Bayesian Analysis (2023) **TBA**, Number TBA, pp. 1

Stochastic Gradient MCMC for Nonlinear State Space Models

Christopher Aicher[*], Srshti Putcha[†], Christopher Nemeth[‡], Paul Fearnhead[‡], and Emily Fox[§]

Abstract. State space models (SSMs) provide a flexible framework for modeling complex time series via a latent stochastic process. Inference for nonlinear, non-Gaussian SSMs is often tackled with particle methods that do not scale well to long time series. The challenge is two-fold: not only do computations scale linearly with time, as in the linear case, but particle filters additionally suffer from increasing particle degeneracy with longer series. Stochastic gradient MCMC methods have been developed to scale Bayesian inference for finite-state hidden Markov models and linear SSMs using buffered stochastic gradient estimates to account for temporal dependencies. We extend these stochastic gradient estimators to nonlinear SSMs using particle methods. We present error bounds that account for both buffering error and particle error in the case of nonlinear SSMs that are log-concave in the latent process. We evaluate our proposed particle buffered stochastic gradient using stochastic gradient MCMC for inference on both long sequential synthetic and minute-resolution financial returns data, demonstrating the importance of this class of methods.

Keywords: Bayesian inference, exponential forgetting, Markov chain Monte Carlo, nonlinear state space model, particle filtering, stochastic gradient.

1 Introduction

Nonlinear *state space models* (SSMs) are widely used in many scientific domains for modeling time series. For example, nonlinear SSMs can be applied in engineering (e.g. target tracking, Gordon et al. 1993), in epidemiology (e.g. compartmental disease models, Dukic et al. 2012), and to financial time series (e.g. stochastic volatility models, Shephard 2005). To capture complex dynamical structure, nonlinear SSMs augment the observed time series with a latent state sequence, inducing a Markov chain dependence structure. Parameter inference for nonlinear SSMs requires us to handle this latent state sequence. This is typically achieved using *particle filtering* methods.

Particle filtering algorithms are a set of flexible Monte Carlo simulation-based methods, which use a set of samples, also known as *particles*, to approximate the posterior distribution over the latent states. Unfortunately, inference in nonlinear SSMs does not scale well to long sequences: (i) the cost of each update requires full passes through the data that scales linearly with the length of the sequence, and (ii) the number

[*] Department of Statistics, University of Washington
[†] STOR-i Centre for Doctoral Training, Lancaster University
[‡] Department of Mathematics and Statistics, Lancaster University
[§] Departments of Statistics and Computer Science, Stanford University

of particles (and hence the computation per data point) required to control the bias of the particle filter scales linearly with the length of the sequence (Kantas et al., 2015).

Stochastic gradient Markov chain Monte Carlo (SG-MCMC) is a popular method for scaling Bayesian inference to large data sets, replacing full data gradients with stochastic gradient estimates based on subsets of data (Welling and Teh, 2011; Ma et al., 2015). In the context of SSMs, naive stochastic gradients are biased because subsampling breaks temporal dependencies in the data (Ma et al., 2017; Aicher et al., 2019). To correct for this, Ma et al. (2017) and Aicher et al. (2019) have developed *buffered* stochastic gradient estimators that control the bias. The latent state sequence is marginalized in a buffer around each subsequence, which reduces the effect that breaking dependencies has on the estimate of the gradient. However, the work so far has been limited to SSMs where analytic marginalization is possible (e.g. finite-state HMMs and linear dynamical systems).

In this work, we propose *particle buffered* gradient estimators that generalize the buffered gradient estimators to nonlinear SSMs. Although straightforward in concept, a number of unique challenges arise in this setting. First, we show how buffering in nonlinear SSMs can be approximated with a modified particle filter. Second, we provide an error analysis of our proposed estimators by decomposing the error into subsequence error, buffering error, and particle filter error and analyze how this error propagates to estimating posterior means with SGMCMC. Third, we extend the buffering error bounds of Aicher et al. (2019) to nonlinear SSMs with log-concave likelihoods and show that buffer error decays geometrically in buffer size, ensuring that a small buffer size can be used in practice.

The theory we present highlights the importance of controlling bias in the estimate of the gradient – as whilst the impact of a high variance estimator on the accuracy of the SG-MCMC algorithm can be controlled by increasing the number of steps and reducing the step size, it is not possible to change the implementation of the SG-MCMC algorithm to reduce the impact of the bias. We then show theoretically that introducing buffering enables us to control the bias of the estimates of the gradient – with the bias decaying geometrically in the size of the buffer. We investigate the accuracy of our new approach on a range of models with both synthetic and real data – and show that for fixed computational cost we have obtained substantial gains in accuracy over alternatives. This is due to the reduced bias relative to unbuffered versions of SG-MCMC and through the fact that using stochastic gradient methods allows for more iterations of the MCMC algorithm when compared to approaches that estimate gradients using all observations.

Python code for our Algorithm and for replicating our numerical studies is available at `https://github.com/aicherc/sgmcmc_ssm_code`.

2 Background

2.1 Nonlinear State Space Models for Time Series

State space models are a class of discrete-time bivariate stochastic processes consisting of a latent state process $X = \{X_t \in \mathbb{R}^{d_x}\}_{t=1}^T$ and a second observed process, $Y =$

$\{Y_t \in \mathbb{R}^{d_y}\}_{t=1}^T$. The evolution of the state variables is typically assumed to be a time-homogeneous Markov process, such that the latent state at time t, X_t, is determined only by the latent state at time $t-1$, X_{t-1}. The observed states are conditionally independent given the latent states. Given the prior $X_0 \sim \nu(x_0|\theta)$ and parameters $\theta \in \Theta$, the generative model for X, Y is thus

$$\begin{aligned} X_t|(X_{t-1} = x_{t-1}, \theta) &\sim p(x_t \,|\, x_{t-1}, \theta), \\ Y_t|(X_t = x_t, \theta) &\sim p(y_t \,|\, x_t, \theta), \end{aligned} \tag{1}$$

where we call $p(x_t \,|\, x_{t-1}, \theta)$ the *transition density* and $p(y_t \,|\, x_t, \theta)$ the *emission density*.

For an arbitrary sequence $\{z_i\}$, we use $z_{i:j}$ to denote the sequence $(z_i, z_{i+1}, \ldots, z_j)$. To infer the model parameters θ, a quantity of interest is the *score function*, the gradient of the marginal loglikelihood, $\nabla_\theta \log p(y_{1:T}|\theta)$. Using the score function, the loglikelihood can be maximized iteratively via a (batch) *gradient ascent* algorithm (Robbins and Monro, 1951), given the observations, $y_{1:T}$.

If the latent state posterior $p(x_{1:T}|y_{1:T}, \theta)$ can be expressed analytically, we can calculate the score using *Fisher's identity* (Cappé et al., 2005),

$$\begin{aligned} \nabla_\theta \log p(y_{1:T} \,|\, \theta) &= \mathbb{E}_{X|Y,\theta}[\nabla_\theta \log p(X_{1:T}, y_{1:T} \,|\, \theta)] \\ &= \sum_{t=1}^T \mathbb{E}_{X|Y,\theta}[\nabla_\theta \log p(X_t, y_t \,|\, x_{t-1}, \theta)]. \end{aligned} \tag{2}$$

If the latent state posterior, $p(x_{1:T}|y_{1:T}, \theta)$, is not available in closed-form, we can approximate the expectations of the latent state posterior. One popular approach is via *particle filtering* methods.

