentrates on

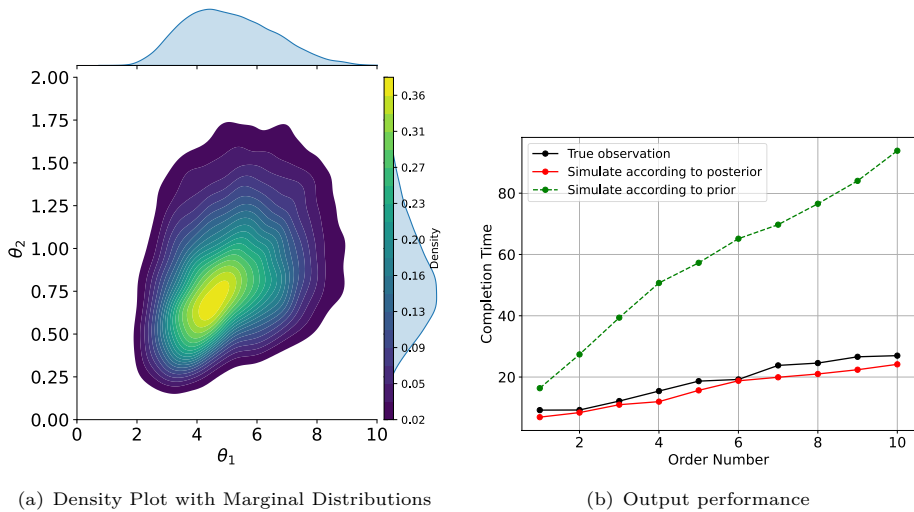


Fig. 6 Posterior estimated by NMTS through neural networks

parameter regions that accurately capture the system's dynamics, thereby demonstrating that we have more understanding of the black box stochastic model. These results provide empirical evidence that the proposed framework is practically applicable in a realistic simulator-calibration setting and can be combined with flexible neural approximators when both the likelihood and the posterior are analytically inaccessible.

7 Conclusion

Our framework is particularly relevant for simulation-based calibration and posterior inference in stochastic operational systems where gradient-based updates are desired but direct likelihood-ratio estimation is unstable under limited simulation budgets. We introduce a ratio-free NMTS scheme that tracks the score on a fast-timescale and updates parameters on a slow-timescale, thereby removing the instability and bias caused by dividing two noisy Monte Carlo estimators. Different from a direct application of vanilla MTS algorithm, we construct a nested SAA structure with M outer scenarios and parallel fast trackers. On the theory side, we establish almost-sure convergence for dual layers, a central-limit characterization for the coupled iterates, and sharp $\mathbb{L}^1$ bounds that decompose error into a timescale mismatch term $O(\beta_k/\alpha_k)$ and an inner Monte Carlo term $O(\sqrt{\alpha_k/N})$, with an additional $O(M^{-1/2})$ outer SAA term when applicable. Furthermore, we have introduced neural network training to our model, showcasing the versatility and scalability of our framework. Future work encompasses eliminating ratio bias in more scenarios, and our framework can be more widely applied and extended.

The Uniform Convergence of Approximate Posterior

Now, we focus on the convergence of the approximate posterior $q_{\lambda}(\theta)$. Thanks to the fact that λ_k converges in different senses as we proved in the sections before, we will prove the functional convergence of $q_{\lambda_k}(\theta)$ in this part.

Proposition 22 *If θ satisfies $\frac{\partial q_{\lambda}(\theta)}{\partial \lambda}|_{\lambda=\bar{\lambda}^M} \neq 0$, and Assumptions 1, 3.1-3.8, and 9.1-9.2 hold, we have*

$$\sqrt{\beta_k^{-1}}(q_{\lambda_k}(\theta) - q_{\bar{\lambda}^M}(\theta)) \xrightarrow{d} \mathcal{N}\left(0, \frac{\partial q_{\lambda}(\theta)}{\partial \lambda}|_{\lambda=\bar{\lambda}^M} \Sigma_{\lambda} \frac{\partial q_{\lambda}(\theta)}{\partial \lambda}|_{\lambda=\bar{\lambda}^M}^{\top}\right).$$

Furthermore, if θ satisfies $\frac{\partial q_{\lambda}(\theta)}{\partial \lambda}|_{\lambda=\bar{\lambda}} \neq 0$,

$$\sqrt{M}(q_{\bar{\lambda}^M}(\theta) - q_{\bar{\lambda}}(\theta)) \xrightarrow{d} \mathcal{N}\left(0, \frac{\partial q_{\lambda}(\theta)}{\partial \lambda}|_{\lambda=\bar{\lambda}} \nabla^2 L(\bar{\lambda})^{-1} \text{Var}_u(h(u; \bar{\lambda})) \nabla^2 L(\bar{\lambda})^{-\top} \frac{\partial q_{\lambda}(\theta)}{\partial \lambda}|_{\lambda=\bar{\lambda}}^{\top}\right).$$

This conclusion is directly derived from the Delta Method (Vaart, 1998). Let $q_{\bar{\lambda}}(\theta)$ be the projection of the true posterior to the variational parameter family $\{q_{\lambda}(\theta)\}$. That is to say $\bar{\lambda}$ is the root of the gradient of ELBO: $\nabla_{\lambda} L(\bar{\lambda}) = 0$. We make the following assumption.

Assumption 23 The variational parameter family $\{q_{\lambda}(\theta)\}$ satisfies: $|q_{\lambda_1}(\theta) - q_{\lambda_2}(\theta)| \leq L\|\lambda_1 - \lambda_2\|$, uniformly with respect to θ .

Under Assumption 23, we have the uniform convergence results of the posterior density function.

Proposition 24 *Under Assumptions 1, 3.1-3.8 and 23, the approximate posterior density function obtained by the algorithm converges uniformly to the $q_{\bar{\lambda}}(\theta)$:*

$$\lim_{M \rightarrow \infty} \lim_{k \rightarrow \infty} \sup_{\theta} |q_{\lambda_k^M}(\theta) - q_{\bar{\lambda}}(\theta)| = 0.$$

Similarly, we can study the uniform convergence rate of $q_{\lambda_k^M}(\theta)$.

Proposition 25 *Under Assumptions 1, 3.1-3.8, 9.1-9.2, and 23, we have*

$$\begin{aligned} \sup_{\theta} |q_{\lambda_k^M}(\theta) - q_{\bar{\lambda}}(\theta)| &= O_p(\beta_k^{\frac{1}{2}} N^{-\frac{1}{2}}) + O_p(M^{-\frac{1}{2}}), \\ \mathbb{E}[\sup_{\theta} \|q_{\lambda_k^M}(\theta) - q_{\bar{\lambda}}(\theta)\|] &= O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right) + O\left(\sqrt{\frac{1}{M}}\right). \end{aligned}$$

The proofs of Proposition 24 and Proposition 25 are directly derived from Assumption 23 and the convergence rate of λ_k^M .

Proof of Strong Convergence

Proof of Proposition 2 Define the parametric function class $\mathcal{C} = \{f_\lambda(x) = h(x; \lambda) : \lambda \in \Lambda\}$. $\mathcal{C}$ is a collection of measurable functions indexed by a bounded set $\Lambda \subset \mathbb{R}^l$. Due to Assumption 1, $\mathcal{C}$ is a P-Donsker by Example 19.7 in (Vaart (1998), Chap 19). This implies

$$\sup_{f \in \mathcal{C}} |\mathbb{P}_n f - Pf| \xrightarrow{a.s.} 0,$$

so the almost sure convergence is uniform with respect to λ . The functional CLT also holds. $\square$

To prove Theorem 4, we will first prove two essential lemmas that ensure the iterated sequence $D_{k,m}$ possesses uniform boundedness almost surely on each trajectory, which plays a crucial role in the subsequent convergence theory.

Lemma 2 Assuming that Assumptions 3.1, 3.2, 3.3, and 3.5(a) hold, it follows that $\sup_{k,m} \mathbb{E}[\|D_{k,m}\|^2] < \infty$.

Proof According to the iteration formula in each parallel block, $D_{k+1,m} = (I - \alpha_k G_{2,k,m})D_{k,m} + \alpha_k G_{1,k,m}$, then we have

$$\|D_{k+1,m}\|^2 \leq \|I - \alpha_k G_{2,k,m}\|^2 \|D_{k,m}\|^2 + 2\alpha_k \|I - \alpha_k G_{2,k,m}\| \cdot \|D_{k,m}\| \cdot \|G_{1,k,m}\| + \alpha_k^2 \|G_{1,k,m}\|^2.$$

Notice the definition of $\mathcal{F}_k$, take the conditional expectation on both sides, we can get

$$\begin{aligned} & \mathbb{E}[\|D_{k+1,m}\|^2 | \mathcal{F}_k] \\ & \leq \mathbb{E}[\|I - \alpha_k G_{2,k,m}\|^2 | \mathcal{F}_k] \cdot \|D_{k,m}\|^2 + 2\alpha_k \mathbb{E}[\|I - \alpha_k G_{2,k,m}\| \cdot \|G_{1,k,m}\| | \mathcal{F}_k] \cdot \|D_{k,m}\| + \alpha_k^2 \mathbb{E}[\|G_{1,k,m}\|^2 | \mathcal{F}_k] \quad (\text{B1}) \\ & \leq \mathbb{E}[\|I - \alpha_k G_{2,k,m}\|^2 | \mathcal{F}_k] \|D_{k,m}\|^2 + 2\alpha_k \sqrt{\mathbb{E}[\|G_{1,k,m}\|^2 | \mathcal{F}_k]} \sqrt{\mathbb{E}[\|I - \alpha_k G_{2,k,m}\|^2 | \mathcal{F}_k]} \|D_{k,m}\| + \alpha_k^2 C_1. \end{aligned}$$

The second inequality comes from Cauchy-Schwarz(C-S) inequality and Assumption 3.1. Note that

$$\mathbb{E}[\|I - \alpha_k G_{2,k,m}\|^2 | \mathcal{F}_k] = \mathbb{E}[(1 - \alpha_k \lambda_{2,k,m})^2 | \mathcal{F}_k] = 1 - \alpha_k (2\mathbb{E}[\lambda_{2,k,m} | \mathcal{F}_k] - \alpha_k \mathbb{E}[\lambda_{2,k,m}^2 | \mathcal{F}_k]) \leq 1 - \alpha_k \epsilon,$$

where $\lambda_{2,k,m}$ is the minimum eigenvalue of $G_{2,k,m}$. Due to Assumptions 3.2 and 3.3, the inequality in the above expression arises because $\alpha_k \rightarrow 0$, there exists $N_1 > 0$ and N_1 is independent of u , such that for every $k \geq N_1$, $2\mathbb{E}[\lambda_{2,k,m} | \mathcal{F}_k] - \alpha_k \mathbb{E}[\lambda_{2,k,m}^2 | \mathcal{F}_k] \geq 2\epsilon - \alpha_k C_2 \geq \epsilon$ w.p.1. So Equation (B1) can be changed to

$$\mathbb{E}[\|D_{k+1,m}\|^2 | \mathcal{F}_k] \leq (1 - \alpha_k \epsilon) \|D_{k,m}\|^2 + \alpha_k^2 C_1 + 2\alpha_k \sqrt{C_1} \sqrt{1 - \alpha_k \epsilon} \|D_{k,m}\|.$$

Take the expectation and apply the C-S inequality, the inequality holds for every $k \geq N_1$,

$$\begin{aligned} \mathbb{E}[\|D_{k+1,m}\|^2] &\leq (1 - \alpha_k \epsilon) \mathbb{E}[\|D_{k,m}\|^2] + \alpha_k^2 C_1 + 2\alpha_k \sqrt{C_1} \sqrt{1 - \alpha_k \epsilon} \sqrt{\mathbb{E}[\|D_{k,m}\|^2]} \\ &= \left(\sqrt{1 - \alpha_k \epsilon} \sqrt{\mathbb{E}[\|D_{k,m}\|^2]} + \alpha_k \sqrt{C_1} \right)^2 \leq \left(\left(1 - \frac{\alpha_k \epsilon}{2}\right) \sqrt{\mathbb{E}[\|D_{k,m}\|^2]} + \frac{\alpha_k \epsilon}{2} \frac{2\sqrt{C_1}}{\epsilon} \right)^2 \\ &\leq \left(\max_k \left\{ \sqrt{\mathbb{E}[\|D_{k,m}\|^2]}, \frac{2\sqrt{C_1}}{\epsilon} \right\} \right)^2. \end{aligned}$$

Since D_0 is independent of u , by using the boundness assumption, taking the expectation and taking superior with respect to m in Equation (B1), it is easy to prove by induction that for every $k \leq N_1$, $\sup_m \mathbb{E}[\|D_{k,m}\|^2] < \infty$. Therefore, $\sup_{k,m} \mathbb{E}[\|D_{k,m}\|^2] \leq \max_{k \leq N_1} \sup_m \mathbb{E}[\|D_{k,m}\|^2] + \frac{4C_1}{\epsilon^2} < \infty$. $\square$

Lemma 3 Assuming Assumptions 3.1, 3.2, 3.3, 3.5(a) hold, $\sup_{k,m} \|D_{k,m}\|^2 < \infty$, w.p.1.

Proof Rewrite the iteration as

$$\begin{aligned} D_{k+1,m} &= (I - \alpha_k G_{2,k,m}) D_{k,m} + \alpha_k G_{1,k,m} \\ &= (I - \alpha_k \mathbb{E}[G_{2,k,m} | \mathcal{F}_k]) D_{k,m} + \alpha_k \mathbb{E}[G_{1,k,m} | \mathcal{F}_k] + \alpha_k W_{k,m} + \alpha_k V_{k,m} \quad (\text{B2}) \\ &= (I - U_{k,m}) D_{k,m} + \tilde{\alpha}_k R_{k,m} + \alpha_k W_{k,m} + \alpha_k V_{k,m}, \end{aligned}$$

where $W_{k,m} = (\mathbb{E}[G_{2,k,m} | \mathcal{F}_k] - G_{2,k,m}) D_{k,m}$, $V_{k,m} = G_{1,k,m} - \mathbb{E}[G_{1,k,m} | \mathcal{F}_k]$, $U_{k,m} = \alpha_k \mathbb{E}[G_{2,k,m} | \mathcal{F}_k]$, $R_{k,m} = \mathbb{E}[G_{2,k,m} | \mathcal{F}_k]^{-1} \mathbb{E}[G_{1,k,m} | \mathcal{F}_k]$. By Assumptions 3.2 and 3.3, $U_{k,m}$ is a diagonal matrix and all of its elements are no less than $\alpha_k \epsilon$ and no more than $\alpha_k \sqrt{C_2}$. Since α_k tends to zero, there exists $N_2 > 0$, for every $k \geq N_2$, all of elements of $U_{k,m}$ are less than 1. Define some of the element of $U_{k,m}$ as $\tilde{\alpha}_k$ and $\alpha_k \epsilon < \tilde{\alpha}_k < \alpha_k \sqrt{C_2} < 1$. So $\forall k \geq N_2$, take norm on both sides of Equation (B2):

$$\begin{aligned} \|D_{k+1,m}\| &\leq \prod_{i=N_2}^k (1 - \tilde{\alpha}_i) \|D_{N_2,m}\| + \sum_{i=N_2}^k \prod_{j=i+1}^k (1 - \tilde{\alpha}_j) \tilde{\alpha}_i \|R_{i,m}\| \\ &\quad + \left\| \sum_{i=N_2}^k \prod_{j=i+1}^k (1 - \tilde{\alpha}_j) \alpha_i W_{i,m} \right\| + \left\| \sum_{i=N_2}^k \prod_{j=i+1}^k (1 - \tilde{\alpha}_j) \alpha_i V_{i,m} \right\|. \end{aligned} \quad (\text{B3})$$

(1) For the first term, by Assumption 3.5, $\sum_{k=0}^{\infty} \alpha_k = \infty$, when $k \rightarrow \infty$. We have the inequality:

$$\prod_{i=N_2}^k (1 - \tilde{\alpha}_i) \|D_{N_2,m}\| \leq e^{-\sum_{i=N_2}^k \tilde{\alpha}_i} \|D_{N_2,m}\| \leq e^{-\epsilon \sum_{i=N_2}^k \alpha_i} \|D_{N_2,m}\| \rightarrow 0.$$

(2) For the second term, by the C-S inequality and Assumption 3.1, we have

$$\|R_{i,m}\| = \frac{\|\mathbb{E}[G_{1,i,m} | \mathcal{F}_i]\|}{\|\mathbb{E}[G_{2,i,m} | \mathcal{F}_i]\|} \leq \frac{\mathbb{E}[\|G_{1,i,m}\| | \mathcal{F}_i]}{\mathbb{E}[\|G_{2,i,m}\| | \mathcal{F}_i]} \leq \frac{\sqrt{C_1}}{\epsilon}, \quad \text{w.p.1.}$$

We prove this by induction: $\sum_{i=N_2}^k \prod_{j=i+1}^k (1 - \tilde{\alpha}_j) \tilde{\alpha}_i \leq 1$. It is easy to check that the conclusion holds when $k = N_2$. Assume that the assumption holds for k . Then for $k + 1$, we plug in the inequality of k , and noting that $0 < \tilde{\alpha}_k < 1$, we have

$\sum_{i=N_2}^{k+1} \prod_{j=i+1}^{k+1} (1 - \tilde{\alpha}_j) \tilde{\alpha}_i \leq \prod_{j=i+1}^{k+1} (1 - \tilde{\alpha}_j) \tilde{\alpha}_{k+1} + (1 - \tilde{\alpha}_{k+1}) \leq \tilde{\alpha}_{k+1} + 1 - \tilde{\alpha}_{k+1} = 1$, which implies the second term of Equation (B3) is bounded.