Particle Filtering and Smoothing

Particle filtering algorithms (see e.g. Doucet and Johansen, 2009; Fearnhead and Künsch, 2018) can be used to create an empirical approximation of the expectation of a function $H(X_{1:T})$ with respect to the posterior density, $p(x_{1:T}|y_{1:T}, \theta)$. This is done by generating a collection of N random samples or *particles*, $\{x_t^{(i)}\}_{i=1}^N$ and calculating their associated importance weights, $\{w_t^{(i)}\}_{i=1}^N$, recursively over time. We update the particles and weights with *sequential importance resampling* (Doucet and Johansen, 2009) in the following manner.

(i) *Resample* auxiliary ancestor indices $\{a_1, \ldots, a_N\}$ with probabilities proportional to the importance weights, i.e. $a_i \sim \text{Categorical}(w_{t-1}^{(i)})$.

(ii) *Propagate* particles $x_t^{(i)} \sim q(\cdot|x_{t-1}^{(a_i)}, y_t, \theta)$, using a proposal distribution $q(\cdot|\cdot)$.

(iii) *Update* and normalize the weight of each particle,

$$w_t^{(i)} \propto \frac{p(y_t|x_t^{(i)}, \theta) p(x_t^{(i)}|x_{t-1}^{(a_i)}, \theta)}{q(x_t^{(i)}|x_{t-1}^{(a_i)}, y_t, \theta)} \quad , \quad \sum_i w_t^{(i)} = 1 \quad . \tag{3}$$

The auxiliary variables, $\{a_i\}_{i=1}^N$, represent the indices of the *ancestors* of the particles, $\{x_t^{(i)}\}_{i=1}^N$, sampled at time t. The introduction of ancestor indices allows us to keep track of the lineage of particles over time (Andrieu et al., 2010). The *multinomial resampling* scheme given in (i) describes the procedure by which *offspring* particles are produced.

Resampling at each iteration is used to mitigate against the problem of *weight degeneracy*. This phenomenon occurs when the variance of the importance weights grows, causing more and more particles to have negligible weight. Aside from the multinomial resampling scheme described above, there are various other resampling schemes outlined in the particle filtering literature, such as stratified sampling (Kitagawa, 1996) and residual sampling (Liu and Chen, 1998).

If the proposal density $q(x_t|x_{t-1}, y_t, \theta)$ is the transition density $p(x_t|x_{t-1}, \theta)$ we obtain the *bootstrap particle filter* (Gordon et al., 1993). By using the transition density for proposals, the importance weight recursion in (3) simplifies to $w_t^{(i)} \propto p(y_t|x_t^{(i)}, \theta)$.

When our target function decomposes into a pairwise sum $H(x_{1:T}) = \sum_{t=1}^T h_t(x_t, x_{t-1})$ – such as for Fisher's identity $h_t(x_t, x_{t-1}) = \nabla_\theta \log p(y_t, x_t \mid x_{t-1}, \theta)$ – then we only need to keep track of the partial sum $H_t = \sum_{s=1}^t h_s(x_s, x_{s-1})$ in the filter (Doucet and Johansen, 2009): see Algorithm 1.

Algorithm 1 Particle Filter

```
1: Input: number of particles, $N$, pairwise statistics, $h_{1:T}$, observations $y_{1:T}$, proposal
   density $q$,
2: Draw $x_0^{(i)} \sim \nu(x_0|\theta)$, set $w_0^{(i)} = \frac{1}{N}$, and $H_0^{(i)} = 0 \ \forall i$.
3: for $t = 1, \ldots, T$ do
4:     Resample ancestor indices $\{a_1, \ldots, a_N\}$.
5:     Propagate particles $x_t^{(i)} \sim q(\cdot|x_{t-1}^{(a_i)}, y_t, \theta)$.
6:     Update each $w_t^{(i)}$ according to (3).
7:     Update statistics $H_t^{(i)} = H_{t-1}^{(a_i)} + h_t(x_t^{(i)}, x_{t-1}^{(a_i)})$.
8: end for
9: Return $H = \sum_{i=1}^N w_T^{(i)} H_T^{(i)}$.
```

A key challenge for particle filters is handling large T. Not only do long sequences require $\mathcal{O}(T)$ computation, but particle filters require a large number of particles, N, to avoid *particle degeneracy*: the use of resampling in the particle filter causes path-dependence over time, depleting the number of distinct particles available overall. For Algorithm 1, the variance in H scales as $\mathcal{O}(T^2/N)$ (Poyiadjis et al., 2011). Therefore to maintain a constant variance, the number of particles would need to increase quadratically with T, which is computationally infeasible for long sequences. Poyiadjis et al. (2011); Nemeth et al. (2016) and Olsson and Westerborn (2017) propose alternatives to Step 7 of Algorithm 1 that trade additional computation or bias to decrease the variance in H to $\mathcal{O}(T/N)$. Fixed-lag particle smoothers provide another approach to avoid particle degeneracy, where sample paths are not updated after a fixed lag (Kitagawa and Sato, 2001; Dahlin et al., 2015). All of these methods perform a full pass over the data $y_{1:T}$, which requires $\mathcal{O}(T)$ computation.

2.2 Stochastic Gradient MCMC

One popular method to conduct scalable Bayesian inference for large data sets is *stochastic gradient* Markov chain Monte Carlo (SGMCMC). Given a prior $p(\theta)$, to draw a sample θ from the posterior $p(\theta|y) \propto p(y|\theta)p(\theta)$, gradient-based MCMC methods simulate a stochastic differential equation (SDE) based on the gradient of the loglikelihood $g_\theta = \nabla_\theta \log p(y|\theta)$, such that the posterior is the stationary distribution of the SDE. SGMCMC methods replace the full-data gradients with stochastic gradients, $\widehat{g}_\theta$, using subsamples of the data to avoid costly computation.

The most common method of the SGMCMC family is the *stochastic gradient Langevin dynamics* (SGLD) algorithm (Welling and Teh, 2011; Nemeth and Fearnhead, 2021):

$$\theta^{(k+1)} \leftarrow \theta^{(k)} + \epsilon^{(k)} \cdot (\widehat{g}_\theta + \nabla \log p(\theta)) + \mathcal{N}(0, 2\epsilon^{(k)}), \tag{4}$$

where $\epsilon^{(k)}$ is the stepsize and θ_1 is an initialization of the chain. When $\widehat{g}_\theta$ is unbiased and with an appropriate decreasing stepsize, the distribution of $\theta^{(k)}$ asymptotically converges to the posterior distribution (Teh et al., 2016). Dalalyan and Karagulyan (2019) provide non-asymptotic bounds on the Wasserstein distance between the posterior and the output of SGLD after K steps for fixed $\epsilon^{(k)} = \epsilon$ and possibly biased $\widehat{g}_\theta$.

Many extensions of SGLD exist in the literature, including using control variates to reduce the variance of $\widehat{g}_\theta$ (Baker et al., 2019; Nagapetyan et al., 2017; Chatterji et al., 2018) and augmented dynamics to improve mixing (Ma et al., 2015) such as stochastic gradient Hamiltonian Monte Carlo (Chen et al., 2014), stochastic gradient Nosé-Hoover thermostat (Ding et al., 2014), and stochastic gradient Riemannian Langevin dynamics (Girolami and Calderhead, 2011; Patterson and Teh, 2013).

Stochastic Gradients for SSMs

An additional challenge when applying SGMCMC to SSMs is handling the temporal dependence between observations. Based on a subset $\mathcal{S}$ of size S, an unbiased stochastic gradient estimate of (2) is

$$\sum_{t \in \mathcal{S}} \Pr(t \in \mathcal{S})^{-1} \cdot \mathbb{E}_{X|y_{1:T},\theta}[\nabla_\theta \log p(X_t, y_t \,|\, X_{t-1}, \theta)]. \tag{5}$$

Although (5) is a sum over S terms, it requires taking expectations with respect to $p(x|y_{1:T},\theta)$, which requires processing the full sequence $y_{1:T}$. One approach to reduce computation is to randomly sample $\mathcal{S}$ as a contiguous subsequence $\mathcal{S} = \{s+1, \ldots, s+S\}$ and approximate (5) using only $y_\mathcal{S}$

$$\sum_{t \in \mathcal{S}} \Pr(t \in \mathcal{S})^{-1} \cdot \mathbb{E}_{X|y_\mathcal{S},\theta}[\nabla_\theta \log p(X_t, y_t \,|\, X_{t-1}, \theta)]. \tag{6}$$

However, (6) is *biased* because the expectation over the latent states $x_\mathcal{S}$ is conditioned only on $y_\mathcal{S}$ rather than $y_{1:T}$.

To control the bias in stochastic gradients while also avoiding accessing the full sequence, previous work on SGMCMC for SSMs proposed *buffered* stochastic gradients (Ma et al., 2017; Aicher et al., 2019).

$$\widehat{g}_\theta(S, B) = \sum_{t \in \mathcal{S}} \frac{\mathbb{E}_{X|y_{\mathcal{S}^*},\theta}[\nabla_\theta \log p(X_t, y_t \,|\, X_{t-1}, \theta)]}{\Pr(t \in \mathcal{S})}, \tag{7}$$

where $\mathcal{S}^* = \{s+1-B, \ldots, s+S+B\}$ is the *buffered* subsequence such that $\mathcal{S} \subseteq \mathcal{S}^* \subseteq \{1, \ldots, T\}$ (see Figure 1). When the "buffer" extends outside of the original subsequence (e.g. $s+1-B < 1$ or $s+S+B > T$), then we can extend the model to $\{1-B, \ldots, T+B\}$ and assume the observations y_t outside of $\{1, \ldots, T\}$ are missing. In practice, we will truncate $\mathcal{S}^*$ by intersecting it with $\{1, \ldots, T\}$.

The unbiased gradient estimate, which conditions on all data (5), is $\widehat{g}(S, T)$ and the estimator with no buffering (6) is $\widehat{g}(S, 0)$. As B increases from 0 to T, the estimator $\widehat{g}_\theta(S, B)$ trades computation for reduced bias. In particular, when the model and gradient both satisfy a Lipschitz property, the error decays geometrically in buffer size B, see Theorem 4.1 of Aicher et al. (2019). Specifically, for all $\mathcal{S}$

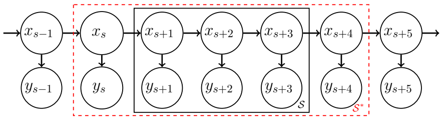


Figure 1: Graphical model of $\mathcal{S}^*$ with $S = 3$ and $B = 1$.

$$\|\widehat{g}_\theta(S, B) - \widehat{g}_\theta(S, T)\|_2 = \mathcal{O}(L_\theta^B \cdot T/S), \tag{8}$$

where L_θ is a bound for the Lipschitz constants of the *forward and backward smoothing kernels*[1]

$$\begin{aligned} \vec{\Psi}_t(x_{t+1}, x_t) &= p(x_{t+1} \,|\, x_t, y_{1:T}, \theta), \\ \overleftarrow{\Psi}_t(x_{t-1}, x_t) &= p(x_{t-1} \,|\, x_t, y_{1:T}, \theta). \end{aligned} \tag{9}$$

The bound provided in (8) ensures that only a modest buffer size B is required (e.g. $\mathcal{O}(\log \delta^{-1})$ for an accuracy of δ). Unfortunately, neither the buffered stochastic gradient $\widehat{g}_\theta(S, B)$ nor the smoothing kernels $\{\vec{\Psi}_t, \overleftarrow{\Psi}_t\}$ have a closed form for nonlinear SSMs.

3 Method

In this section, we propose a particle buffered stochastic gradient for nonlinear SSMs, by applying the particle approximations of Section 2.1 to (7).

[1] We follow Aicher et al. (2019) and consider Lipschitz constants for a kernel Ψ measured in terms of the p-Wasserstein distance between distributions of x, x' and $\Psi(x), \Psi(x')$.

3.1 Buffered Stochastic Gradient Estimates for Nonlinear SSMs

Let $g_\theta^{\mathrm{PF}}(S, B, N)$ denote the particle approximation of $\widehat{g}_\theta(S, B)$ with N particles. We approximate the expectation over $p(x|y_{\mathcal{S}^*}, \theta)$ in (7) using Algorithm 1 run over $\mathcal{S}^*$. In the following we will use ν_0 as the prior distribution for X_{s+1-B}, which is a natural choice if the state process is stationary and ν_0 is its stationary distribution; for other cases better choices for the prior distribution of X_{s+1-B} may be possible.

The complete data loglikelihood, $\log p(y_\mathcal{S}, x_\mathcal{S}, \theta)$, in (7) decomposes into a sum of pairwise statistics

$$H = \sum_{t \in \mathcal{S}^*} h_t(x_t, x_{t-1}) \ , \tag{10}$$

where

$$h_t(x_t, x_{t-1}) = \begin{cases} \dfrac{\nabla_\theta \log p(x_t, y_t \,|\, x_{t-1}, \theta)}{\Pr(t \in \mathcal{S})} & \text{if } t \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

We highlight that the statistic is zero for t in the left and right buffers $\mathcal{S}^* \backslash \mathcal{S}$. Although H_t is not updated by h_t for t in $\mathcal{S}^* \backslash \mathcal{S}$, running the particle filter over the buffers is *crucial* to reduce the bias of $g_\theta^{\mathrm{PF}}(S, B, N)$.

Note that $g_\theta^{\mathrm{PF}}(S, B, N)$ allows us to approximate the non-analytic expectation in (7) with a modest number of particles N, by avoiding the particle degeneracy and full sequence runtime bottlenecks, as the particle filter is only run over $\mathcal{S}^*$, which has length $S + 2B \ll T$.

3.2 SGMCMC Algorithm

Using $g_\theta^{\mathrm{PF}}(S, B, N)$ as our stochastic gradient estimate in SGLD, (4), gives us Algorithm 2.