(3) For the third term, since $W_{k,m} = (\mathbb{E}[G_{2,k,m}|\mathcal{F}_k] - G_{2,k,m})D_{k,m}$, and $D_{k,m} \in \mathcal{F}_k$, so $\mathbb{E}[W_{k,m}|\mathcal{F}_k] = 0$. Moreover,

$$\mathbb{E}\left[\sum_{i=N_2}^k \alpha_i W_{i,m}|\mathcal{F}_k\right] = \sum_{i=N_2}^k \alpha_i \mathbb{E}[W_{i,m}|\mathcal{F}_k] = \sum_{i=N_2}^{k-1} \alpha_i \mathbb{E}[(\mathbb{E}[G_{2,i,m}|\mathcal{F}_i] - G_{2,i,m})D_{i,m}|\mathcal{F}_k] = \sum_{i=N_2}^{k-1} \alpha_i W_{i,m}.$$

Thus, $\sum_{i=N_2}^k \alpha_i W_{i,m}$ is a martingale sequence for every m . Note that for every $i < j$,

$$\mathbb{E}[\langle W_{i,m}, W_{j,m} \rangle] = \mathbb{E}[\mathbb{E}[\langle W_{i,m}, W_{j,m} \rangle | \mathcal{F}_j]] = \mathbb{E}[\langle W_{i,m}, \mathbb{E}[W_{j,m} | \mathcal{F}_j] \rangle] = 0,$$

so we can derive that

$$\begin{aligned} \mathbb{E}[\|\sum_{i=N_2}^k \alpha_i W_{i,m}\|^2] &= \sum_{i=N_2}^k \alpha_i^2 \mathbb{E}[\|W_{i,m}\|^2] \leq \sum_{i=N_2}^k \alpha_i^2 \mathbb{E}[\|\mathbb{E}[G_{2,i,m}|\mathcal{F}_i] - G_{2,i,m}\|^2 \cdot \|D_{i,m}\|^2] \\ &= \sum_{i=N_2}^k \alpha_i^2 \mathbb{E}[\mathbb{E}[\|\mathbb{E}[G_{2,i,m}|\mathcal{F}_i] - G_{2,i,m}\|^2 \|D_{i,m}\|^2 | \mathcal{F}_i]] \\ &= \sum_{i=N_2}^k \alpha_i^2 \mathbb{E}[\|D_{i,m}\|^2 (\mathbb{E}[\|\mathbb{E}[G_{2,i,m}|\mathcal{F}_i]\|^2] - 2\langle \mathbb{E}[G_{2,i,m}|\mathcal{F}_i], G_{2,i,m} \rangle + \|G_{2,i,m}\|^2 | \mathcal{F}_i)] \\ &= \sum_{i=N_2}^k \alpha_i^2 \mathbb{E}[\|D_{i,m}\|^2 (\mathbb{E}[\|G_{2,i}\|^2 | \mathcal{F}_i] - \|\mathbb{E}[G_{2,i,m}|\mathcal{F}_i]\|^2)] \\ &\leq \sum_{i=N_2}^k \alpha_i^2 \mathbb{E}[\|D_{i,m}\|^2 \mathbb{E}[\|G_{2,i,m}\|^2 | \mathcal{F}_i]] = C_2 \sum_{i=N_2}^k \alpha_i^2 \mathbb{E}[\|D_{i,m}\|^2] < \infty, \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ represents the inner product of two matrices, the last inequality holds because of Assumptions 3.3, 3.5(a) and Lemma 2. So $\sum_{i=N_2}^k \alpha_i W_{i,m}$ is an $\mathbb{L}^2$ martingale. By the martingale convergence theorem, for every u , it converges.

Let $a_i = \prod_{j=N_2}^i \frac{1}{1 - \tilde{\alpha}_j}$, and $\forall i > N_2, 0 < a_i \leq a_{i+1}$, we have $\lim_{i \rightarrow \infty} a_i \geq \lim_{i \rightarrow \infty} e^{\sum_{j=N_2}^i \tilde{\alpha}_j} \geq \lim_{i \rightarrow \infty} e^{\epsilon \sum_{j=N_2}^i \alpha_j} = \infty$. Furthermore,

$$\sum_{i=N_2}^k \prod_{j=i+1}^k (1 - \tilde{\alpha}_j) \alpha_i W_{i,m} = \prod_{j=N_2}^k (1 - \tilde{\alpha}_j) \sum_{i=N_2}^k \frac{1}{\prod_{j=N_2}^i (1 - \tilde{\alpha}_j)} \alpha_i W_{i,m} = \frac{1}{a_k} \sum_{i=N_2}^k a_i \alpha_i W_{i,m}.$$

Because of $\sum_{i=N_2}^{\infty} \alpha_i W_{i,m} < \infty$, and $\lim_{i \rightarrow \infty} a_i = \infty$, by Kronecker's Lemma (Shiryaev and Boas, 1995) we can reach the conclusion that for every m , $\lim_{k \rightarrow \infty} \frac{1}{a_k} \sum_{i=N_2}^k a_i \alpha_i W_{i,m} = 0$.

Thus, $\lim_{k \rightarrow \infty} \sup_m \left\| \sum_{i=N_2}^k \prod_{j=i+1}^k (1 - \tilde{\alpha}_j) \alpha_i W_{i,m} \right\| = 0$. The uniform convergence is obvious because the supremum is taken in a finite set. A similar conclusion can be drawn for part (4), $\lim_{k \rightarrow \infty} \sup_m \left\| \sum_{i=N_2}^k \prod_{j=i+1}^k (1 - \tilde{\alpha}_j) \alpha_i V_{i,m} \right\| = 0$. All the inequalities hold uniformly with respect to m , so by Equation (B3), $\sup_{k,m} \|D_{k,m}\|^2 < \infty$, $w.p.1$, which ends the proof. $\square$

Next, we proceed to prove the main part of the convergence theory. The key idea is to transform the discrete sequence $\{D_{k,m}, \lambda_k\}$ into a continuous form. The iterative formulas (8) and (9) are approximated by a system of ODEs. First, we construct the corresponding step interpolation functions $\{D_m^k(t), \lambda^k(t)\}$ for the sequence. Then, we demonstrate that these functions $\{D_m^k(t), \lambda^k(t)\}$ converge to a solution of the ODE as the number of iterations becomes sufficiently large. The asymptotic stability point of this ODE corresponds to the limiting point of the iterative sequence $\{D_{k,m}, \theta_k\}$. Finally, we show that the condition satisfied by this convergence point is $D = 0$, $\lambda = \bar{\lambda}^M$.

We begin the process of continuity by introducing the notation. Let $t_0 = 0$, $t_n = \sum_{i=0}^{n-1} \alpha_i$. Define $m(t) = \max\{n : t_n \leq t\}$ for $t \geq 0$, and $m(t) = 0$ for $t < 0$. The function $m(t)$ represents the number of iterations that have occurred by the time t .

Define the piecewise constant interpolation function for D_k : $D_m^0(t) = D_{k,m}$, $\forall t_k \leq t < t_{k+1}$, $D_m^0(t) = D_{0,m}$, $\forall t < 0$. Define the translation process $D_m^n(t) = D_m^0(t_n + t)$, $t \in (-\infty, \infty)$.

Similarly define the piecewise constant interpolation function $\lambda^0(t)$ and the translation function $\lambda^n(t)$ of λ as $\lambda^0(t) = \lambda_k$, $\forall t_k \leq t < t_{k+1}$; $\lambda^0(t) = 0$, $\forall t \leq t_0$. $\lambda^n(t) = \lambda^0(t_n + t)$, $t \in (-\infty, \infty)$.

Rewrite the m th block of iterative equation (8) as

$$D_{k+1,m} = D_{k,m} + \alpha_k (H(u_m, D_{k,m}, \lambda_k) + b_{1,k} + b_{2,k} + V_{k,m} + W_{k,m}), \quad (\text{B4})$$

where

$$H(u, D, \lambda) := \nabla_{\theta} p(y|\theta)|_{\theta=\theta(u;\lambda)} - p(y|\theta(u;\lambda))D, \\ b_{1,k} := \mathbb{E}[G_{1,k,m}|\mathcal{F}_k] - \nabla_{\theta} p(y|\theta)|_{\theta=\theta_{k,m}} = 0, \quad b_{2,k} := p(y|\theta_{k,m})D_{k,m} - \mathbb{E}[G_{2,k}|\mathcal{F}_k]D_{k,m} = 0, \\ V_{k,m} := G_{1,k,m} - \mathbb{E}[G_{1,k,m}|\mathcal{F}_k], \quad W_{k,m} := \mathbb{E}[G_{2,k,m}|\mathcal{F}_k]D_{k,m} - G_{2,k,m}D_{k,m}, \quad \text{where } b_{1,k} \text{ and } b_{2,k} \text{ are equal to 0 due to Assumption 3.4.}$$

Define $H_{k,m} = H(u_m, D_{k,m}, \lambda_k)$ for $k \geq 0$, and $H_{k,m} = 0$ for $k < 0$. Define the piecewise constant interpolation function of H as $H_m^0(t) = \sum_{i=0}^{m(t)-1} \alpha_i H_{i,m}$, and the translation function of H as

$$H_m^n(t) = H_m^0(t + t_n) - H_m^0(t_n) = \sum_{i=n}^{m(t_n+t)-1} \alpha_i H_{i,m}, \quad t \geq 0; \quad H_m^n(t) = \sum_{i=m(t_n+t)}^{n-1} \alpha_i H_{i,m}, \quad t < 0.$$

Two other terms of Equation (B4) are similarly defined; for simplicity, we omit the definition part of the negative numbers: $V_m^n(t) = \sum_{i=n}^{m(t_n+t)-1} \alpha_i V_{i,m}$, $W_m^n(t) = \sum_{i=n}^{m(t_n+t)-1} \alpha_i W_{i,m}$. Make the Equation (B4) continuous and we can get

$$\begin{aligned} D_m^n(t) &= D_{n,m} + \sum_{i=n}^{m(t_n+t)-1} \alpha_i (H_{i,m} + V_{i,m} + W_{i,m}) = D_m^n(0) + H_m^n(t) + V_m^n(t) + W_m^n(t) \\ &= D_m^n(0) + \int_0^t H(u_m, D_m^n(s), \lambda^n(s)) ds + \rho_m^n(t) + V_m^n(t) + W_m^n(t), \end{aligned} \quad (\text{B5})$$

where $\rho_m^n(t) = H_m^n(t) - \int_0^t H(u_m, D_m^n(s), \lambda^n(s)) ds$. Since $\lambda_{k+1} = \lambda_k + \beta_k S_k + \beta_k Z_k$, define $\lambda_{k+1} = \lambda_k + \alpha_k \tilde{D}_k$, where $\tilde{D}_k = \frac{\beta_k}{\alpha_k} (S_k + Z_k)$. Define $\eta^n(t) = \sum_{i=n}^{m(t_n+t)-1} \alpha_i \tilde{D}_i$, we obtains the continuation of λ_n as

$$\lambda^n(t) = \lambda^n(0) + \eta^n(t). \quad (\text{B6})$$

The following lemmas reveal that $\rho_m^n(t)$, $V_m^n(t)$, $W_m^n(t)$, $\eta^n(t)$ all converge uniformly to 0 in a bounded interval of t . As a result, these terms can be neglected, and the asymptotic behavior of these continuous processes is governed by a system of ODEs.

Lemma 4 Assuming that Assumptions 3.1 to 3.5 hold, and that $\mathbb{T}$ is a bounded interval on $\mathbb{R}$, we have $\lim_{n \rightarrow \infty} \sup_{t \in \mathbb{T}, m} \|\rho_m^n(t)\| = 0, w.p.1$.

Proof Given $T > 0$, consider an arbitrary time $t \in [0, T]$. If there exists an integer d such that $t = t_{n+d} - t_n$, then

$$\begin{aligned} \rho_m^n(t) &= H_m^n(t) - \int_0^t H(u_m, D_m^n(s), \lambda^n(s)) ds = \sum_{i=n}^{m(t_n+t)-1} \alpha_i H_{i,m} - \int_0^{t_{n+d}-t_n} H(u_m, D_m^n(s), \lambda^n(s)) ds \\ &= \sum_{i=n}^{m(t_{n+d})-1} \alpha_i H_{i,m} - \int_{t_n}^{t_{n+d}} H(u_m, D_m^n(s - t_n), \lambda^n(s - t_n)) ds = \sum_{i=n}^{n+d-1} \alpha_i H_{i,m} - \int_{t_n}^{t_{n+d}} H(u_m, D_m^0(s), \theta^0(s)) ds = 0. \end{aligned}$$

The last equality sign comes from the definition of $H_{i,m}$: $H_{i,m} = H(u_m, D_{i,m}, \lambda_i)$. If there exists an integer d satisfying $t_{n+d} < t_n + t < t_{n+d+1}$, then

$$\rho_m^n(t) = \sum_{i=n}^{n+d-1} \alpha_i H_{i,m} - \int_{t_n}^{t_{n+d}} H(u_m, D_m^0(s), \theta^0(s)) ds = - \int_{t_{n+d}}^{t_n+t} H(u_m, D_m^0(s), \theta^0(s)) ds.$$

Also by Assumptions 3.1, 3.3 and 3.4, $\|\nabla_{\theta} p(y|\theta)|_{\theta=\theta(u;\lambda_k)}\| = \|\mathbb{E}[G_{1,k,m}|\mathcal{F}_k]\| \leq \sqrt{C_1}$, $\|p(y|\theta(u;\lambda_k))\| = \|\mathbb{E}[G_{2,k,m}|\mathcal{F}_k]\| \leq \sqrt{C_2}$. Furthermore, note that

$$\|H(u_m, D_{k,m}, \lambda_k)\| = \|\nabla_{\theta} p(y|\theta)|_{\theta=\theta(u;\lambda_k)} - p(y|\theta) D_{k,m}\| \leq \sqrt{C_1} + \sqrt{C_2} \|D_{k,m}\| \leq \bar{C}, w.p.1,$$

where the last equality sign comes from Lemma 3 with $\|D_{k,m}\|$ being uniformly bounded. This leads to

$$\|\rho_m^n(t)\| \leq \left\| \int_{t_{n+d}}^{t_n+t} H(u_m, D_m^0(s), \theta^0(s)) ds \right\| \leq \alpha_{n+d} \bar{C}.$$

This holds for almost every orbit, the right end being independent of t and u . By Assumption 3.5, $\alpha_k \rightarrow 0$, this leads to the conclusion that $\lim_{n \rightarrow \infty} \sup_{t \in \mathbb{T}, m} \|\rho_m^n(t)\| = 0$, w.p.1. $\square$

Lemma 5 *Assuming that Assumptions 3.1 to 3.5 hold, and that $\mathbb{T}$ is a bounded interval on $\mathbb{R}$, then when $n \rightarrow \infty$, $\sup_{t \in \mathbb{T}, m} \|V_m^n(t)\| \rightarrow 0$ w.p.1.*