Algorithm 2 Buffered PF-SGLD

```
Input: data $y_{1:T}$, initial $\theta^{(0)}$, stepsize $\epsilon$, subsequence size $S$, buffer size $B$, particle size $N$
for $k = 1, 2, \ldots, K$ do
    Sample $\mathcal{S} = \{s+1, \ldots, s+S\}$
    Set $\mathcal{S}^* = \{s+1-B, \ldots, s+S+B\}$.
    Calculate $g_\theta^{\mathrm{PF}}$ over $\mathcal{S}^*$ using Alg. 1 on (11).
    Set $\theta^{(k+1)} \leftarrow \theta^{(k)} + \epsilon \cdot (g_\theta^{\mathrm{PF}} + \nabla \log p(\theta)) + \mathcal{N}(0, 2\epsilon)$
end for
Return $\theta^{(K+1)}$
```

Algorithm 2 can be extended by (i) averaging over multiple sequences or varying the subsequence sampling method (Schmidt et al., 2015; Ou et al., 2018), (ii) using different particle filters such as those listed in Section 2.1, and (iii) using more advanced SGMCMC schemes such as those listed in Section 2.2.

4 Error Analysis

In this section, we analyze the error of our particle buffered stochastic gradient g_θ^{PF} and its effect on approximating posterior means with finite sample averages using Algorithm 2. We first present error bounds for approximating posterior means using SGLD with biased gradients (Theorem 1). We then present bounds on the gradient bias and MSE of g_θ^{PF}, extending the error bounds of Aicher et al. (2019) (Theorem 2). In particular, we provide bounds for the Lipschitz constant L_θ of the smoothing kernels (9) without requiring an explicit form for the smoothing kernels (Theorem 3), allowing (8) to apply to nonlinear SSMs.

4.1 Error of Biased SGLD's Finite Sample Averages

We consider the estimation error of the posterior expected value of some test function of the parameters $\phi : \Theta \to \mathbb{R}$ using samples $\theta^{(k)}$ drawn using SGLD with a fixed step size ϵ and stochastic gradients g_θ.

Let $\bar{\phi}$ be the posterior expected value

$$\bar{\phi} = \mathbb{E}_{p(\theta|y)}[\phi(\theta)] \ , \tag{12}$$

and let $\hat{\phi}_{K,\epsilon}$ be the K-sample estimator for $\bar{\phi}$

$$\hat{\phi}_{K,\epsilon} = \frac{1}{K}\sum_{k=1}^{K} \phi(\theta^{(k)}) \ . \tag{13}$$

The error of the *finite sample average* $|\hat{\phi}_{K,\epsilon} - \bar{\phi}|$ has been previously studied for SGLD with *unbiased* gradients by Vollmer et al. (2016) and Chen et al. (2015). Following Chen et al. (2015), we make the following assumption on ϕ.

Assumption 1. *Let $\mathcal{L}$ be the generator of the Langevin diffusion*

$$\mathcal{L}[\psi(\theta_t)] = -\nabla \log p(\theta_t) \cdot \nabla\psi(\theta_t) + \frac{\epsilon^2}{2}\operatorname{tr}(\nabla^2\psi(\theta_t)) \ .$$

Then, we define ψ to solve the Poisson equation

$$\frac{1}{K}\sum_{k=1}^{K} \mathcal{L}[\psi(\theta^{(k)})] = \hat{\phi}_{K,\epsilon} - \bar{\phi} \ . \tag{14}$$

We assume that $\psi(\theta)$ and its derivatives (up to third order) are bounded.

We now present Theorem 1, which bounds the error of a finite sample Monte Carlo estimator based on SGLD when the stochastic gradients $\hat{g}_\theta$ are potentially *biased*.

Theorem 1 (Error of Finite Sample Average)**.** *If the gradient g_θ is smooth in θ, the test function ϕ satisfies a moment condition (Assumption 1) and the bias and MSE of the gradient estimates $\hat{g}_\theta$ are uniformly bounded, that is,*

$$\|\mathbb{E}\,\hat{g}_\theta - g_\theta\| \leq \delta \textit{ and } \mathbb{E}\,\|\hat{g}_\theta - g_\theta\|^2 \leq \sigma^2 \textit{ for all } \theta \ , \tag{15}$$

then there exists some constant $C > 0$, such that the bias and MSE of $\hat{\phi}_{K,\epsilon}$ satisfy

$$|\mathbb{E}\, \hat{\phi}_{K,\epsilon} - \bar{\phi}\,| \leq C \cdot \left(\frac{1}{K\epsilon} + \delta\right) + \mathcal{O}(\epsilon) \ , \tag{16}$$

$$\mathbb{E}\,|\hat{\phi}_{K,\epsilon} - \bar{\phi}\,|^2 \leq C\left(\frac{1}{K^2\epsilon^2} + \frac{\sigma^2}{K} + \delta^2 + \frac{\delta}{\epsilon}\right) + \mathcal{O}\left(\frac{1}{K\epsilon} + \delta\epsilon + \epsilon^2\right) \ . \tag{17}$$

The bias bound, (16), is a direct application of Theorem 2 in Chen et al. (2015). The MSE bound, (17), is an extension of Theorem 3 in Chen et al. (2015) when the stochastic gradient estimates $\hat{g}_\theta$ are biased (i.e. $\delta \neq 0$). The additional bias terms δ arise from keeping track of additional cross terms in $(\hat{\phi}_{K,\epsilon} - \bar{\phi})^2$. The proof of Theorem 1 is presented in the Supplement (Aicher et al., 2023).

From Theorem 1, we see that the error bounds on $\hat{\phi}_{K,\epsilon}$ are more sensitive to the bias δ of $\hat{g}$ than the variance σ^2: the term involving σ^2 decays with increasing K, while terms involving δ do not decay regardless of stepsize ϵ or number of samples K. A similar conclusion comes from the bound on error of SGLD in Theorem 4 of Dalalyan and Karagulyan (2019): the impact of bias on the error bound is not affected by step size, whereas the impact of the variance can be reduced by taking more steps of smaller size; however, we do not require the posterior distribution be log-concave.

Therefore for the samples from Algorithm 2 to be useful, it is important for the bias of g_θ^{PF} to be controlled.

4.2 Gradient Bias and MSE Bounds

To apply Theorem 1 to the samples from Algorithm 2, we develop bounds on the bias δ and MSE σ^2 of our particle buffered stochastic gradients g_θ^{PF}.

Theorem 2 (Bias and MSE Bounds for g_θ^{PF})**.** *For fixed θ, if the model and gradient satisfy a Lipschitz condition and there is a bound on the autocorrelation between $\mathbb{E}_{X|y_{1:T}} \nabla \log p(y_t, X_t | X_{t-1}, \theta)$ for different t, then the bias δ and MSE σ^2 of g_θ^{PF} is bounded by*

$$\delta \leq \gamma \cdot \left[C_1 \cdot L_\theta^B + \mathcal{O}\left(\frac{S + 2B}{N}\right)\right] \ , \tag{18}$$

$$\sigma^2 \leq 3\gamma^2 \cdot \left[C_1^2 \cdot L_\theta^{2B} + C_2 S + \mathcal{O}\left(\frac{(S + 2B)^2}{N}\right)\right] \ , \tag{19}$$

where $\gamma = \max_t \Pr(t \in \mathcal{S})^{-1}$ and C_1, C_2 are constants with respect to S, B, N.