Proof Let $M_{n,m} = \sum_{i=0}^{n-1} \alpha_i V_{i,m}$, so

$$\mathbb{E}[M_{n,m} | \mathcal{F}_{n-1}] = \sum_{i=0}^{n-1} \alpha_i \mathbb{E}[(G_{1,i,m} - \mathbb{E}[G_{1,i,m} | \mathcal{F}_i]) | \mathcal{F}_{n-1}] = \sum_{i=0}^{n-2} \alpha_i (G_{1,i,m} - \mathbb{E}[G_{1,i,m} | \mathcal{F}_i]) = M_{n-1,m}.$$

Thus $M_{n,m}$ is a martingale for every m . Note that for every $i < j$, $\mathbb{E}[\langle V_{i,m}, V_{j,m} \rangle] = \mathbb{E}[\mathbb{E}[\langle V_{i,m}, V_{j,m} \rangle | \mathcal{F}_j]] = \mathbb{E}[\langle V_{i,m}, \mathbb{E}[V_{j,m} | \mathcal{F}_j] \rangle] = 0$, so we can derive that:

$$\begin{aligned} \mathbb{E}[\|M_{n,m}\|^2] &= \mathbb{E}\left[\left\|\sum_{i=0}^{n-1} \alpha_i V_{i,m}\right\|^2\right] = \sum_{i=0}^{n-1} \alpha_i^2 \mathbb{E}[\|V_{i,m}\|^2] = \sum_{i=0}^{n-1} \alpha_i^2 \mathbb{E}[\|G_{1,i,m} - \mathbb{E}[G_{1,i,m} | \mathcal{F}_i]\|^2] \\ &= \sum_{i=0}^{n-1} \alpha_i^2 \mathbb{E}[\mathbb{E}[\|G_{1,i,m} - \mathbb{E}[G_{1,i,m} | \mathcal{F}_i]\|^2 | \mathcal{F}_i]] = \sum_{i=0}^{n-1} \alpha_i^2 \mathbb{E}[\mathbb{E}[\|G_{1,i,m}\|^2 - 2\langle G_{1,i,m}, \mathbb{E}[G_{1,i,m} | \mathcal{F}_i] \rangle + \|\mathbb{E}[G_{1,i,m} | \mathcal{F}_i]\|^2 | \mathcal{F}_i]] \\ &= \sum_{i=0}^{n-1} \alpha_i^2 (\mathbb{E}[\mathbb{E}[\|G_{1,i,m}\|^2 | \mathcal{F}_i]] - \|\mathbb{E}[G_{1,i,m} | \mathcal{F}_i]\|^2) \leq \sum_{i=0}^{n-1} \alpha_i^2 \mathbb{E}[\mathbb{E}[\|G_{1,i,m}\|^2 | \mathcal{F}_i]] \leq \sum_{i=0}^{n-1} C_1 \alpha_i^2 < \infty. \end{aligned}$$

The right-hand side is independent of m . Therefore, $M_{n,m}$ is an $\mathbb{L}^2$ martingale for every m and by the martingale convergence theorem $\lim_n \sup_m \|M_{n,m} - M_m\| = 0$. The uniform convergence is obvious because the supremum is taken in a finite set. So when $n \rightarrow \infty$

$$\sup_{t \in \mathbb{T}, m} \|V_m^n(t)\| = \sup_{t \in \mathbb{T}, m} \left\| \sum_{i=n}^{m(t_n+t)-1} \alpha_i V_{i,m} \right\| = \sup_{t \in \mathbb{T}, m} \|M_{m(t_n+t), m} - M_{n,m}\| \rightarrow 0,$$

i.e., $\sup_{t \in \mathbb{T}, m} \|V_m^n(t)\| \rightarrow 0$ w.p.1 when $n \rightarrow \infty$. $\square$

Lemma 6 *Assuming that Assumptions 3.1-3.5 hold, $\mathbb{T}$ is a bounded interval on $\mathbb{R}$, when $n \rightarrow \infty$, $\sup_{t \in \mathbb{T}, m} \|W_m^n(t)\| \rightarrow 0$ w.p.1.*

Proof Define $M'_{n,m} = \sum_{i=0}^{n-1} \alpha_i w_{i,m}$, then

$M'_{n,m}$ is a martingale sequence by Lemma 3. Similar to lemma 4, we can prove that $\mathbb{E}[\|M'_n\|^2] \leq \sum_{i=0}^{n-1} C_2 \alpha_i^2 \mathbb{E}[\|D_i\|^2] \leq \infty$, so $M'_{n,m}$ is an $\mathbb{L}^2$ martingale. By the martingale convergence theorem, we can reach the same conclusion. $\square$

Lemma 7 *Assuming Assumptions 3.1-3.6 hold, $\mathbb{T}$ is a bounded interval on $\mathbb{R}$, then $\lim_{n \rightarrow \infty} \sup_{t \in \mathbb{T}} \|\eta^n(t)\| = 0$, w.p.1.*

Proof Given $T_0 > 0$, for every $t \in [0, T_0]$, $\lambda_{k+1} = \lambda_k + \beta_k(S_k + Z_k)$, the direction of Z_k is the projection direction from $\lambda_k + \beta_k S_k$ to feasible region Λ . By the property of the projection operator, Z_k satisfies $Z_k^\top(\lambda_k - \lambda_{k+1}) \geq 0$, $\forall \lambda \in \Lambda$. Furthermore,

$$0 \geq -Z_k^\top(\lambda_k - \lambda_{k+1}) = -Z_k^\top(-\beta_k(S_k + Z_k)) = \beta_k Z_k^\top S_k + \beta_k \|Z_k\|^2.$$

Thus, $0 \leq \beta_k \|Z_k\|^2 \leq -\beta_k Z_k^\top S_k \leq \beta_k \|Z_k\| \cdot \|S_k\|$. Therefore, $\|Z_k\| \leq \|S_k\|$ and $\|\tilde{D}_k\| = \|\frac{\beta_k}{\alpha_k}(Z_k + S_k)\| \leq \frac{\beta_k}{\alpha_k}(\|Z_k\| + \|S_k\|) \leq \frac{2\beta_k \|S_k\|}{\alpha_k}$. We can get boundness of $\|S_k\|$ due to Assumption 3.7 and the boundness of $\|D_k\|$ and other terms. So when $k \rightarrow \infty$,

$$\|\eta^n(t)\| = \left\| \sum_{k=n}^{m(t_n+t)-1} \alpha_k \tilde{D}_k \right\| \leq T_0 \sup_{k \geq n} \|\tilde{D}_k\| \leq \frac{2\beta_k T_0}{\alpha_k} \sup_k \|S_k\| \rightarrow 0.$$

The zero limit comes from Assumption 3.6: $\beta_k = o(\alpha_k)$, which is one of the essential conditions for the convergence of NMTS algorithms. Then $\lim_{n \rightarrow \infty} \sup_{t \in \mathbb{T}} \|\eta^n(t)\| = 0$ w.p.1. $\square$

Relate Equation (B5) and Equation (B6):

$$\begin{cases} D_1^n(t) = D_1^n(0) + \int_0^t H(u_1, D_1^n(s), \lambda^n(s)) ds + \rho_{u_1}^n(t) + V_{u_1}^n(t) + W_{u_1}^n(t) \\ \dots \\ D_M^n(t) = D_M^n(0) + \int_0^t H(u_M, D_M^n(s), \lambda^n(s)) ds + \rho_{u_M}^n(t) + V_{u_M}^n(t) + W_{u_M}^n(t) \\ \lambda^n(t) = \lambda^n(0) + \eta^n(t). \end{cases} \quad (\text{B7})$$

We show below, by the asymptotic property of this set of ODEs, that the sequence $D_{k,m}$ converges uniformly to the gradient $\nabla_\theta \log p(y|\theta)|_{\theta=\theta_{k,m}}$, where $\nabla_\theta \log p(y|\theta)|_{\theta=\theta_{k,m}}$ is a long vector with $T \times d$ dimensions and the t th block $\frac{\nabla_\theta p(Y_t|\theta_{k,m})}{p(Y_t|\theta_{k,m})}$.

Proof of Theorem 4 By Lemma 3, $D_{k,m}$ is uniformly bounded, and λ_k is also uniformly bounded by the projection operator. The functions $D_m^n(t)$ and $\lambda^n(t)$ are constructed by interpolating $D_{k,m}$ and λ_k , it follows that $\{D_m^n(t)\}_{m=1}^M$ and $\lambda^n(t)$ are uniformly bounded for almost every orbit. On the other hand, by Lemmas 4-7, the sequences $\{D_m^n(t)\}_{m=1}^M$ and $\lambda^n(t)$ are equicontinuous along almost every sample path on every finite interval. Applying the Arzelà-Ascoli theorem, we conclude that there exists a uniformly convergent subsequence of $\{D_m^n(t)\}_{m=1}^M$ and $\lambda^n(t)$ for almost every orbit. Let the limit of this subsequence be $\{D_m(t)\}_{m=1}^M$ and $\lambda(t)$.

Note that in Lemma 2, we proved that H is uniformly bounded for almost all orbits. By the dominated convergence theorem, we can interchange the integrals and limits when taking the limit. Taking $n \rightarrow \infty$ in Equation (B7) and applying the uniform convergence established in Lemmas 4-7, Equation (B7) simplifies to

$$\begin{cases} D_1(t) = D_1(0) + \int_0^t H(u_1, D_{u_1}(s), \lambda(s)) ds \\ \dots \\ D_M(t) = D_M(0) + \int_0^t H(u_M, D_m(s), \lambda(s)) ds \\ \lambda(t) = \lambda(0). \end{cases}$$

Its differential form is

$$\begin{cases} \dot{D}_1(t) = H(u_1, D_1(t), \lambda(t)) \\ \dots \\ \dot{D}_M(t) = H(u_M, D_M(t), \lambda(t)) \\ \dot{\lambda}(t) = 0. \end{cases}$$

Then

$$\lambda(t) = \lambda(0) := \tilde{\lambda}, \quad \dot{D}_{u_m}(t) = H(u_m, D_m(t), \tilde{\lambda}) = \nabla_{\theta} p(y|\theta(u; \lambda))|_{(u; \lambda) = (u_m; \tilde{\lambda})} - p(Y, \tilde{\lambda}) D_m(t).$$

This is a first-order linear ODE for a matrix D . For every u , construct the Lyapunov function as

$$V(t) = \frac{1}{2} \|\nabla_{\theta} p(y|\theta)|_{\theta = \theta(u; \tilde{\lambda})} - p(y|\theta(u; \tilde{\lambda})) D_m(t)\|^2,$$

then

$$\dot{V} = -tr \left(p(y|\theta(u; \tilde{\lambda})) \left(\nabla_{\theta} p(y|\theta(u; \tilde{\lambda})) - p(y|\theta(u; \tilde{\lambda})) D_m(t) \right) \cdot \left(\nabla_{\theta} p(y|\theta(u; \tilde{\lambda})) - p(y|\theta(u; \tilde{\lambda})) D_m(t) \right)^{\top} \right) < 0,$$

so $D_m(t)$ has unique global asymptotic stable point of $p(y|\theta(u; \tilde{\lambda}))^{-1} \nabla_{\theta} p(y|\theta)|_{\theta = \theta(u_m; \tilde{\lambda})}$. Since $(D_{k,m}, \lambda_k)$ and $(D_m^n(\cdot), \lambda_m^n(\cdot))$ have the same asymptotic performance, so

$$(D_{k,m}, \lambda_k) \rightarrow (p(y|\theta(u; \tilde{\lambda}))^{-1} \nabla_{\theta} p(y|\theta)|_{\theta = \theta(u; \tilde{\lambda})}, \tilde{\lambda}).$$

Note that

$$\begin{aligned} \left\| D_{k,m} - \nabla_{\theta} \log p(y|\theta(u; \lambda))|_{\theta = \theta(u_m; \lambda_k)} \right\| &\leq \left\| D_{k,m} - p(y|\theta(u; \tilde{\lambda}))^{-1} \nabla_{\theta} p(y|\theta)|_{\theta = \theta(u_m; \tilde{\lambda})} \right\| \\ &\quad + \left\| p(y|\theta(u; \tilde{\lambda}))^{-1} \nabla_{\theta} p(y|\theta)|_{\theta = \theta(u_m; \tilde{\lambda})} - (p(y|\theta_{k,m}))^{-1} \nabla_{\theta} p(y|\theta)|_{\theta = \theta(u_m; \lambda_k)} \right\|. \end{aligned}$$

The first term converges to 0 previously shown, while the second term also converges to 0 by Assumption 3.7, which states $\log p(y|\theta(u; \lambda))$ is continuously differentiable, and $\lambda_k \rightarrow \tilde{\lambda}$ when $k \rightarrow \infty$. This establishes the following convergence result. $\square$

Thus, we have proven that the sequence of D_k converges asymptotically to the gradient of the likelihood function $\log p(y|\theta(u; \lambda))$. Later, we need to confirm that the limit point $\tilde{\lambda}$ to which λ_k converges is exactly the point where the gradient is 0, i.e., $\nabla_{\lambda} \hat{L}_M(\tilde{\lambda}) = 0$.

Proof of Proposition 5 Notice that

$$A(\lambda_k)B(\lambda_k) = \sum_{m=1}^M \nabla_{\lambda} \theta(u_m; \lambda_k) \nabla_{\theta} \log p(\theta_{k,m}), \quad A(\lambda_k)C(\lambda_k) = \sum_{m=1}^M \nabla_{\lambda} \theta(u_m; \lambda_k) \nabla_{\theta} \log q_{\lambda}(\theta_{k,m}).$$

By the definition of the two notations,

$$\begin{aligned} S_k - \nabla_{\lambda} \hat{L}_M(\lambda_k) &= \frac{A(\lambda_k)}{M} \left(E^M D_k + B(\lambda_k) - C(\lambda_k) \right) \\ &- \frac{1}{M} \sum_{m=1}^M \nabla_{\lambda} \theta(u_m; \lambda_k) \left(E \nabla_{\theta} \log p(y|\theta(u_m; \lambda_k)) + \nabla_{\theta} \log p(\theta(u_m; \lambda_k)) - \nabla_{\theta} \log q_{\lambda}(\theta(u_m; \lambda_k)) \right) \\ &= \frac{1}{M} \left(A(\lambda_k) E^M D_k - \sum_{m=1}^M \nabla_{\lambda} \theta(u_m; \lambda_k) E \nabla_{\theta} \log p(y|\theta(u_m; \lambda_k)) \right) \\ &= \frac{1}{M} \sum_{m=1}^M \nabla_{\lambda} \theta(u_m; \lambda_k) E \left(D_{k,m} - \nabla_{\theta} \log p(y|\theta(u_m; \lambda_k)) \right). \end{aligned}$$

$\nabla_{\lambda} \theta(u_m; \lambda)$ is bounded since Λ is a compact set and $\theta(u_m; \lambda)$ is continuously differentiable with respect to λ . By Theorem 4, we can reach the conclusion. $\square$

Proof of Theorem 6 From the iterative equation, we have $\lambda_{k+1} = \lambda_k + \beta_k(S_k + Z_k) = \lambda_k + \beta_k h(\lambda_k) + \beta_k b_k + \beta_k Z_k$, where $h(\lambda_k) = \nabla_{\lambda} \hat{L}_M(\lambda)|_{\lambda=\lambda_k}$, $b_k = -\nabla_{\lambda} \hat{L}_M(\lambda)|_{\lambda=\lambda_k} + S_k$. Define $\zeta_0 = 0$, $\zeta_n = \sum_{i=0}^{n-1} \beta_i$, $m(\zeta) = \max\{n : \zeta_n \leq \zeta\}$. Under the time scale β , define the translation process similarly as before $\lambda^n(\cdot)$ and $Z^n(\cdot)$. Let $Z^n(\zeta) = \sum_{i=n}^{m(\zeta_n+\zeta)-1} \beta_i Z_i$ for $\zeta \geq 0$. Assume that for given $T_0 > 0$, $Z^n(\zeta)$ is not equicontinuous on $[0, T_0]$, then there exists a sequence $n_k \rightarrow \infty$, which is dependent on pathway, bounded time $\xi_k \in [0, T]$, $v_k \rightarrow 0^+$, $\epsilon > 0$, such that $\|Z^{n_k}(\xi_k + v_k) - Z^{n_k}(\xi_k)\| = \left\| \sum_{i=m(\zeta_{n_k}+\xi_k)}^{m(\zeta_{n_k}+\xi_k+v_k)} \beta_i Z_i \right\| \geq \epsilon$. By the conclusion in Lemma 7 $\|Z_k\| \leq \|S_k\|$, we have $\|Z_k\| \leq \|S_k\| \leq \|\nabla_{\lambda} \hat{L}_M(\lambda)|_{\lambda=\lambda_k}\| + \|\nabla_{\lambda} \hat{L}_M(\lambda)|_{\lambda=\lambda_k} + S_k\| = \|h(\lambda_k)\| + \|b_k\|$. Furthermore,

$$\epsilon \leq \left\| \sum_{i=m(\zeta_{n_k}+\xi_k)}^{m(\zeta_{n_k}+\xi_k+v_k)} \beta_i Z_i \right\| \leq \left\| \sum_{i=m(\zeta_{n_k}+\xi_k)}^{m(\zeta_{n_k}+\xi_k+v_k)} \beta_i (\|h(\lambda_i)\| + \|b_i\|) \right\|. \quad (\text{B8})$$

Since $h(\lambda) = \nabla_{\lambda} \hat{L}_M(\lambda)$ is continuous, it is bounded in Λ . By Proposition 5, we have $\|b_k\| \rightarrow 0$ and $\beta_k \rightarrow 0$ when $k \rightarrow \infty$. Therefore, the left-hand side of Equation (B8) is a constant, while the right end tends to 0, leading to a contradiction with the assumption that $Z^n(t)$ is not equicontinuous. Hence, $Z^n(t)$ is equicontinuous. Moreover, $\lambda^n(t)$ is also equicontinuous on $[0, T_0]$. By applying Theorem 5.2.3 in Harold et al. (1997), we can verify that all conditions

are satisfied, and the convergent subsequence of $(\lambda^n(\cdot), Z^n(\cdot))$ satisfies the ODE. Thus, the iterative sequence $\{\lambda_k\}$ converges to the limit point. Consequently, the value $\bar{\lambda}^M$ obtained in Theorem 6 is the equilibrium of the ODE, which satisfies $\nabla_{\lambda} \hat{L}_M(\lambda)|_{\lambda=\bar{\lambda}^M} = 0$. Therefore, the limit of $\{\lambda_k\}$ is precisely the optimal value of the approximate ELBO. $\square$

Proof of Proposition 7 We have $\|S_k^M - \nabla_{\lambda} L(\lambda_k)\| \leq \|S_k^M - \nabla_{\lambda} \hat{L}_M(\lambda_k)\| + \|\nabla_{\lambda} \hat{L}_M(\lambda_k) - \nabla_{\lambda} L(\lambda_k)\|$. Let $k \rightarrow \infty$ first, Proposition 5 shows the first term tends to 0. Then let $M \rightarrow \infty$, Proposition 2 shows the uniform convergence with respect to k as $M \rightarrow \infty$:

$$\sup_k \|\nabla_{\lambda} \hat{L}_M(\lambda_k) - \nabla_{\lambda} L(\lambda_k)\| \xrightarrow{a.s.} 0.$$

For $\epsilon > 0$, there exists $M_0 > 0$, for every $M \geq M_0$, there exists K_M , $\|S_k^M - \nabla_{\lambda} \hat{L}_M(\lambda_k)\| < \epsilon/2$ holds for every $k \geq K_M$. Also, $\|\nabla_{\lambda} \hat{L}_M(\lambda_k) - \nabla_{\lambda} L(\lambda_k)\| < \epsilon/2$ holds for every k when $M \geq M_0$. Therefore, for $\epsilon > 0$, there exists $M_0 > 0$, for every $M \geq M_0$, there exists K_M , when $K > K_M$, $\|S_k^M - \nabla_{\lambda} L(\lambda_k)\| < \epsilon$, which ends the proof. $\square$

Proof of Proposition 8 By Proposition 2, there exists an event Ω_0 with $\mathbb{P}(\Omega_0) = 1$ such that, on Ω_0 , $\sup_{\lambda \in \Lambda} \|\nabla_{\lambda} \hat{L}_M(\lambda) - \nabla_{\lambda} L(\lambda)\| \rightarrow 0$ as $M \rightarrow \infty$. Fix any sample path in Ω_0 .

Suppose, to the contrary, that $\bar{\lambda}^M$ does not converge to $\bar{\lambda}$. Then there exist $\epsilon_0 > 0$ and a subsequence $\{\bar{\lambda}^{M_i}\}$ such that $\|\bar{\lambda}^{M_i} - \bar{\lambda}\| > \epsilon_0$ for all i . Since Λ is compact, there exists a further subsequence, still denoted by $\{\bar{\lambda}^{M_i}\}$, and some $\tilde{\lambda} \in \Lambda$ such that $\bar{\lambda}^{M_i} \rightarrow \tilde{\lambda}$.

By definition, $\nabla_{\lambda} \hat{L}_{M_i}(\bar{\lambda}^{M_i}) = 0$. Hence,

$$\begin{aligned} \|\nabla_{\lambda} L(\tilde{\lambda})\| &\leq \|\nabla_{\lambda} L(\tilde{\lambda}) - \nabla_{\lambda} L(\bar{\lambda}^{M_i})\| \\ &\quad + \|\nabla_{\lambda} L(\bar{\lambda}^{M_i}) - \nabla_{\lambda} \hat{L}_{M_i}(\bar{\lambda}^{M_i})\| + \|\nabla_{\lambda} \hat{L}_{M_i}(\bar{\lambda}^{M_i})\|. \end{aligned}$$

The first term converges to zero by the continuity of $\nabla_{\lambda} L$, the second term is bounded by $\sup_{\lambda \in \Lambda} \|\nabla_{\lambda} L(\lambda) - \nabla_{\lambda} \hat{L}_{M_i}(\lambda)\|$ and hence converges to zero by Proposition 2, and the third term is zero. Therefore, $\nabla_{\lambda} L(\tilde{\lambda}) = 0$.

By Assumption 3.8, the equation $\nabla_{\lambda} L(\lambda) = 0$ has a unique solution in Λ , denoted by $\bar{\lambda}$. Thus $\tilde{\lambda} = \bar{\lambda}$. This contradicts $\|\bar{\lambda}^{M_i} - \bar{\lambda}\| > \epsilon_0$ for all i , because $\bar{\lambda}^{M_i} \rightarrow \tilde{\lambda} = \bar{\lambda}$. Therefore, $\bar{\lambda}^M \rightarrow \bar{\lambda}$ on Ω_0 . Since $\mathbb{P}(\Omega_0) = 1$, the convergence holds w.p.1. $\square$

Proof of Weak Convergence

Proof of Proposition 10 Since $\lambda \in \Lambda$, we can omit the projection term Z_k in recursion (9). The convergence of (λ_k, D_k) to $(\bar{\lambda}^M, \bar{D})$ has been proved. Let $f(\lambda, D) = \frac{A(\lambda)}{M}(E^M D + B(\lambda) - C(\lambda))$

, $g(\lambda, D) = \nabla_{\theta} p(y|\theta(\lambda)) - p(y|\theta(\lambda))D$. Applying the Taylor expansion at the limit point $(\bar{\lambda}^M, \bar{D})$, we have

$$\begin{pmatrix} f(\lambda, D) \\ g(\lambda, D) \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \cdot \begin{pmatrix} \lambda - \bar{\lambda}^M \\ D - \bar{D} \end{pmatrix} + O\left(\left\| \begin{pmatrix} \lambda - \bar{\lambda}^M \\ D - \bar{D} \end{pmatrix} \right\|^2\right), \quad (C9)$$

where $Q_{11} = \frac{\partial f(\lambda, D)}{\partial \lambda}|_{(\bar{\lambda}^M, \bar{D})}$, $Q_{12} = \frac{\partial f(\lambda, D)}{\partial D}|_{(\bar{\lambda}^M, \bar{D})} = \frac{A(\bar{\lambda}^M)E^M}{M}$, $Q_{21} = \frac{\partial g(\lambda, D)}{\partial \lambda}|_{(\bar{\lambda}^M, \bar{D})}$, $Q_{22} = \frac{\partial g(\lambda, D)}{\partial D}|_{(\bar{\lambda}^M, \bar{D})} = -p(y|\theta(\bar{\lambda}^M))$. By the optimal condition for limit point $f(\bar{\lambda}^M, \bar{D}) = g(\bar{\lambda}^M, \bar{D}) = 0$, we have $Q_{11} = \nabla_{\lambda}^2 \hat{L}_M(\bar{\lambda}^M)$, $Q_{21} = 0$. In the framework of the NMTS algorithm,

$$\begin{cases} \lambda_{k+1} = \lambda_k + \beta_k A_k \\ D_{k+1} = D_k + \alpha_k B_k, \end{cases}$$

where $A_k = f(\lambda_k, D_k)$, $B_k = G_{1,k}(\lambda_k) - G_{2,k}(\lambda_k)D_k = g(\lambda_k, D_k) + W_k$. Here $W_k = G_{1,k}(\lambda_k) - \nabla_{\theta} p(y|\theta_k) + p(y|\theta_k)D_k - G_{2,k}(\lambda_k)D_k$, and $\mathbb{E}[W_k|\mathcal{F}_k] = 0$ by Assumption 3.4.

Set $H = Q_{11} - Q_{12}Q_{22}^{-1}Q_{21} = Q_{11} = \nabla_{\lambda}^2 \hat{L}_M(\bar{\lambda}^M)$, then the largest eigenvalue of H is negative by Assumption 9.1. Also, the largest eigenvalue of Q_{22} is negative by its definition.

Define the following equations: $\Gamma_{22} = \lim_{k \rightarrow \infty} \mathbb{E}[W_k W_k^T | \mathcal{F}_k]$, $\Gamma_{\theta} = Q_{12}Q_{22}^{-1}\Gamma_{22}Q_{22}^{-\top}Q_{12}^{\top}$,

$$\Sigma_{\lambda} = \int_0^{\infty} \exp(Ht)\Gamma_{\theta} \exp(H^{\top}t)dt, \quad \Sigma_D = \int_0^{\infty} \exp(Q_{22}t)\Gamma_{22} \exp(Q_{22}t)dt. \quad (C10)$$

Therefore, we will reach the conclusion by checking all the conditions and applying Theorem 1 in Mokkadem and Pelletier (2006). $\square$

Proof of Theorem 11 By the definition of S_k and $\nabla_{\lambda} \hat{L}_M(\bar{\lambda}^M)$,

$$\begin{aligned} S_k - \nabla_{\lambda} \hat{L}_M(\bar{\lambda}^M) &= \frac{A(\lambda_k)}{M} \left(E^M D_k + B(\lambda_k) - C(\lambda_k) \right) - \frac{A(\bar{\lambda}^M)}{M} \left(E^M \bar{D} + B(\bar{\lambda}^M) - C(\bar{\lambda}^M) \right) \\ &= \frac{1}{M} \left(A(\lambda_k)E^M D_k - A(\bar{\lambda}^M)E^M \bar{D} + A(\lambda_k)B(\lambda_k) - A(\bar{\lambda}^M)B(\bar{\lambda}^M) - A(\lambda_k)C(\lambda_k) + A(\bar{\lambda}^M)C(\bar{\lambda}^M) \right), \end{aligned}$$

where the first two terms satisfy

$$\begin{aligned} \sqrt{\alpha_k^{-1}}(A(\lambda_k)E^M D_k - A(\bar{\lambda}^M)E^M \bar{D}) &= \sqrt{\alpha_k^{-1}}(A(\lambda_k)E^M D_k - A(\bar{\lambda}^M)E^M D_k + A(\bar{\lambda}^M)E^M D_k - A(\bar{\lambda}^M)E^M \bar{D}) \\ &= \sqrt{\frac{\beta_k}{\alpha_k}} \sqrt{\beta_k^{-1}}(A(\lambda_k) - A(\bar{\lambda}^M))E^M D_k + \sqrt{\alpha_k^{-1}}A(\bar{\lambda}^M)E^M(D_k - \bar{D}). \end{aligned}$$

By the Delta method (Vaart, 1998) and Proposition 10, we have

$$\begin{aligned} \sqrt{\beta_k^{-1}}((A(\lambda_k) - A(\bar{\lambda}^M)) \xrightarrow{d} A'(\bar{\lambda}^M)\mathcal{N}(0, \Sigma_{\lambda}), \\ \sqrt{\alpha_k^{-1}}A(\bar{\lambda}^M)E^M(D_k - \bar{D}) \xrightarrow{d} \mathcal{N}(0, A(\bar{\lambda}^M)E^M \Sigma_D (E^M)^{\top} A(\bar{\lambda}^M)^{\top}). \end{aligned}$$

Note that $\frac{\beta_k}{\alpha_k} \rightarrow 0$ and by Slutsky's Theorem,

$$\sqrt{\frac{\beta_k}{\alpha_k}} \sqrt{\beta_k^{-1}(A(\lambda_k) - A(\bar{\lambda}^M))E^M D_k} = \sqrt{\frac{\beta_k}{\alpha_k}} E \bar{D} O_p(1) \xrightarrow{d} 0.$$

The same weak convergence rate is also true for the convergence of $A(\lambda_k)B(\lambda_k)$ and $A(\lambda_k)C(\lambda_k)$:

$$\sqrt{\alpha_k^{-1}(A(\lambda_k)B(\lambda_k) - A(\bar{\lambda}^M)B(\bar{\lambda}^M))} = \sqrt{\frac{\beta_k}{\alpha_k}} O_p(1) = o_p(1), \sqrt{\alpha_k^{-1}(A(\lambda_k)C(\lambda_k) - A(\bar{\lambda}^M)C(\bar{\lambda}^M))} = o_p(1).$$

Combining all these terms, by Slutsky's Theorem, we will have

$$\sqrt{\alpha_k^{-1}(A(\lambda_k)E^M D_k - A(\bar{\lambda}^M)E \bar{D})} \xrightarrow{d} \mathcal{N}(0, A(\bar{\lambda}^M)E^M \Sigma_D E^\top A(\bar{\lambda}^M)^\top).$$

In conclusion, $\sqrt{\alpha_k^{-1}(S_k - \nabla_{\lambda} \hat{L}_M(\bar{\lambda}^M))} = \sqrt{\alpha_k^{-1} \frac{1}{M} A(\bar{\lambda}^M)E(D_k - \bar{D})} + o_p(1) \xrightarrow{d} \mathcal{N}(0, \Sigma_s^M)$. $\square$

Proof of Lemma 1 We can analyze the order with respect to M and N for every part. Define $O(\cdot)$ as the order of elements in a matrix. Q_{11} is a square matrix with l dimensions and all the elements in Q_{11} are constant order since $Q_{11} = \nabla_{\lambda}^2 \hat{L}_M(\bar{\lambda}^M)$. Q_{12} is a matrix with l rows and $M \times d \times T$ columns and the order of element is $O(\frac{1}{M})$ by the form of Q_{12} and the boundness of $A(\lambda)$. So H is a square matrix with l dimensions and $O(H) = O(1)$. Q_{22} is a diagonal matrix with $M \times d \times T$ dimensions and for every element $O(Q_{22}) = O(1)$. Furthermore, by the variance of Monte Carlo simulation in Equation (3), $\Gamma_{22} = \lim_{k \rightarrow \infty} \mathbb{E}[W_k W_k^\top | \mathcal{F}_k] = O(\frac{1}{N})$. Then $O(\Gamma_{\theta})$ is a square matrix with l dimensions and $O(\Gamma_{\theta}) = O(Q_{12} Q_{22}^{-1} \Gamma_{22} Q_{22}^{-\top} Q_{12}^\top) = O(\frac{1}{N})$. Therefore, Σ_{λ} is a matrix with l dimensions and $O(\Sigma_{\lambda}) = O(\frac{1}{N})$.