From Theorem 2, we see that the bias δ (18) can be controlled by selecting large enough N and B when $L_\theta < 1$.

We now sketch the proof of Theorem 2 and discuss its assumptions. The complete proof can be found in the Supplement (Aicher et al., 2023).

We decompose the error between g_θ^{PF} and the full gradient g_θ through $\hat{g}_\theta(S,B)$ and $\hat{g}_\theta(S,T)$ into three error sources:

$$\begin{aligned}\|g_\theta^{\mathrm{PF}}(S,B,N) - g_\theta\| \leq\ & \underbrace{\|g_\theta^{\mathrm{PF}}(S,B,N) - \hat{g}_\theta(S,B)\|}_{\text{particle error (I)}} + \\ & \underbrace{\|\hat{g}_\theta(S,B) - \hat{g}_\theta(S,T)\|}_{\text{buffering error (II)}} + \underbrace{\|\hat{g}_\theta(S,T) - g_\theta\|}_{\text{subsequence error (III)}} \ . \end{aligned} \tag{20}$$

(I) *Particle error*: the Monte Carlo error of the particle filter. From Kantas et al. (2015), the asymptotic bias and MSE of a particle approximation to the sum of R test functions (using Algorithm 1) is $\mathcal{O}(R/N)$ and $\mathcal{O}(R^2/N)$ respectively. Since $g^{\mathrm{PF}}(S,B,N)$ is a particle approximation to the sum of $R = S + 2B$ test functions (i.e., $h_t(x_t, x_{t-1})$), we have

$$\begin{aligned}\|\mathbb{E}\, g_\theta^{\mathrm{PF}}(S,B,N) - \hat{g}_\theta(S,B)\| &= \mathcal{O}\left(\gamma \cdot \frac{S+2B}{N}\right) \\ \mathbb{E}\,\|g_\theta^{\mathrm{PF}}(S,B,N) - \hat{g}_\theta(S,B)\|^2 &= \mathcal{O}\left(\gamma^2 \cdot \frac{(S+2B)^2}{N}\right) \ , \end{aligned} \tag{21}$$

where γ is a upper bound on the sampling scale factor $\gamma = \max_t \Pr(t \in \mathcal{S})^{-1}$.

Using a more advanced particle filter, such as the “PaRIS” or “Poyiadjis N^2” algorithm, Corollary 6 of Olsson and Westerborn (2017) gives a tighter bound for the MSE

$$\mathbb{E}\,\|g_\theta^{\mathrm{PF}}(S,B,N) - \hat{g}_\theta(\,S,B)\|^2 = \mathcal{O}\left(\gamma^2 \cdot \frac{S+2B}{N}\right) \ .$$

However in our experiments, we found that the improved MSE of these other particle filters was not worth the additional computational overhead for the small subsequences we considered, where $S + 2B \lesssim 100$. See experiments in the Supplement (Aicher et al., 2023).

(II) *Buffering error,*: error in approximating the latent state posterior $p(x_{1:T}|y_{1:T})$ with $p(x_{1:T}|y_{\mathcal{S}^*})$. The error stems from conditioning on only a buffered subsequence $y_{\mathcal{S}^*}$ instead of $y_{1:T}$ and the initial distribution approximation ν_0 for X_{s+1-B}. If the smoothing kernels $\{\vec{\Psi}_t, \overleftarrow{\Psi}_t\}$ are contractions for all t (i.e. $L_\theta < 1$), then according to (8), the error in this term is proportional to γL_θ^B. In Section 4.3, we show sufficient conditions for $L_\theta < 1$.

(III) *Subsequence error*: the error in approximating Fisher’s identity using a randomly chosen subsequence of data points. The error in this term depends on the subsequence size S and how subsequences are sampled. Because we sample random *contiguous* subsequences of size S, the MSE scales $\mathcal{O}(\gamma^2 S \frac{1+\rho}{1-\rho})$, where ρ is a bound on the autocorrelation between $\mathbb{E}_{X|y_{1:T}} \nabla \log p(y_t, X_t | X_{t-1}, \theta)$ for different t. See the Supplement (Aicher et al., 2023) for details.

Combining these error bounds gives us Theorem 2.

We present examples of the asymptotic bias and MSE bounds given by Theorem 2 for four different gradient estimators in Table 1. The four gradient estimators are: (i) naive stochastic subsequence (without buffering) $g^{\mathrm{PF}}(S,0,N)$ (ii) buffered stochastic subsequence $g^{\mathrm{PF}}(S,B,N)$, (iii) fully buffered stochastic subsequence $g^{\mathrm{PF}}(S,T,N)$, and (iv) full sequence $g^{\mathrm{PF}}(T,T,N)$. For simplicity, we assume the subsequences $\mathcal{S}$ are sampled from a *strict* partition of $1:T$ such that $\gamma = T/S$ and assume B is on the same order as S (i.e. B is $\mathcal{O}(S)$).

Table 1: Asymptotic bias and compute cost for four different gradient estimators.

Gradient	(S,B,N)	Bias δ	Compute
Naive Subsequence	$(S,0,N)$	$C_1 \cdot T/S + \mathcal{O}(T/N)$	$\mathcal{O}(SN)$
Buffered Subsequence	(S,B,N)	$C_1 \cdot L_\theta^B \cdot T/S + \mathcal{O}(T/N)$	$\mathcal{O}(SN)$
Fully Buffered Subsequence	(S,T,N)	$\mathcal{O}(T/N)$	$\mathcal{O}(TN)$
Full Sequence	(T,T,N)	$\mathcal{O}(T/N)$	$\mathcal{O}(TN)$

From Table 1, we see that without buffering, the naive stochastic gradient has a $C_1 \cdot T/S$ term in the bias bound δ. The fully buffered subsequence and full sequence gradients remove the buffering error entirely, but require $\mathcal{O}(TN)$ computation. Instead, our proposed buffered stochastic gradient controls the bias, with the geometrically decaying factor L_θ^B, using only $\mathcal{O}(SN)$ computation.

4.3 Buffering Error Bound for Nonlinear SSMs

To obtain a bound for the buffering error term (II), we require the Lipschitz constant L_θ of smoothing kernels $\{\vec{\Psi}_t, \overleftarrow{\Psi}_t\}$ to be less than 1. Typically the smoothing kernels $\vec{\Psi}_t, \overleftarrow{\Psi}_t$ are not available in closed-form for nonlinear SSMs and therefore directly bounding the Lipschitz constant is difficult. However, we now show that when the model's transition and emission densities are *log-concave* in x_t, x_{t-1}, we can bound the Lipschitz constant of $\vec{\Psi}_t, \overleftarrow{\Psi}_t$ in terms of the Lipschitz constant of either the *prior kernels* $\vec{\Psi}_t^{(0)}, \overleftarrow{\Psi}_t^{(0)}$, or the *filtered kernels* $\vec{\Psi}_t^{(1)}, \overleftarrow{\Psi}_t^{(1)}$

$$\begin{aligned} \vec{\Psi}_t^{(0)} &:= p(x_t \,|\, x_{t-1}, \theta), & \vec{\Psi}_t^{(1)} &:= p(x_t \,|\, x_{t-1}, y_t, \theta), \\ \overleftarrow{\Psi}_t^{(0)} &:= p(x_t \,|\, x_{t+1}, \theta), & \overleftarrow{\Psi}_t^{(1)} &:= p(x_t \,|\, x_{t+1}, y_t, \theta), \end{aligned} \tag{22}$$

Unlike the smoothing kernels, the prior kernels are defined by the model and are therefore usually available. If the filtered kernels are available, then they can be used to obtain even tighter bounds.

Theorem 3 (Lipschitz Kernel Bound)**.** *Assume the prior for x_0 is log-concave in x. If the transition density $p(x_t \,|\, x_{t-1}, \theta)$ is log-concave in (x_t, x_{t-1}) and the emission density $p(y_t \,|\, x_t)$ is log-concave in x_t, then*

$$\|\vec{\Psi}_t\|_{Lip} \leq \|\vec{\Psi}_t^{(1)}\|_{Lip} \leq \|\vec{\Psi}_t^{(0)}\|_{Lip} \tag{23}$$

$$\|\overleftarrow{\Psi}_t\|_{Lip} \leq \|\overleftarrow{\Psi}_t^{(1)}\|_{Lip} \leq \|\overleftarrow{\Psi}_t^{(0)}\|_{Lip}. \tag{24}$$

Therefore

$$\begin{aligned} L_\theta &= \max_t\{\|\vec{\Psi}_t\|_{Lip}, \|\overleftarrow{\Psi}_t\|_{Lip}\} \\ &\leq \max_t\{\|\vec{\Psi}_t^{(1)}\|_{Lip}, \|\overleftarrow{\Psi}_t^{(1)}\|_{Lip}\} \\ &\leq \max_t\{\|\vec{\Psi}_t^{(0)}\|_{Lip}, \|\overleftarrow{\Psi}_t^{(0)}\|_{Lip}\} \end{aligned} \tag{25}$$

This theorem lets us bound L_θ with the Lipschitz constant of either the prior kernels or filtered kernels. The proof of Theorem 3 is provided in the Supplement (Aicher et al., 2023) and uses Caffarelli's log-concave perturbation theorem (Villani, 2008; Colombo et al., 2017). Examples of SSMs for which Theorem 3 applies include the linear Gaussian SSM, the stochastic volatility model, or any linear SSM with log-concave transition and emission distributions.

Theorem 3 lets us calculate analytic bounds on L_θ for the buffering error of Theorem 2. We provide explicit bounds for L_θ for the linear Gaussian SSM and stochastic volatility model in Section 5.1 with proofs in the Supplement (Aicher et al., 2023).

5 Experiments

We first empirically test the bias of our particle buffered gradient estimator g_θ^{PF} on synthetic data for fixed θ. We then evaluate the performance of our proposed SGLD algorithm (Algorithm 2) on both real and synthetic data.

5.1 Models

For our experiments, we consider three models: (i) the linear Gaussian SSM (LGSSM), a case where analytic buffering is possible, to assess the impact of the particle filter; (ii) the stochastic volatility model (SVM) (Shephard, 2005), where the emissions are non-Gaussian; and (iii) the generalized autoregressive conditional heteroskedasticity (GARCH) model (Bollerslev, 1986), where the latent transitions are nonlinear.

Linear Gaussian SSM

The *linear Gaussian SSM* (LGSSM) is

$$\begin{aligned} X_t \,|\, (X_{t-1} = x_{t-1}, \theta) &\sim \mathcal{N}(x_t \,|\, \phi x_{t-1}\, , \, \sigma^2), \\ Y_t \,|\, (X_t = x_t, \theta) &\sim \mathcal{N}(y_t \,|\, x_t\, , \, \tau^2), \end{aligned} \tag{26}$$

with $\nu_0(x_0) = \mathcal{N}(x_0 \,|\, 0, \frac{\phi^2}{1-\sigma^2})$ and parameters $\theta = (\phi, \sigma, \tau)$.

The transition and emission distributions are both Gaussian and log-concave in x, so Theorem 3 applies. In the Supplement (Aicher et al., 2023), we show that the filtered kernels of the LGSSM are bounded with the Lipschitz constant $L_\theta = |\phi| \cdot \sigma^2/(\sigma^2+\tau^2)$. Thus, the buffering error decays geometrically with increasing buffer size B when $|\phi| < (1 + \frac{\tau^2}{\sigma^2})$. This linear model serves as a useful baseline since the various terms in (20) can be calculated analytically.

Stochastic Volatility Model

The *stochastic volatility model* (SVM) is

$$\begin{aligned} X_t \,|\, (X_{t-1} = x_{t-1}, \theta) &\sim \mathcal{N}(x_t \,|\, \phi x_{t-1}\,,\, \sigma^2), \\ Y_t \,|\, (X_t = x_t, \theta) &\sim \mathcal{N}(y_t \,|\, 0\,,\, \exp(x_t)\tau^2), \end{aligned} \tag{27}$$

with $\nu_0(x_0) = \mathcal{N}(x_0 \,|\, 0, \frac{\phi^2}{1-\sigma^2})$ and parameters $\theta = (\phi, \sigma, \tau)$.

For the SVM, the transition and emission distributions are log-concave in x, allowing Theorem 3 to apply. In the Supplement (Aicher et al., 2023), we show that the prior kernels $\{\vec{\Psi}_t^{(0)}, \overleftarrow{\Psi}_t^{(0)}\}$ of the SVM are bounded with the Lipschitz constant $L_\theta = |\phi|$. Thus, the buffering error decays geometrically with increasing buffer size B when $|\phi| < 1$.

GARCH Model

We finally consider a GARCH(1,1) model (with noise)

$$\begin{aligned} &X_t \,|\, (X_{t-1} = x_{t-1}, \sigma_t^2, \theta) \sim \mathcal{N}(x_t \,|\, 0, \, \sigma_t^2), \\ &\sigma_t^2(x_{t-1}, \sigma_{t-1}^2, \theta) = \alpha + \beta x_{t-1}^2 + \gamma \sigma_{t-1}^2, \\ &Y_t \,|\, (X_t = x_t, \theta) \sim \mathcal{N}(y_t \,|\, x_t\,,\, \tau^2), \end{aligned} \tag{28}$$

with $\nu_0(x_0) = \mathcal{N}(0, \frac{\alpha}{1-\beta-\gamma})$ and parameters $\theta = (\alpha, \beta, \gamma, \tau)$. Unlike the LGSSM and SVM, the noise between X_t and X_{t-1} is multiplicative in X_{t-1} rather than additive. This model's transition distribution is *not* log-concave in (x_t, x_{t-1}) and therefore our theory (Theorem 3) does not hold. However, we see empirically that buffering can help reduce the gradient error for the GARCH in the experiments below and in the Supplement (Aicher et al., 2023).

5.2 Stochastic Gradient Bias

We compare the error of stochastic gradient estimates using a buffered subsequence with $S = 16$, while varying B and N on synthetic data from each model. We generated synthetic data of length $T = 256$ using $(\phi = 0.9, \sigma = 0.7, \tau = 1.0)$ for the LGSSM, $(\phi = 0.9, \sigma = 0.5, \tau = 0.5)$ for the SVM, and $(\alpha = 0.1, \beta = 0.8, \gamma = 0.05, \tau = 0.3)$ for the GARCH model.