Γ_{22} is a square matrix with $M \times d \times T$ dimensions and $O(\Gamma_{22}) = O(\frac{1}{N})$. Therefore, Σ_D is also a square matrix with $M \times d \times T$ dimensions and its every element satisfies $O(\Sigma_D) = O(\frac{1}{N})$. $\square$

Proof of Theorem 12 We can use the same method as Theorem 11 to check that $\text{Var}(S_k^M - \nabla_{\lambda} \hat{L}_M(\lambda_k)) = O(\frac{\alpha_k}{N})$. We have

$$\begin{aligned} S_k^M - \nabla_{\lambda} \hat{L}_M(\lambda_k) &= \frac{A(\lambda_k)}{M} (E^M D_k + B(\lambda_k) - C(\lambda_k)) - \frac{A(\lambda_k)}{M} (\nabla_{\theta} \log p(y|\theta(u; \lambda_k)) + B(\lambda_k) - C(\lambda_k)) \\ &= \frac{1}{M} (A(\lambda_k)E^M D_k - A(\lambda_k)E^M \nabla_{\theta} \log p(y|\theta(u; \lambda_k))) \\ &= \frac{1}{M} A(\lambda_k)E^M (D_k - \bar{D}) + \frac{1}{M} A(\lambda_k)E^M (\nabla_{\theta} \log p(y|\theta(u; \bar{\lambda}^M)) - \nabla_{\theta} \log p(y|\theta(u; \lambda_k))). \end{aligned}$$

Therefore, by Slutsky's Theorem and the Delta method, the asymptotic variance of the first term and the second term are

$$\begin{aligned}\text{Var}\left(\frac{1}{M}A(\lambda_k)E^M(D_k - \bar{D})\right) &= O(\alpha_k)O\left(\frac{1}{M^2}A(\bar{\lambda}^M)E^M\Sigma_D(E^M)^\top A(\bar{\lambda}^M)^\top\right) = O\left(\frac{\alpha_k}{N}\right), \\ \text{Var}\left(\frac{1}{M}A(\lambda_k)E^M\left(\nabla_\theta \log p(y|\theta(u; \bar{\lambda}^M)) - \nabla_\theta \log p(y|\theta(u; \lambda_k))\right)\right) &= O(\beta_k)O(\Sigma_D) = O\left(\frac{\beta_k}{N}\right).\end{aligned}$$

Proposition 2 shows that $\text{Var}(\nabla_{\lambda} \hat{L}_M(\lambda_k) - \nabla_{\lambda} L(\lambda_k)) = O(\frac{1}{M})$ uniformly for every λ_k . Then we have

$$\begin{aligned}\text{Var}(S_k^M - \nabla_{\lambda} L(\lambda_k)) &= \text{Var}(S_k^M - \nabla_{\lambda} \hat{L}_M(\lambda_k)) + \text{Var}(\nabla_{\lambda} \hat{L}_M(\lambda_k) - \nabla_{\lambda} L(\lambda_k)) \\ &\quad + 2 \text{Cov}(S_k^M - \nabla_{\lambda} \hat{L}_M(\lambda_k), \nabla_{\lambda} \hat{L}_M(\lambda_k) - \nabla_{\lambda} L(\lambda_k)) \\ &\leq 2 \text{Var}(S_k^M - \nabla_{\lambda} \hat{L}_M(\lambda_k)) + 2 \text{Var}(\nabla_{\lambda} \hat{L}_M(\lambda_k) - \nabla_{\lambda} L(\lambda_k)) = O\left(\frac{\alpha_k}{N}\right) + O\left(\frac{1}{M}\right).\end{aligned}$$

By using Chebyshev's inequality, we can reach the conclusion. $\square$

Proof of Theorem 13 By the Taylor expansion, $\nabla_{\lambda} \hat{L}_M(\bar{\lambda}^M) - \nabla_{\lambda} \hat{L}_M(\bar{\lambda}) = \nabla^2 \hat{L}_M(\bar{\lambda})(\bar{\lambda}^M - \bar{\lambda}) + o(\bar{\lambda}^M - \bar{\lambda})$. And notice that $\nabla_{\lambda} L(\bar{\lambda}) = \nabla_{\lambda} \hat{L}_M(\bar{\lambda}^M) = 0$, by Assumption 9.1, we have $\bar{\lambda}^M - \bar{\lambda} = \nabla^2 \hat{L}_M(\bar{\lambda})^{-1}(\nabla_{\lambda} L(\bar{\lambda}) - \nabla_{\lambda} \hat{L}_M(\bar{\lambda})) + o(\bar{\lambda}^M - \bar{\lambda})$. By Slutsky's Theorem and the asymptotic normality of $\nabla_{\lambda} \hat{L}_M(\bar{\lambda})$, we have

$$\sqrt{M}(\bar{\lambda}^M - \bar{\lambda}) \xrightarrow{d} \mathcal{N}(0, \nabla^2 L(\bar{\lambda})^{-1} \text{Var}_u(h(u; \bar{\lambda})) \nabla^2 L(\bar{\lambda})^{-\top}).$$

$\square$

Proof of Theorem 14 Proposition 10 and Lemma 1 show that $\text{Var}(\lambda_k^M - \bar{\lambda}^M) = O(\beta_k \Sigma_{\lambda}) = O(\frac{\beta_k}{N})$. Theorem 13 shows that $\text{Var}(\bar{\lambda}^M - \bar{\lambda}) = O(\frac{1}{M})$. Therefore, we have

$$\begin{aligned}\text{Var}(\lambda_k^M - \bar{\lambda}) &= \text{Var}(\lambda_k^M - \bar{\lambda}^M) + \text{Var}(\bar{\lambda}^M - \bar{\lambda}) + 2 \text{Cov}(\lambda_k^M - \bar{\lambda}^M, \bar{\lambda}^M - \bar{\lambda}) \\ &\leq 2 \text{Var}(\lambda_k^M - \bar{\lambda}^M) + 2 \text{Var}(\bar{\lambda}^M - \bar{\lambda}) = O\left(\frac{\beta_k}{N}\right) + O\left(\frac{1}{M}\right).\end{aligned}$$

By using Chebyshev's inequality, we can reach the conclusion. $\square$

Proof of $\mathbb{L}^1$ Convergence

Proof of Theorem 15 Let $h(\lambda) = p(y|\theta(u; \lambda))^{-1} \nabla_{\theta} p(y|\theta(u; \lambda))$, $\zeta_k = D_k - h(\lambda_k)$, we have

$$\zeta_{k+1} = \zeta_k + \alpha_k (G_1(\lambda_k) - G_2(\lambda_k) D_k) + h(\lambda_k) - h(\lambda_{k+1}).$$

Since p is twice continuously differentiable and Λ is compact, h is Lipschitz continuous on Λ and denote its Lipschitz constant as L_h , then we have

$$\|h(\lambda_k) - h(\lambda_{k+1})\| \leq L_h \|\lambda_k - \lambda_{k+1}\| = L_h \|\beta_k \left(\frac{A(\lambda_k)}{M} (E^M D_k + B(\lambda_k) - C(\lambda_k)) + Z_k \right)\| \leq 2L_h \beta_k C_D,$$

where C_D is the bound of $\frac{A(\lambda_k)}{M} (E^M D_k + B(\lambda_k) - C(\lambda_k))$ by Lemma 3 and the boundness of continuous function $A(\lambda)$, $B(\lambda)$ and $C(\lambda)$. Then we have

$$\begin{aligned} \|\zeta_{k+1}\|^2 &\leq \|\zeta_k\|^2 + \alpha_k^2 \|G_1(\lambda_k) - G_2(\lambda_k) D_k\|^2 + 4L_h^2 \beta_k^2 C_D^2 + 4\|\zeta_k\| L_h \beta_k C_D + \\ &2\alpha_k \zeta_k^\top (G_1(\lambda_k) - G_2(\lambda_k) D_k) + 2\alpha_k (h(\lambda_k) - h(\lambda_{k+1}))^\top (G_1(\lambda_k) - G_2(\lambda_k) D_k). \end{aligned}$$

By the form of G_1 and G_2 in Equation (3), we have $\mathbb{E}[\|G_1(\lambda_k) - \nabla_\theta p(y|\theta(u; \lambda_k))\|^2 | \mathcal{F}_k] = O(\frac{1}{N})$, $\mathbb{E}[\|G_2(\lambda_k) - p(y|\theta(u; \lambda_k))\|^2 | \mathcal{F}_k] = O(\frac{1}{N})$. Set $W_k = G_1(\lambda_k) - G_2(\lambda_k) D_k + p(\lambda_k) \zeta_k$ and it follows that $\mathbb{E}[W_k | \mathcal{F}_k] = 0$, $\mathbb{E}[\|W_k\|^2 | \mathcal{F}_k] = O(\frac{1}{N})$. Take the conditional expectation on both sides, and we can yield

$$\begin{aligned} \mathbb{E}[\|\zeta_{k+1}\|^2 | \mathcal{F}_k] &\leq \|\zeta_k\|^2 + \alpha_k^2 \mathbb{E}[\|W_k - p(\lambda_k) \zeta_k\|^2 | \mathcal{F}_k] + 4L_h^2 \beta_k^2 C_D^2 + 4\|\zeta_k\| L_h \beta_k C_D - 2\alpha_k \zeta_k^\top p(\lambda_k) \zeta_k \\ &\quad + 2\alpha_k (h(\lambda_k) - h(\lambda_{k+1}))^\top (-p(\lambda_k) \zeta_k) \\ &\leq \|\zeta_k\|^2 + 2\alpha_k^2 \left(\frac{C_G}{N} + C_p^+ \|\zeta_k\|^2 \right) + 4L_h^2 \beta_k^2 C_D^2 + 4\|\zeta_k\| L_h \beta_k C_D - 2\alpha_k C_p^- \|\zeta_k\|^2 \\ &\quad + 2\alpha_k * 2L_h \beta_k C_D \|\zeta_k\| C_p^+, \end{aligned}$$

where C_p^- and C_p^+ are the bounds of $\|p(\theta(u; \lambda))\|$ in Λ and C_G is the bound for the variance term in the Monte Carlo simulation. Taking the expectation again, when k is large enough, we have

$$\begin{aligned} \mathbb{E}[\|\zeta_{k+1}\|^2] &\leq (1 - 2\alpha_k C_p^- + 2\alpha_k^2 C_p^+) \mathbb{E}[\|\zeta_k\|^2] + 4L_h \beta_k C_D (1 + \alpha_k C_p^+) \mathbb{E}[\|\zeta_k\|] + 4L_h^2 \beta_k^2 C_D^2 + 2\alpha_k^2 \frac{C_G}{N} \\ &\leq (1 - \alpha_k C_p^-) \mathbb{E}[\|\zeta_k\|^2] + 4L_h \beta_k C_D (1 + \alpha_k C_p^+) \sqrt{\mathbb{E}[\|\zeta_k\|^2]} + 4L_h^2 \beta_k^2 C_D^2 + 2\alpha_k^2 \frac{C_G}{N} \\ &\leq \left(\sqrt{1 - \alpha_k C_p^-} \sqrt{\mathbb{E}[\|\zeta_k\|^2]} + \frac{2\beta_k L_h C_D (1 + \alpha_k C_p^+)}{\sqrt{1 - \alpha_k C_p^-}} \right)^2 + 2\alpha_k^2 \frac{C_G}{N} \\ &\leq \left((1 - \frac{1}{2} \alpha_k C_p^-) \sqrt{\mathbb{E}[\|\zeta_k\|^2]} + C_3 \beta_k \right)^2 + \frac{\alpha_k^2}{N} C_4. \end{aligned}$$

Now, define the mapping $T_k(x) := \sqrt{\left((1 - \frac{1}{2} \alpha_k C_p^-) x + C_3 \beta_k \right)^2 + \frac{\alpha_k^2}{N} C_4}$, and consider the sequence of $\{x_k\}$ generated by $x_{k+1} = T_k(x_k)$ for all k with $x_0 := \sqrt{\mathbb{E}[\|\zeta_0\|^2]}$. A simple induction shows that $\sqrt{\mathbb{E}[\|\zeta_k\|^2]} \leq x_k$. In addition, it is obvious that the gradient of $T_k(x)$ is less than 1, which implies that T_k is a contraction mapping. The unique fixed point is the form of

$$\bar{x} = O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right) + \text{higher order terms}.$$

Then applying the same technique in Jiang et al. (2023), we can conclude that $\mathbb{E}[\|\zeta_k\|]$ has the same order. $\square$

Proof of Theorem 16 Define $\psi_k = \lambda_k - \bar{\lambda}^M$, and $\eta_k = S_k - \nabla_{\lambda} \hat{L}_M(\lambda_k)$. Then

$$\psi_{k+1} = \psi_k + \beta_k(S_k + Z_k) = \psi_k + \beta_k\eta_k + \beta_k\nabla_{\lambda} \hat{L}_M(\lambda_k) + \beta_k Z_k.$$

Apply the Taylor expansion of $\nabla_{\lambda} \hat{L}_M(\lambda_k)$ around $\bar{\lambda}^M$, it follows that

$$\nabla_{\lambda} \hat{L}_M(\lambda_k) = \nabla_{\lambda}^2 \hat{L}_M(\bar{\lambda})(\lambda_k - \bar{\lambda}^M) = H(\bar{\lambda})\psi_k.$$

We have $\psi_{k+1} = \psi_k + \beta_k(S_k + Z_k) = (I + \beta_k H(\bar{\lambda}))\psi_k + \beta_k\eta_k + \beta_k Z_k$. By applying Rayleigh-Ritz inequality (Rugh, 1996) and Assumption 9.1, we can get

$$\|\psi_{k+1}\| \leq \|I + \beta_k H(\bar{\lambda})\| \|\psi_k\| + \beta_k \|\eta_k\| + \beta_k \|Z_k\| \leq (1 - \beta_k K_L) \|\psi_k\| + \beta_k \|\eta_k\| + \beta_k \|Z_k\|. \quad (\text{D11})$$

We now derive a bound for $\mathbb{E}[\|Z_k\|]$. Since $\bar{\lambda}^M$ is in the interior of Λ , there is a constant $\epsilon_{\lambda} > 0$ such that the $2\epsilon_{\lambda}$ -neighborhood of $\bar{\lambda}^M$ is contained in Λ . Let $A_k = \{\|\lambda_{k+1} - \bar{\lambda}^M\| \geq 2\epsilon_{\lambda}\}$. We have

$$\begin{aligned} \mathbb{E}[\|Z_k\|] &= \mathbb{E}[\|Z_k\| | A_k] P(A_k) + \mathbb{E}[\|Z_k\| | A_k^c] P(A_k^c) \leq \mathbb{E}[\|S_k\|] P(\|\lambda_{k+1} - \bar{\lambda}^M\| \geq 2\epsilon_{\lambda}) \\ &\leq \mathbb{E}[\|S_k\|] P(\|\lambda_{k+1} - \lambda_k\| \geq \epsilon_{\lambda} \cup \|\bar{\lambda}^M - \lambda_k\| \geq \epsilon_{\lambda}) \\ &\leq \mathbb{E}[\|S_k\|] \left(\frac{\mathbb{E}[\|\lambda_{k+1} - \lambda_k\|]}{\epsilon_{\lambda}} + \frac{\mathbb{E}[\|\bar{\lambda}^M - \lambda_k\|]}{\epsilon_{\lambda}} \right) \leq \frac{2\beta_k \mathbb{E}^2[\|S_k\|]}{\epsilon_{\lambda}} + \mathbb{E}[\|S_k\|] \frac{\mathbb{E}[\|\psi_k\|]}{\epsilon_{\lambda}}, \end{aligned}$$

where the last step follows from $\|\lambda_{k+1} - \lambda_k\| \leq \beta_k \|Z_k + S_k\| \leq 2\beta_k \|S_k\|$.