Figures 2-4 display the bias of our particle buffered stochastic gradient $g_\theta^{\mathrm{PF}}(S, B, N)$ and g_θ averaged over 1000 replications. We evaluate the gradients at θ equal to the data generating parameters. We vary the buffer size $B \in [0, 16]$, the subsequence size

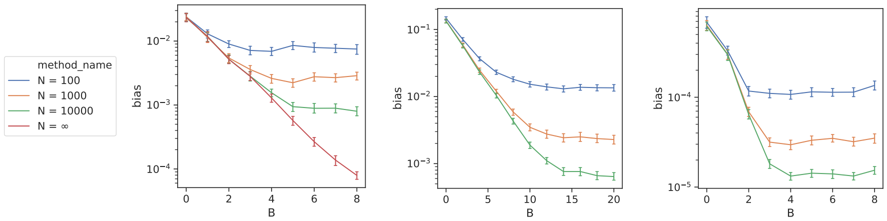


Figure 2: Stochastic gradient bias varying buffer size B for $S = 16$ for different values of N. (left) LGSSM ϕ, (middle) SVM ϕ, (right) GARCH β. Error bars are 95% confidence interval over 1000 replications.

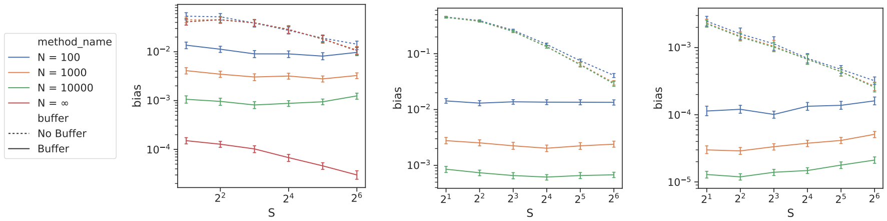


Figure 3: Stochastic gradient bias varying subsequence size S for No Buffer ($B = 0$) and Buffer ($B > 0$) for different values of N. (left) LGSSM ϕ, (middle) SVM ϕ, (right) GARCH β. The buffer size $B = 8$ for LGSSM and GARCH and $B = 16$ for the SVM. Error bars are 95% confidence interval over 1000 replications.

$S \in [1, T]$ and the number of samples $N \in \{100, 1000, 10000\}$. For the LGSSM, we also consider $N = \infty$, by calculating $g_\theta^{\mathrm{PF}}(S, B, \infty)$ using the Kalman filter (Kalman, 1960), which is tractable in the linear setting. We calculate g_θ using the Kalman filter for the LGSSM, and use $g_\theta \approx g_\theta^{\mathrm{PF}}(T, 0, 10^7)$ for the SVM and the GARCH model, assuming that $N = 10^7$ particles is sufficient for an accurate approximation in these 1-dimensional settings.

Figure 2 shows the bias as we vary the buffer size B for different N and $S = 16$. From Figure 2, we see the trade-off between the buffering error (II) and the particle error (III) in the bias bound, (18) of Theorem 2. For all N, when B is small, the buffering error (II) dominates, and therefore the MSE decays exponentially as B increases. However for $N < \infty$, the particle error (III) dominates for larger values of B. In fact, the bias slightly increases due to particle degeneracy, as $|\mathcal{S}^*| = S + 2B$ increases with B. For $N = \infty$ in the LGSSM case, we see that the bias continues to decreases exponentially with large B as there is no particle filter error when using the Kalman filter.

Figure 3 shows the bias as we vary the subsequence size S for different N and with and without buffering. We see that buffering helps regardless of subsequence size (as the bias for all buffered methods are lower than the no buffer methods for all $S \in [2, 64]$). We also see that increasing S can increase the bias for fixed N (when buffering) as the particle error (III) dominates.

Figure 4 shows the bias as we vary the number of particles N for the four different

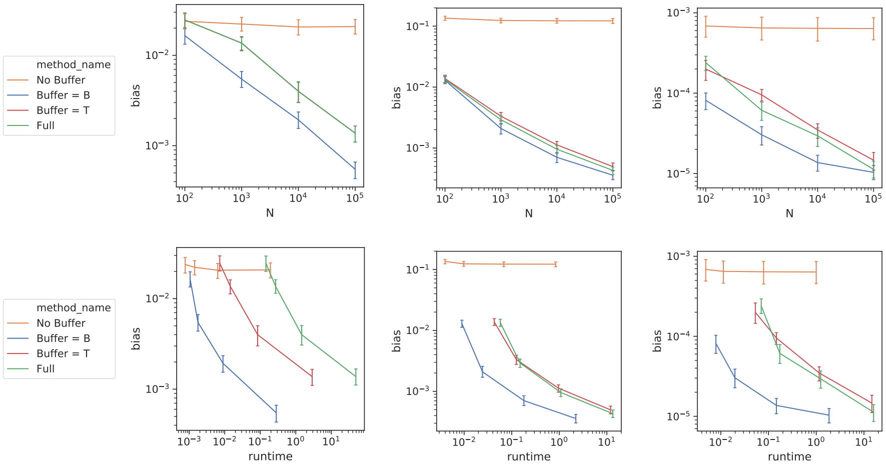


Figure 4: Stochastic gradient bias varying N for different S, B. (left) LGSSM ϕ, (middle) SVM ϕ, (right) GARCH β. (top) x-axis is N, (bottom) x-axis is runtime in seconds. **No Buffer** is $g^{\mathrm{PF}}(16, 0, N)$, **Buffer** $B = B$ is $g^{\mathrm{PF}}(16, B, N)$, **Buffer** $B = T$ is $g^{\mathrm{PF}}(16, T, N)$, and **Full** is $g^{\mathrm{PF}}(T, T, N)$. The moderate buffer size $B = 8$ for LGSSM and GARCH and $B = 16$ for the SVM. Error bars are 95% confidence interval over 1000 replications.

methods correspond to Table 1. In the top row, we compare the bias against N and in the bottom row, we compare the bias against the runtime required to calculate g_θ^{PF}. We see that the method without buffering (orange) is significantly biased regardless of N, where as buffering with moderate B (blue), buffering with large $B = T$ (red), and using the full sequence (green) have similar (lower) bias as we increase N. However the runtime plots show that buffering with moderate B takes significantly less time.

In summary, Figures 2-4 show that buffering cannot be ignored in these three example models: there is high bias for $B = 0$. In general, buffering has diminishing returns when B is excessively large relative to N.

In the Supplement (Aicher et al., 2023), we present plots of the bias varying B, S, N using different particle filters (PaRIS and Poyiadjis N^2) instead of the naive PF. We find that they perform similarly to the naive PF for the small subsequence lengths $|\mathcal{S}^*|$ considered, while taking ≈ 10 times longer to run. We also present plots of the bias as we vary the parameters of the data generating model. We find that as the parameters become more challenging (e.g. $L_\theta \to 1$), we need to increase both B and N to control bias; otherwise, the buffer stochastic subsequence methods are more biased than using full sequence gradient.

5.3 SGLD Experiments

Having examined the stochastic gradient bias, we now examine using our buffered stochastic gradient estimators in SGLD (Algorithm 2).