Then we take expectation in Equation (D11) and substitute the bound to get

$$\begin{aligned} \mathbb{E}[\|\psi_{k+1}\|] &\leq (1 - \beta_k K_L) \mathbb{E}[\|\psi_k\|] + \beta_k \mathbb{E}[\|\eta_k\|] + \beta_k \mathbb{E}[\|Z_k\|] \\ &\leq \left(1 - \beta_k (K_L - \frac{\mathbb{E}[\|S_k\|]}{\epsilon_{\lambda}}) \right) \mathbb{E}[\|\psi_k\|] + \beta_k \mathbb{E}[\|\eta_k\|] + \frac{2\beta_k^2 \mathbb{E}^2[\|S_k\|]}{\epsilon_{\lambda}}. \end{aligned}$$

By Proposition 5, $S_k - \nabla_{\lambda} \hat{L}_M(\lambda_k) \xrightarrow{a.s.} 0$ as k goes to infinity. Note that since $\lambda_k \rightarrow \bar{\lambda}^M$ w.p.1 and $\nabla_{\lambda} \hat{L}_M(\bar{\lambda}^M) = 0$, the continuity of $\nabla_{\lambda} \hat{L}_M(\cdot)$ shows that $\nabla_{\lambda} \hat{L}_M(\lambda_k) \rightarrow 0$. By dominated convergence theorem, $\mathbb{E}[\|S_k\|] \leq \mathbb{E}[\|S_k - \nabla_{\lambda} \hat{L}_M(\lambda_k)\|] + \mathbb{E}[\|\nabla_{\lambda} \hat{L}_M(\lambda_k)\|] \rightarrow 0$, which implies there exists an integer $K_S > 0$ such that $\mathbb{E}[\|S_k\|] \leq \frac{K_L \epsilon_{\lambda}}{2}$ for all $k \geq K_S$. Therefore, we obtain that for all $k \geq K_S$,

$$\mathbb{E}[\|\psi_{k+1}\|] \leq (1 - \frac{\beta_k K_L}{2}) \mathbb{E}[\|\psi_k\|] + \beta_k \mathbb{E}[\|\eta_k\|] + \frac{2\beta_k^2 \mathbb{E}^2[\|S_k\|]}{\epsilon_{\lambda}}.$$

Successive use of this inequality yields

$$\mathbb{E}[\|\psi_k\|] \leq \prod_{i=K_L}^k (1 - \frac{\beta_i K_L}{2}) \mathbb{E}[\|\psi_{K_L}\|] + \sum_{i=K_L}^k \prod_{j=i+1}^k (1 - \frac{\beta_j K_L}{2}) \beta_i \mathbb{E}[\|\eta_i\|] + \sum_{i=K_L}^k \prod_{j=i+1}^k (1 - \frac{\beta_j K_L}{2}) \frac{2\beta_i^2 \mathbb{E}^2[\|S_i\|]}{\epsilon_{\lambda}}. \quad (\text{D12})$$

By Theorem 15 and definition of S_k ,

$$\mathbb{E}[\|\eta_k\|] = \mathbb{E}[\|S_k - \nabla_{\lambda} \hat{L}_M(\lambda_k)\|] = \mathbb{E}\left[\left\|\frac{A(\lambda_k)}{M} E^M(D_k - \nabla_{\theta} \log p(y|\theta(u; \lambda_k)))\right\|\right] = O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right).$$

Due to Lemma 2, $\mathbb{E}^2[\|S_k\|] = O(1)$. When $\alpha_k = \frac{A}{k^a}$ and $\beta_k = \frac{B}{k^b}$, we can apply Lemma 3 in Hu et al. (2024) to estimate the order of this summation based on the order of $\mathbb{E}[\|\eta_k\|]$:

$$\sum_{i=K_L}^k \prod_{j=i+1}^k \left(1 - \frac{\beta_j K_L}{2}\right) \beta_i \mathbb{E}[\|\eta_i\|] = O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right), \quad \sum_{i=K_L}^k \prod_{j=i+1}^k \left(1 - \frac{\beta_j K_L}{2}\right) \frac{2\beta_i^2 \mathbb{E}^2[\|S_i\|]}{\epsilon_{\lambda}} = O(\beta_k).$$

It is evident that $\prod_{i=K_L}^k \left(1 - \frac{\beta_i K_L}{2}\right) = e^{\sum_{i=K_L}^k \log(1 - \frac{\beta_i K_L}{2})} \leq e^{-\sum_{i=K_L}^k \frac{\beta_i K_L}{2}} \leq O\left(\frac{1}{k}\right)$. Combine the above inequalities and leave out the higher-order terms, and we can get the conclusion. $\square$

Proof of Proposition 17 Define $\psi_k = \lambda_k - \bar{\lambda}^M$, and $\eta_k = S'_k - \nabla_{\lambda} \hat{L}_M(\lambda_k)$, where S'_k is the corresponding definition in STS in Equation (11). Then

$$\psi_{k+1} = \psi_k + \beta_k(S'_k + Z_k) = \psi_k + \beta_k \eta_k + \beta_k \nabla_{\lambda} \hat{L}_M(\lambda_k) + \beta_k Z_k.$$

A same derivation of Theorem 16 leads us to the similar result as Equation (D12). Then we have the following results by applying Theorem 1 in Peng et al. (2017):

$$\begin{aligned} \mathbb{E}[\|\eta_k\|] &= \mathbb{E}[\|S'_k - \nabla_{\lambda} \hat{L}_M(\lambda_k)\|] \\ &= \mathbb{E}\left[\left\|\frac{A(\lambda_k)}{M} E^M\left(\frac{G_1(X, y, \theta_{k,m})}{G_2(X, y, \theta_{k,m})} - \nabla_{\theta} \log p(y|\theta(u; \lambda_k))\right)\right\|\right] = O\left(\sqrt{\frac{1}{N}}\right). \end{aligned}$$

Therefore, it follows that

$$\sum_{i=K_L}^k \prod_{j=i+1}^k \left(1 - \frac{\beta_j K_L}{2}\right) \beta_i \mathbb{E}[\|\eta_i\|] = O\left(\sqrt{\frac{1}{N}}\right), \quad \sum_{i=K_L}^k \prod_{j=i+1}^k \left(1 - \frac{\beta_j K_L}{2}\right) \frac{2\beta_i^2 \mathbb{E}^2[\|S'_i\|]}{\epsilon_{\lambda}} = O(\beta_k).$$

Finally, we can get the conclusion:

$$\mathbb{E}[\|\psi_k\|] = O\left(\sqrt{\frac{1}{N}}\right) + O(\beta_k). \quad \square$$

Proof of Theorem 18 By Lemma 3, we have a uniform bound for each block $\|D_{k,m}\|$, implying $\|D_{k,m}\| \leq C_D$ almost surely. Since $D_k = [D_{k,1}^{\top}, \dots, D_{k,M}^{\top}]^{\top}$, its Euclidean norm satisfies $\|D_k\| = O(\sqrt{M})$. Substituting this scaling into the recursive inequality for λ_k , we follow the same notation as Theorem 15.

Let $\zeta_k = \lambda_k - \bar{\lambda}^M$ denote the error at iteration k . We first bound the difference in $h(\lambda_k)$:

$$\|h(\lambda_k) - h(\lambda_{k+1})\| \leq L_h \|\lambda_{k+1} - \lambda_k\| \leq L_h \beta_k \left\| \frac{A(\lambda_k)}{M} (E^M D_k + B(\lambda_k) - C(\lambda_k)) + Z_k \right\| \leq 2L_h \beta_k C_D,$$

where C_D bounds $\frac{A(\lambda_k)}{M} (E^M D_k + B(\lambda_k) - C(\lambda_k))$ uniformly by continuity and Lemma 3.

Expanding $\|\zeta_{k+1}\|^2$ gives

$$\begin{aligned}\|\zeta_{k+1}\|^2 &\leq \|\zeta_k\|^2 + \alpha_k^2 \|G_1(\lambda_k) - G_2(\lambda_k)D_k\|^2 + 4L_h^2\beta_k^2 C_D^2 + 4L_h\beta_k C_D \|\zeta_k\| \\ &\quad + 2\alpha_k \zeta_k^\top (G_1(\lambda_k) - G_2(\lambda_k)D_k) + 2\alpha_k (h(\lambda_k) - h(\lambda_{k+1}))^\top (G_1(\lambda_k) - G_2(\lambda_k)D_k).\end{aligned}$$

Taking conditional expectation and applying Lemma 3, we obtain

$$\begin{aligned}\mathbb{E}[\|\zeta_{k+1}\|^2 | \mathcal{F}_k] &\leq \|\zeta_k\|^2 - 2\alpha_k \zeta_k^\top p(\lambda_k) \zeta_k + 4L_h\beta_k C_D \|\zeta_k\| + 4L_h^2\beta_k^2 C_D^2 \\ &\quad + 2\alpha_k \|h(\lambda_k) - h(\lambda_{k+1})\| \|p(\lambda_k) \zeta_k\| + \alpha_k^2 \mathbb{E}[\|G_1(\lambda_k) - G_2(\lambda_k)D_k + p(\lambda_k) \zeta_k\|^2 | \mathcal{F}_k].\end{aligned}$$

Using the Lipschitz bounds and the independence of the inner Monte Carlo samples, $\text{Var}(G_1 - G_2 D_k) = O(1/N)$, while outer blocks contribute additively as $O(1/M)$ due to the sample-average structure across $\{u_m\}_{m=1}^M$. Hence

$$\mathbb{E}[\|G_1(\lambda_k) - G_2(\lambda_k)D_k + p(\lambda_k) \zeta_k\|^2 | \mathcal{F}_k] \leq C_G \left(\frac{1}{N} + \|\zeta_k\|^2 \right).$$

Substituting the bounds and using $C_p^- I \leq p(\lambda_k) \leq C_p^+ I$, we obtain

$$\mathbb{E}[\|\zeta_{k+1}\|^2 | \mathcal{F}_k] \leq (1 - \alpha_k C_p^-) \|\zeta_k\|^2 + 4L_h\beta_k C_D (1 + \alpha_k C_p^+) \|\zeta_k\| + 2\alpha_k^2 \left(\frac{C_G}{N} + C_p^+ \|\zeta_k\|^2 \right) + 4L_h^2\beta_k^2 C_D^2.$$

Taking the total expectation yields

$$\mathbb{E}[\|\zeta_{k+1}\|^2] \leq (1 - \alpha_k C_p^-) \mathbb{E}[\|\zeta_k\|^2] + C_1 \beta_k \mathbb{E}[\|\zeta_k\|] + C_2 \alpha_k^2 \frac{1}{N} + C_3 \beta_k^2,$$

for some constants $C_1, C_2, C_3 > 0$.

Applying the same bounding technique as in Theorem 15, define

$$T_k(x) := \sqrt{\left(\left(1 - \frac{1}{2} \alpha_k C_p^- \right) x + C_1 \beta_k \right)^2 + \alpha_k^2 \frac{C_2}{N}}.$$

Its unique fixed point satisfies

$$\bar{x} = O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right).$$

Thus,

$$\mathbb{E}[\|\lambda_k^M - \bar{\lambda}^M\|] = O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right).$$

Finally, by Proposition 2, the outer SAA introduces an additional bias $O(M^{-1/2})$ due to finite outer samples. Combining all components, we obtain

$$\mathbb{E} \|\lambda_k^M - \tilde{\lambda}\| = O\left(\frac{\beta_k}{\alpha_k}\right) + O\left(\sqrt{\frac{\alpha_k}{N}}\right) + O\left(\frac{1}{\sqrt{M}}\right),$$

which completes the proof. $\square$

Proposition 19 is a direct corollary of the above two proofs, so we omit the proof.

Proof of Proposition 21 We derive the convergence rate of the second time scale by the shrinking bias of the first time scale implied by Assumption 20. Therefore, the proof is similar to Proposition 17. By Assumptions 3.1–3.3 and 20, we have

$$\mathbb{E} \left\| \frac{\nabla_{\theta} p_{\phi_k}(y|\theta)}{p_{\phi_k}(y|\theta)} - \nabla_{\theta} \log p(y|\theta) \right\| \leq \frac{\sqrt{C_1} + \sqrt{C_2}}{\epsilon^2} (O(\gamma_k^{(1)}) + O(\gamma_k^{(2)})) = O(\gamma_k^{(1)}) + O(\gamma_k^{(2)}).$$

The same derivation of Theorem 16 and Proposition 17 leads us to a similar result as Equation (D12). Here η_k is the bias of the first time scale. Then we have

$$\mathbb{E}[\|\eta_k\|] = \mathbb{E} \left[\left\| \frac{A(\lambda_k)}{M} E^M \left(\frac{\nabla_{\theta} p_{\phi_k}(y|\theta)}{p_{\phi_k}(y|\theta)} - \nabla_{\theta} \log p(y|\theta(u; \lambda_k)) \right) \right\| \right] = O(\gamma_k^{(1)}) + O(\gamma_k^{(2)}).$$

Therefore, it follows that

$$\sum_{i=K_L}^k \prod_{j=i+1}^k \left(1 - \frac{\beta_j K_L}{2}\right) \beta_i \mathbb{E}[\|\eta_i\|] = O(\gamma_k^{(1)}) + O(\gamma_k^{(2)}), \quad \sum_{i=K_L}^k \prod_{j=i+1}^k \left(1 - \frac{\beta_j K_L}{2}\right) \frac{2\beta_i^2 \mathbb{E}^2[\|S'_i\|]}{\epsilon_{\lambda}} = O(\beta_k).$$

Finally, we can get the conclusion

$$\mathbb{E}[\|\psi_k\|] = O(\gamma_k^{(1)}) + O(\gamma_k^{(2)}) + O(\beta_k). \quad \square$$

Other Supplement Information

Supplement Information for GLR estimators

Under the problem setting in Sect. 3.1 and Equation (3), the GLR estimator for density is

$$G_2(x, Y, \theta) := \mathbb{I}\{g(x; \theta) \leq Y\} \psi(x; \theta),$$

where $\mathbb{I}$ is the indicator function and

$$\psi(x; \theta) := \left(\frac{\partial g(x; \theta)}{\partial x_1} \right)^{-1} \left(\frac{\partial \log f(x; \theta)}{\partial x_1} - \frac{\partial^2 g(x; \theta)}{\partial x_1^2} \left(\frac{\partial g(x; \theta)}{\partial x_1} \right)^{-1} \right).$$

The GLR estimator for the derivative of the density is

$$G_1(x, Y, \theta) := \mathbb{I}\{g(x; \theta) \leq Y\} \left(\frac{\partial \log f(x; \theta)}{\partial \theta} + \frac{\partial \psi(x; \theta)}{\partial \theta} - \left(\frac{\partial g(x; \theta)}{\partial x_1} \right)^{-1} \left[\frac{\partial^2 g(x; \theta)}{\partial \theta \partial x_1} + \left(\frac{\partial g(x; \theta)}{\partial x_1} \right) \left\{ \frac{\partial \psi(x; \theta)}{\partial x_1} + \psi(x_1; \theta) \left(\frac{\partial \log f(x; \theta)}{\partial x_1} - \frac{\partial^2 g(x; \theta)}{\partial x_1^2} \left(\frac{\partial g(x; \theta)}{\partial x_1} \right)^{-1} \right) \right\} \right] \right).$$

By Theorem 1 and Theorem 2 in Peng et al. (2020), we have $\mathbb{E}_X[G_1(X, Y, \theta)] = \nabla_\theta p(Y; \theta)$ and $\mathbb{E}_X[G_2(X, Y, \theta)] = p(Y; \theta)$ for every observation Y under some soft conditions.

Supplement Information for Sect. 6.1

To further demonstrate that the empirical advantage of NMTS is not restricted to the Gaussian latent-noise setting, we additionally consider a non-Gaussian MLE example under the same latent-variable structure. Specifically, let the observations be generated by

$$Y_t = g(X_t; \theta) = X_{1,t} + \theta X_{2,t},$$

where $X_{1,t} \sim N(0, 1)$ and $X_{2,t} \sim \text{Exp}(1)$ are independent. Again, Y_t is observable while $X_t = (X_{1,t}, X_{2,t})$ is latent, and the goal is to estimate θ from $\{Y_t\}_{t=1}^T$. In contrast to the Gaussian benchmark, the likelihood in this exponential-noise setting does not lead to the same simple closed-form MLE, so we use a numerically computed MLE as the reference target for accuracy assessment.

In this experiment, we set the true parameter to $\theta = 1$, the observed sample size to $T = 100$, the feasible region to $\Theta = [0.5, 2]$, and the initialization to $\theta_0 = 0.8$. For NMTS, we use the fast and slow step-sizes $\alpha_k = \frac{10}{k^{0.7}}$, $\beta_k = \frac{0.2}{k}$, while for STS we use $\beta_k = \frac{0.05}{k}$. The number of iterations is set to 5000, and all results are based on 10 independent experiments. As in our other comparisons, NMTS and STS are evaluated under matched experimental budgets: the two methods use the same observed data, the same initialization, the same batch size N , the same number of iterations, and the same number of independent repetitions.

The results confirm that the empirical advantage of NMTS persists in this non-Gaussian likelihood-free setting. Figure 7 shows the convergence behavior for batch size $N = 10^3$. Relative to the numerically computed MLE, NMTS exhibits substantially smaller bias and much tighter variability than STS; the convergence trajectory is also markedly more stable. Table 5 further shows that this advantage is robust across multiple batch sizes. For $N = 10^2, 10^3, 10^4$, the absolute bias of NMTS remains at the order of 10^{-3} , while the corresponding STS bias stays at the order of 10^{-2} to 10^{-1} . These results provide additional evidence that the gain of NMTS is not tied to controlled Gaussian examples only, but extends to broader likelihood-free inference settings with non-Gaussian latent noise.

To further strengthen the robustness evidence beyond the Gaussian and exponential latent-noise settings, we additionally tested a third non-Gaussian MLE example under the same latent-variable structure, now with $X_{2,t} \sim \text{Unif}(0, 1)$ and $X_{1,t} \sim N(0, 1)$ independent, so that

$$Y_t = g(X_t; \theta) = X_{1,t} + \theta X_{2,t}.$$

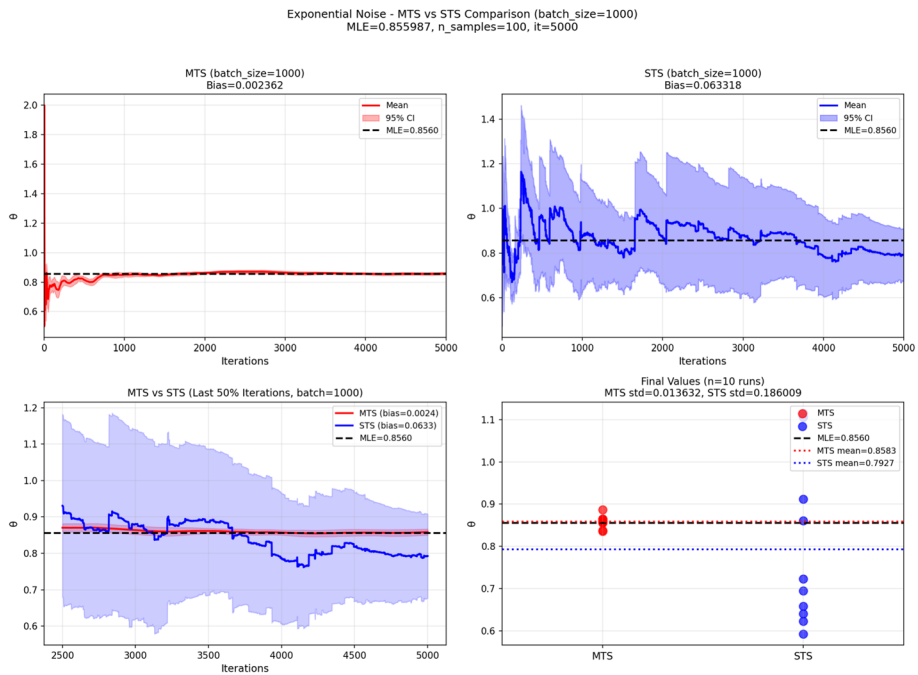


Fig. 7 The convergence plot of NMTS and STS methods with batchsize 10^3 , based on 10 independent experiments after 5000 iterations

Table 5 The MAE of the NMTS and STS methods, based on 10 independent experiments after 5000 iterations

Batch size	Absolute Bias $\pm$ std	
	NMTS	STS
10^2	$1.85 \times 10^{-03} \pm 3.76 \times 10^{-02}$	$9.73 \times 10^{-02} \pm 4.47 \times 10^{-01}$
10^3	$2.36 \times 10^{-03} \pm 1.36 \times 10^{-02}$	$6.33 \times 10^{-02} \pm 1.86 \times 10^{-01}$
10^4	$8.83 \times 10^{-04} \pm 1.99 \times 10^{-03}$	$8.64 \times 10^{-02} \pm 1.99 \times 10^{-01}$

Since the MLE is no longer available in simple closed form, we again use a numerically computed MLE as the reference target.

We keep the experimental protocol aligned with the previous non-Gaussian experiment: $\theta = 1$, $T = 100$, $\Theta = [0.5, 2]$, $\theta_0 = 0.8$, 5000 iterations, and 10 independent experiments. The NMTS step-sizes remain $\alpha_k = \frac{10}{k^{0.7}}$ and $\beta_k = \frac{0.2}{k}$, while the STS step-size is $\beta_k = \frac{0.05}{k}$. The two methods are again compared under matched experimental budgets, using the same observed data, initialization, batch size N , number of iterations, and number of independent repetitions.

The conclusions are fully consistent with the previous experiments. Figure 8 shows that, for batch size $N = 10^3$, NMTS remains much more stable than STS and stays much closer to the numerically computed MLE throughout the iterative process. In particular, the absolute bias of NMTS is 8.25×10^{-3} , compared with 1.39×10^{-1} for STS, and the final-value vari-

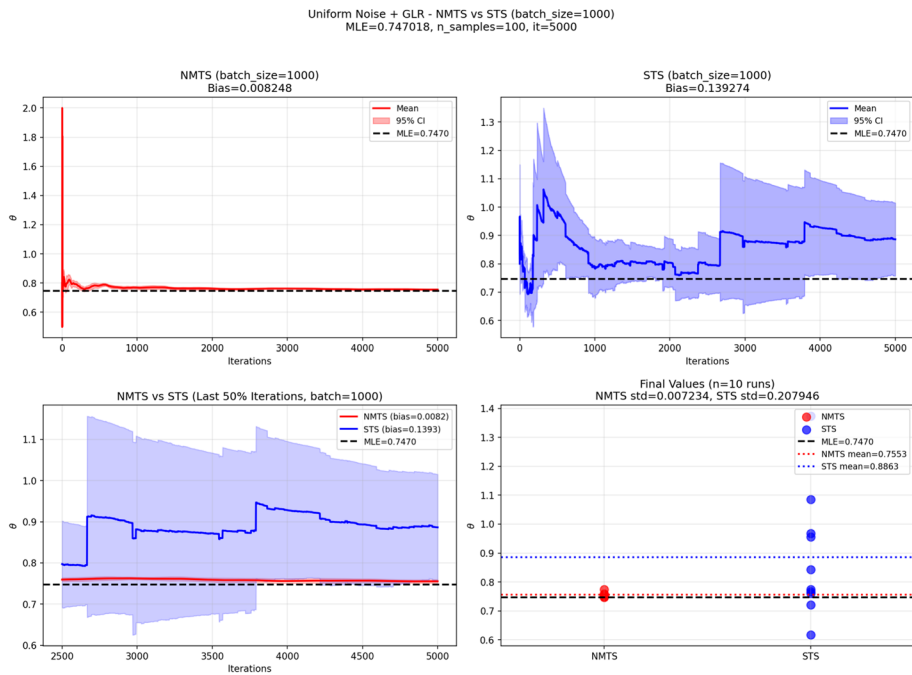


Fig. 8 The convergence plot of NMTS and STS methods in the uniform-noise setting, with batchsize 10^3 , based on 10 independent experiments after 5000 iterations

Table 6 The MAE of the NMTS and STS methods in the uniform-noise setting, based on 10 independent experiments after 5000 iterations

Batch size	Absolute Bias $\pm$ std	
	NMTS	STS
10^2	$1.43 \times 10^{-2} \pm 4.64 \times 10^{-2}$	$1.47 \times 10^{-1} \pm 3.98 \times 10^{-1}$
10^3	$8.25 \times 10^{-3} \pm 7.23 \times 10^{-3}$	$1.39 \times 10^{-1} \pm 2.08 \times 10^{-1}$
10^4	$3.50 \times 10^{-4} \pm 2.62 \times 10^{-3}$	$2.86 \times 10^{-2} \pm 5.28 \times 10^{-2}$

ability is also substantially smaller for NMTS. Table 6 further confirms that this advantage is robust across multiple batch sizes. Therefore, the empirical gain of NMTS is not limited to controlled Gaussian-style examples, nor to one specific non-Gaussian perturbation, but persists across multiple likelihood-free latent-noise settings under the same GLR-based inference framework.

Supplement Information for Sect. 6.2

To further broaden the empirical evaluation in the PDE setting, we additionally replace the Gaussian latent noise in Sect. 6.2 by a non-Gaussian logistic noise model. Specifically, we consider

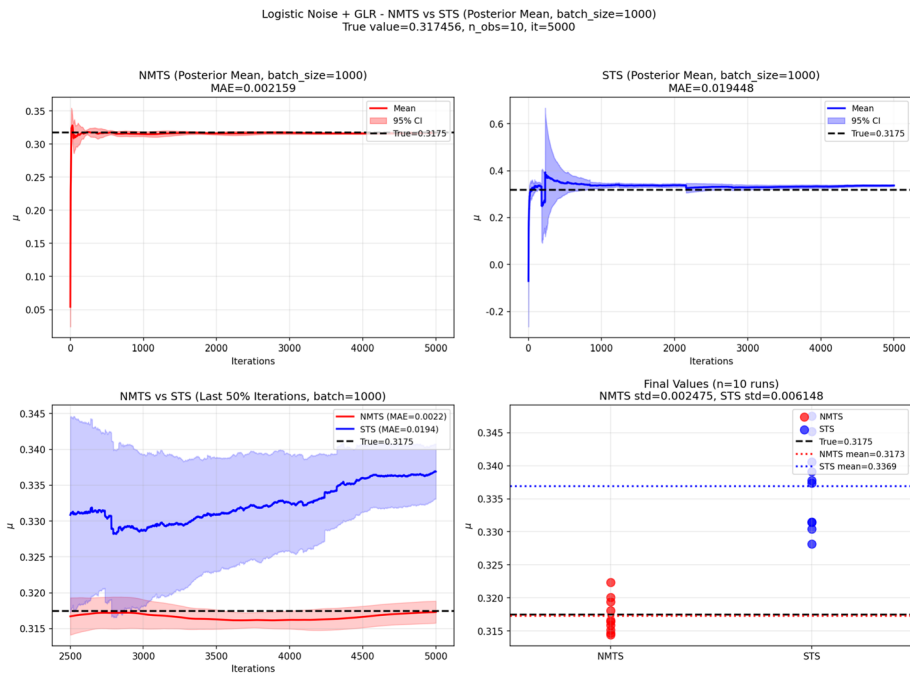


Fig. 9 Convergence comparison of NMTS and STS for the posterior mean in the logistic-noise PDE setting, with batch size 10^3 , based on 10 independent experiments after 5000 iterations

$$Y_t = X_t + \theta, \quad X_t \sim \text{Logistic}(0, 1),$$

while keeping the same prior $\theta \sim N(0, 1)$ and the same variational family $q_\lambda(\theta) \sim N(\mu, \sigma^2)$. Thus, relative to the original PDE experiment, the inference task and the optimization framework remain unchanged, and only the latent-noise distribution is replaced by a non-Gaussian alternative. In this experiment, we continue to use the same GLR construction as in Appendix E to form Monte Carlo estimators of the likelihood and its gradient, and we compute the true posterior mean and posterior variance numerically as the reference targets.

The experimental protocol is kept aligned with the original PDE case. We use 10 observations in one batch, the feasible region $\Lambda = [-1, 10] \times [0.01, 2]$, the initial value $(\mu_0, \sigma_0) = (0, 1)$, $M = 10$ outer-layer samples, and 5000 iterations. For NMTS, we use the fast and slow step-sizes $\alpha_k = \frac{25}{((k+10)\log(k+11))^{0.7}}$, $\beta_k = \frac{5}{(k+10)\log(k+11)}$, while for STS we use $\beta_k = \frac{0.5}{k+10}$. As in all our other comparisons, NMTS and STS are evaluated under matched experimental budgets: the two methods use the same observed data, the same initialization, the same batch size N , the same number of outer-layer samples M , the same number of iterations, and the same number of independent repetitions.

The results again support the robustness of the proposed framework beyond the original Gaussian setting. Figure 9 shows that, for posterior estimation, NMTS exhibits a more stable convergence trajectory and stays consistently closer to the numerically computed

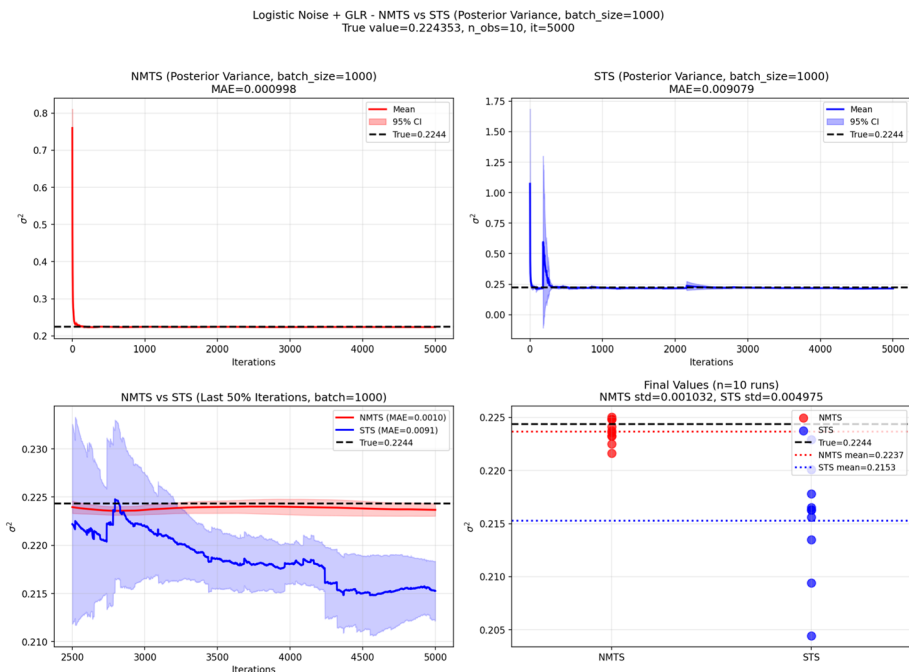


Fig. 10 Convergence comparison of NMTS and STS for the posterior variance in the logistic-noise PDE setting, with batch size 10^3 , based on 10 independent experiments after 5000 iterations

posterior target than STS. Figure 10 shows the same pattern for posterior variance estimation. Table 7 further confirms that this advantage is robust across multiple batch sizes. For $N = 10, 10^2, 10^3, 10^4$, NMTS consistently achieves smaller MAE than STS for both the posterior mean and the posterior variance. In particular, for moderate batch sizes such as 10^2 and 10^3 , the improvement remains substantial, and even at $N = 10^4$, NMTS still maintains a slight edge over STS. These additional PDE results therefore provide further evidence that the empirical gain of NMTS is not tied only to Gaussian examples, but persists in a broader class of likelihood-free inference settings with non-Gaussian latent noise.

Supplement Information for Sect. 6.3

In this part, we describe the methodologies employed to estimate the conditional density $p_\phi(y|\theta)$ using an MAF network and to approximate the posterior $q_\lambda(\theta)$ using an IAF network. Both networks utilize a similar architecture based on autoregressive models, leveraging their distinct advantages for density estimation and sampling. Autoregressive models facilitate the modeling of complex distributions by ensuring that each output feature depends solely on its preceding features. This is achieved through a masking mechanism called Masked Autoencoder for Distribution Estimation (MADE), as detailed in Germain et al. (2015). Figure 11 illustrates the forward MAF algorithm workflow with a single MADE layer.

Table 7 The MAE of the NMTS and STS methods in the logistic-noise PDE setting, based on 10 independent experiments after 5000 iterations

Batch size	Posterior Mean		Posterior Variance	
	NMTS		STS	
	NMTS	STS	NMTS	STS
10	2.94×10^{-2}	1.46×10^0	9.22×10^{-3}	3.90×10^{-1}
10^2	5.29×10^{-3}	4.96×10^{-1}	1.10×10^{-3}	4.75×10^{-1}
10^3	2.16×10^{-3}	1.94×10^{-2}	9.98×10^{-4}	9.08×10^{-3}
10^4	1.15×10^{-3}	1.52×10^{-3}	9.97×10^{-4}	1.32×10^{-3}

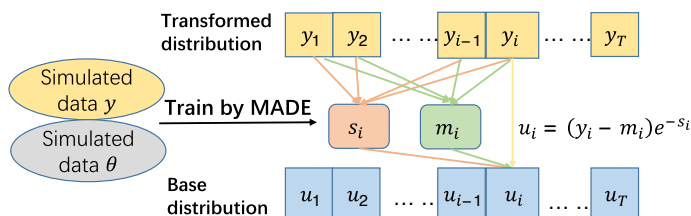


Fig. 11 The autoregressive layer in MAF

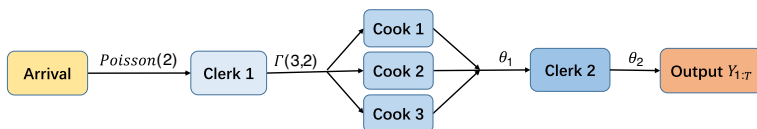


Fig. 12 The flowchart in Sect. 6.3.2

Our constructed MAF network consists of 5 MADE layers, with each MADE layer containing 3 hidden layers and 50 neurons per hidden layer. Each MADE layer produces a series of mean m_i and scale parameters e^{s_i} by training on simulated data y and θ . These parameters enable the transformation of the target distribution into a base distribution, typically a standard normal distribution, through an invertible transformation $u = T(y)$. Note that m_i and s_i are only determined by θ and $y_{1:i-1}$ due to the autoregressive model in MADE, so u_i can be calculated in parallel by formula $u_i = (y_i - m_i)e^{-s_i}$. Furthermore, the calculation of conditional density requires the Jacobian determinant: $\log p(y|\theta) = \log p_u(u) + \log |det(\frac{\partial T}{\partial y})|$. Since this Jacobian matrix is lower diagonal, hence determinant can be computed efficiently, which ensures that we can efficiently calculate the conditional density $p_\phi(y|\theta)$ by plugging the value of u and the Jacobian determinant.

On the other hand, the IAF network mirrors the architecture of the MAF in Fig. 11, which serves as a variational distribution to model the posterior $q_\lambda(\theta)$. It also employs an autoregressive structure, which allows for effective sampling from the approximate posterior. Our IAF network comprises 5 autoregressive layers with 3 hidden layers and 11 neurons per hidden layer. The IAF network generates an invertible transformation that facilitates mapping from a base distribution to the approximate posterior distribution: $y_i = u_i \exp(s_i) + m_i$. Here s_i and m_i are determined by $u_{1:i-1}$, which makes it calculated in parallel. Therefore, IAF is particularly effective for sampling θ from its posterior.

In our experiment, after constructing the above two neural networks, we set up the training parameters as below. The learning rate for the faster scale is $\alpha_k = 10^{-3}$. While the learning rate for the slower scale is $\beta_k = 0.996^k \times 10^{-3}$, satisfying the NMTS condition $\beta_k/\alpha_k \rightarrow 0$. In every iteration, we simulate $M = 10^3$ outer layer samples and $N = 1$ inner layer samples to train the two networks. After 10 rounds of coupled iterations, we can get the posterior of θ based on this sequence of observations $\hat{Y}(X;\theta)$. The process in Sect. 6.3.2 is illustrated as follows.

Author Contributions Zehao Li wrote the manuscript and did the proofs and experiments; Zhouchen Lin reviewed and revised the paper; Yijie Peng proposed the research and reviewed and revised the paper.

Data Availability No datasets were generated or analysed during the current study.

References

- Anastasiou, A., Barp, A., Briol, F.-X., Ebner, B., Gaunt, R. E., Ghaderinezhad, F., Gorham, J., Gretton, A., Ley, C., Liu, Q., et al. (2023). Stein's method meets computational statistics: A review of some recent developments. *Statistical Science*, 38(1), 120–139.
- Bhandari, J., Russo, D., & Singal, R. (2018). A finite time analysis of temporal difference learning with linear function approximation. *Conference on Learning Theory* (pp. 1691–1692) PMLR.
- Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (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* (Vol. Vol. 48). Springer.
- Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, 60(2), 223–311.
- Cao, H., Hu, J.-Q., & Hu, J. (2025). Black-box covar and its gradient estimation. *INFORMS Journal on Computing*. <https://doi.org/10.1287/ijoc.2024.0833>
- Cao, H., Hu, J.-Q., & Hu, J. (2026). A kernel-based stochastic approximation framework for contextual optimization. *European Journal of Operational Research*. <https://doi.org/10.1016/j.ejor.2026.04.021>
- Chung, N.-P., Han, J., Li, B., & Li, Z. (2025). Unbalanced optimal total variation transport: A theoretical approach to spatial resource allocation problems. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems
- Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using real nvp [arXiv:1605.08803](https://arxiv.org/abs/1605.08803) arXiv preprint.
- Doan, T. T. (2021). Finite-time analysis and restarting scheme for linear two-time-scale stochastic approximation. *SIAM Journal on Control and Optimization*, 59(4), 2798–2819.
- Doan, T. T. (2022). Nonlinear two-time-scale stochastic approximation: Convergence and finite-time performance. *IEEE Transactions on Automatic Control*, 68(8), 4695–4705.
- Doan, T. T. (2025). Fast nonlinear two-time-scale stochastic approximation: Achieving $O(1/k)$ finite-sample complexity. *IEEE Transactions on Automatic Control*.
- Doucet, A., & Tadić, V. (2017). Asymptotic bias of stochastic gradient search. *Annals of Applied Probability*, 27(6), 12.
- Figurnov, M., Mohamed, S., & Mnih, A. (2018). Implicit reparameterization gradients. *Advances in neural information processing systems* (p. 31)
- Fu, M. C., Hong, L. J., & Hu, J.-Q. (2009). Conditional Monte Carlo estimation of quantile sensitivities. *Management Science*, 55(12), 2019–2027.
- Germain, M., Gregor, K., Murray, I., & Larochelle, H. (2015). Made: Masked autoencoder for distribution estimation. *International Conference on Machine Learning* (pp. 881–889) PMLR.
- Glasserman, P. (1990). *Gradient Estimation Via Perturbation Analysis* (Vol. Vol. 116). Springer.
- Glöckler, M., Deistler, M., & Macke, J. H. (2022). Variational methods for simulation-based inference [arXiv:2203.04176](https://arxiv.org/abs/2203.04176) arXiv preprint.
- Glynn, P. W., Peng, Y., Fu, M. C., & Hu, J.-Q. (2021). Computing sensitivities for distortion risk measures. *INFORMS Journal on Computing*, 33(4), 1520–1532.
- Greenberg, D., Nonnenmacher, M., & Macke, J. (2019). Automatic posterior transformation for likelihood-free inference. *International Conference on Machine Learning* (pp. 2404–2414) PMLR.
- Gross, D., Shortle, J. F., Thompson, J. M., & Harris, C. M. (2011). *Fundamentals of queueing theory* (Vol. Vol. 627). John Wiley & sons.
- Han, J., Sethi, S., & Yan, T. (2025). Production and risk hedging with cointegration analytics. *Available at SSRN*, 5219499.
- Han, Y., Li, X., & Zhang, Z. (2024). Finite-time decoupled convergence in nonlinear two-time-scale stochastic approximation [arXiv:2401.03893](https://arxiv.org/abs/2401.03893) arXiv preprint.
- Harold, J., Kushner, G., & Yin, G. (1997). Stochastic approximation and recursive algorithm and applications. *Application of Mathematics*, 35(10), 50.
- He, Z., Xu, Z., & Wang, X. (2022). Unbiased MLMC-based variational bayes for likelihood-free inference. *SIAM Journal on Scientific Computing*, 44(4), 1884–1910.

- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). *Gans trained by a two time-scale update rule converge to a local nash equilibrium*. *Advances in neural information processing systems* (p. 30)
- Hong, M., Wai, H.-T., Wang, Z., & Yang, Z. (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., & Doshi, V. (2024). Central limit theorem for two-timescale stochastic approximation with markovian noise: Theory and applications. *International Conference on Artificial Intelligence and Statistics* (pp. 1477–1485) PMLR.
- Hu, J., Peng, Y., Zhang, G., & Zhang, Q. (2022). A stochastic approximation method for simulation-based quantile optimization. *INFORMS Journal on Computing*, 34(6), 2889–2907.
- Hu, J., Song, M., & Fu, M. C. (2024). Quantile optimization via multiple-timescale local search for black-box functions. *Operations Research*. <https://doi.org/10.1287/opre.2022.0534>
- Hyvärinen, A., & Dayan, P. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 120.
- Jiang, J., Hu, J., & Peng, Y. (2023). Quantile-based deep reinforcement learning using two-timescale policy gradient algorithms [arXiv:2305.07248](https://arxiv.org/abs/2305.07248) arXiv preprint.
- Kaledin, M., Moulines, E., Naumov, A., Tadic, V., & Wai, H.-T. (2020). Finite time analysis of linear two-timescale stochastic approximation with markovian noise. *Conference on Learning Theory* (pp. 2144–2203) PMLR.
- Karimi, B., Miasojedow, B., Moulines, E., & Wai, H.-T. (2019). Non-asymptotic analysis of biased stochastic approximation scheme. *Conference on Learning Theory* (pp. 1944–1974) PMLR.
- Karmakar, P., & Bhatnagar, S. (2018). Two time-scale stochastic approximation with controlled markov noise and off-policy temporal-difference learning. *Mathematics of Operations Research*, 43(1), 130–151.
- Khodadadian, S., Doan, T. T., Romberg, J., & Maguluri, S. T. (2022). Finite-sample analysis of two-time-scale natural actor-critic algorithm. *IEEE Transactions on Automatic Control*, 68(6), 3273–3284.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) arXiv preprint.
- Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*, 29, 10.
- Konda, V. R., & Tsitsiklis, J. N. (2004). Convergence rate of linear two-time-scale stochastic approximation
- Lei, L., Peng, Y., Fu, M. C., & Hu, J. (2018). Applications of generalized likelihood ratio method to distribution sensitivities and steady-state simulation. *Discrete Event Dynamic Systems*, 28, 109–125.
- Li, Z., & Peng, Y. (2025). A new stochastic approximation method for gradient-based simulated parameter estimation. *2025 Winter Simulation Conference (WSC)* (pp. 3298–3309). IEEE.
- Li, Z., Ren, T., Zhang, Z., Chen, X., & Peng, Y. (2026). Optimal low-rank stochastic gradient estimation for llm training [arXiv:2603.20632](https://arxiv.org/abs/2603.20632) arXiv preprint.
- Lin, T., Jin, C., & Jordan, M. I. (2025). Two-timescale gradient descent ascent algorithms for nonconvex minimax optimization. *Journal of Machine Learning Research*, 26(11), 1–45.
- Liu, S. D., Chen, S., & Zhang, S. (2025). The ODE method for stochastic approximation and reinforcement learning with markovian noise. *Journal of Machine Learning Research*, 26(24), 1–76.
- Mohamed, S., Rosca, M., Figurnov, M., & Mnih, A. (2020). Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132), 1–62.
- Mokkadem, A., & Pelletier, M. (2006). Convergence rate and averaging of nonlinear two-time-scale stochastic approximation algorithms
- Ong, V. M., Nott, D. J., Tran, M.-N., Sisson, S. A., & Drovandi, C. C. (2018). Variational Bayes with synthetic likelihood. *Statistics and Computing*, 28, 971–988.
- Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57), 1–64.
- Papamakarios, G., Pavlakou, T., & Murray, I. (2017). *Masked autoregressive flow for density estimation*. *Advances in neural information processing systems* (p. 30)
- Papamakarios, G., Sterratt, D., & Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. *The 22nd International Conference on Artificial Intelligence and Statistics* (pp. 837–848) PMLR.
- Peng, Y.-J., Fu, M. C., & Hu, J.-Q. (2014). Gradient-based simulated maximum likelihood estimation for lévy-driven ornstein-uhlenbeck stochastic volatility models. *Quantitative Finance*, 14(8), 1399–1414.
- Peng, Y., Fu, M. C., & Hu, J.-Q. (2016). Gradient-based simulated maximum likelihood estimation for stochastic volatility models using characteristic functions. *Quantitative Finance*, 16(9), 1393–1411.
- Peng, Y., Fu, M. C., Glynn, P. W., & Hu, J. (2017). On the asymptotic analysis of quantile sensitivity estimation by Monte Carlo simulation. *2017 Winter Simulation Conference (WSC)* (pp. 2336–2347). IEEE.

- Peng, Y., Fu, M. C., Heidergott, B. F., & Lam, H. (2020). Maximum likelihood estimation by Monte Carlo simulation: Toward data-driven stochastic modeling. *Operations Research*, 68, 1896–1912.
- Peng, Y., Fu, M. C., Hu, J.-Q., & Heidergott, B. (2018). A new unbiased stochastic derivative estimator for discontinuous sample performances with structural parameters. *Operations Research*, 66(2), 487–499.
- Peters, G. W., Sisson, S. A., & Fan, Y. (2012). Likelihood-free bayesian inference for α -stable models. *Computational Statistics & Data Analysis*, 56(11), 3743–3756.
- Puchkin, N., Samsonov, S., Belomestny, D., Moulines, E., & Naumov, A. (2024). Rates of convergence for density estimation with generative adversarial networks. *Journal of Machine Learning Research*, 25(29), 1–47.
- Ranganath, R., Gerrish, S., & Blei, D. (2014). Black box variational inference. *Artificial Intelligence and Statistics*, 814–822. PMLR.
- Ren, T., Zhang, Z., Li, Z., Jiang, J., Qin, S., Li, G., Li, Y., Zheng, Y., Li, X., Zhan, M., et al. (2025). Zeroth-order informed fine-tuning for diffusion model: A recursive likelihood ratio optimizer. arXiv e-prints, 2502
- Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning* (pp. 1278–1286) PMLR.
- Rezende, D., & Mohamed, S. (2015). Variational inference with normalizing flows. *International Conference on Machine Learning* (pp. 1530–1538) PMLR.
- Rugh, W. J. (1996). *Linear System Theory*. Prentice-Hall Inc.
- Ruiz, F.R., AUEB, T.R., Blei, D., et al. (2016). The generalized reparameterization gradient. *Advances in neural information processing systems* 29
- Sasaki, H., Kanamori, T., & Sugiyama, M. (2017). Estimating density ridges by direct estimation of density-derivative-ratios. *Artificial Intelligence and Statistics*, 204–212. PMLR.
- Sasaki, H., Kanamori, T., Hyvärinen, A., Niu, G., & Sugiyama, M. (2018). Mode-seeking clustering and density ridge estimation via direct estimation of density-derivative-ratios. *Journal of Machine Learning Research*, 18(180), 1–47.
- Shepherd, S. (2014). A review of system dynamics models applied in transportation. *Transportmetrica B: Transport Dynamics*, 2(2), 83–105.
- Shiryayev, A. N., & Boas, R. P. (1995). *Technometrics*, Probability. 2nd ed.
- Sugiyama, M., Suzuki, T., & Kanamori, T. (2012). *Density Ratio Estimation in Machine Learning*. Cambridge University Press.
- Sugiyama, M., Suzuki, T., & Kanamori, T. (2012). Density-ratio matching under the bregman divergence: A unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5), 1009–1044.
- Sugiyama, M., Takeuchi, I., Suzuki, T., Kanamori, T., Hachiya, H., & Okanohara, D. (2010). Least-squares conditional density estimation. *IEICE Transactions on Information and Systems*, 93(3), 583–594.
- Tavaré, S., Balding, D. J., Griffiths, R. C., & Donnelly, P. (1997). Inferring coalescence times from dna sequence data. *Genetics*, 145(2), 505–518.
- Thomas, O., Dutta, R., Corander, J., Kaski, S., & Gutmann, M. U. (2022). Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 17(1), 1–31.
- Tran, D., Ranganath, R., & Blei, D. (2017). Hierarchical implicit models and likelihood-free variational inference. *Advances in Neural Information Processing Systems*, 30, 10.
- Tsitsiklis, J. N. (1994). Asynchronous stochastic approximation and q-learning. *Machine Learning*, 16(3), 185–202.
- Vaart, A. W. V. D. (1998). *Asymptotic statistics cambridge series in statistical and probabilistic mathematics*. Cambridge University Press.
- Wang, Q., Cao, H., Hu, J.-Q., Hu, M., & Xia, L. (2026). Long-run conditional value-at-risk reinforcement learning. arXiv preprint [arXiv:2603.09734](https://arxiv.org/abs/2603.09734)
- Wu, Y. F., Zhang, W., Xu, P., & Gu, Q. (2020). A finite-time analysis of two time-scale actor-critic methods. *Advances in Neural Information Processing Systems*, 33, 17617–17628.
- Zeng, S., Doan, T. T., & Romberg, J. (2024). A two-time-scale stochastic optimization framework with applications in control and reinforcement learning. *SIAM Journal on Optimization*, 34(1), 946–976.
- Zheng, Y., Li, Z., Jiang, P., & Peng, Y. (2024). Dual-agent deep reinforcement learning for dynamic pricing and replenishment [arXiv:2410.21109](https://arxiv.org/abs/2410.21109) arXiv preprint.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Zehao Li¹ · Zhouchen Lin² · Yijie Peng^{3,4} 

✉ Yijie Peng
pengyijie@nju.edu.cn

Zehao Li
zhaoli@stu.pku.edu.cn

Zhouchen Lin
zlin@pku.edu.cn

¹ Department of Management Science and Information Systems, Guanghua School of Management, Peking University, Beijing 100871, China

² State Key Lab of General AI, School of Intelligence Science and Technology, Peking University, Beijing 100871, China

³ School of Management and Engineering, Nanjing University, Nanjing 210093, China

⁴ Xiangjiang Laboratory, Changsha 410023, China