SGLD Evaluation Method

We measure the sample quality of our MCMC chains $\{\theta^{(k)}\}_{k=1}^{K}$ using the *kernel Stein discrepancy* (KSD) for equal compute time (Gorham and Mackey, 2017; Liu et al., 2016). We choose to use KSD rather than classic MCMC diagnostics such as effective sample size (ESS) (Gelman et al., 2013), because KSD penalizes the bias present in our MCMC chains. Whilst it can be hard to interpret the absolute value of KSD for any problem, it is informative for comparing between different algorithms. Given a sample chain (after burnin and thinning) $\{\theta^{(k)}\}_{k=1}^{\tilde{K}}$, let $\hat{p}(\theta|y)$ be the empirical distribution of the samples. Then the KSD between $\hat{p}(\theta|y)$ and the posterior distribution $p(\theta|y)$ is

$$\mathrm{KSD}(\hat{p}, p) = \sum_{d=1}^{\dim(\theta)} \sqrt{\sum_{k,k'=1}^{\tilde{K}} \frac{\mathcal{K}_0^d(\theta^{(k)}, \theta^{(k')})}{\tilde{K}^2}}, \tag{29}$$

where

$$\mathcal{K}_0^d(\theta, \theta') = \tfrac{1}{p(\theta|y)p(\theta'|y)} \nabla_{\theta_d} \nabla_{\theta'_d} (p(\theta|y)\mathcal{K}(\theta, \theta')p(\theta'|y)) \tag{30}$$

and $\mathcal{K}(\cdot, \cdot)$ is a valid kernel function. Following Gorham and Mackey (2017), we use the inverse multiquadratic kernel $\mathcal{K}(\theta, \theta') = (1 + \|\theta - \theta'\|_2^2)^{-0.5}$ in our experiments. Since (30) requires full gradient evaluations of $\log p(\theta|y)$ that are computationally intractable, we replace these terms with corresponding stochastic estimates using the full particle filter estimate, g_θ^{PF} (Gorham et al., 2020).

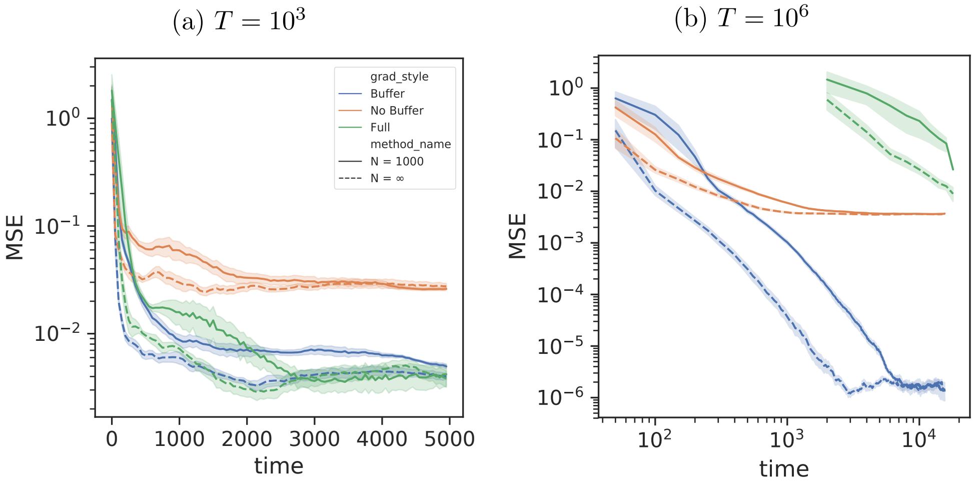
SGLD on Synthetic LGSSM Data

To assess the effect of using particle filters with buffered stochastic gradients, we first focus on SGLD on synthetic LGSSM data, where calculating $\widehat{g}_\theta(S, B)$ is possible. We generate training sequences of length $T = 10^3$ or 10^6 using the same parametrization as Section 5.2.

We consider three pairs of different gradient estimators: **Full** ($S = T$), **Buffered** ($S = 40, B = 10$) and **No Buffer** ($S = 40, B = 0$) each with $N = 1000$ particles using the particle filter and with $N = \infty$ using the Kalman filter. To select the stepsize, we performed a grid search over $\epsilon \in \{1, 0.1, 0.01, 0.001\}$ and selected the method with smallest KSD to the posterior on the training set. We present the KSD results (for the best ϵ) in Table 2 and trace plots of the metrics in Figure 5.

From Figure 5, we see that the methods without buffering ($B = 0$) have higher MSE as they are biased. We also see that the full sequence methods ($S = T$) perform poorly for large $T = 10^6$.

The KSD results further support this story. Table 2 presents the mean and standard deviation on our estimated $\log_{10}$ KSD for θ. Tables of the marginal KSD for individual components of θ can be found in the Supplement (Aicher et al., 2023). The methods without buffering have larger KSD, as the inherent bias of $\widehat{g}_\theta(S, B = 0)$ led to an incorrect stationary distribution. The full sequence methods perform poorly for $T = 10^6$ because of a lack of samples that can be computed in a fixed runtime.


(a) $T = 10^3$

(b) $T = 10^6$

Figure 5: Comparison of SGLD with different gradient estimates on synthetic LGSSM data: $T = 10^3$ (left), $T = 10^6$ (right). MSE of estimated posterior mean to true $\phi = 0.9$.

In the Supplement (Aicher et al., 2023), we present similar results on synthetic SVM and GARCH data. Also in the Supplement (Aicher et al., 2023), we present results on LGSSM in higher dimensions. As is typical in the particle filtering literature, the performance degrades with increasing dimensions for N fixed.

SGLD on Exchange Rate Log-Returns

We now consider fitting the SVM and the GARCH model to EUR-USD exchange rate data at the minute resolution from November 2017 to October 2018. The data consists of 350,000 observations of demeaned log-returns. As the market is closed during non-business hours, we further break the data into 53 weekly segments of roughly 7,000 observations each. In our model, we assume independence between weekly segments and divide the data into a training set of the first 45 weeks and a test set of the last 8 weeks. Full processing details and example plots are in the Supplement (Aicher et al., 2023). Our method (Algorithm 2) easily scales to the unsegmented series; however the abrupt changes between starts of weeks are not adequately modeled by (27)

We fit both the SVM and the GARCH model using SGLD with four different gradient methods: (i) **Full**, the full gradient over all segments in the training set; (ii) **Weekly**, a stochastic gradient over a randomly selected segment in the training set; (iii) **No Buffer**, a stochastic gradient over a randomly selected *subsequence* of length $S = 40$; and (iv) **Buffer**, our buffered stochastic gradient for a subsequence of length $S = 40$ with buffer length $B = 10$. To estimate the stochastic gradients, we use Algorithm 1 with $N = 1000$. To select the stepsize parameter, we performed a grid search over $\epsilon \in \{1, 0.1, 0.01, 0.001\}$ and selected the method with smallest KSD. We present the KSD results in Table 3.

Table 2: KSD for Synthetic LGSSM. Mean and SD. Results are shown after running each method for a fixed computational time.

			$\log_{10}$KSD	
S	B	N	$T = 10^3$	$T = 10^6$
T	–	1000	0.85 (0.08)	4.92 (0.40)
		∞	**0.64 (0.17)**	4.85 (0.36)
40	0	1000	1.58 (0.03)	4.68 (0.10)
		∞	1.55 (0.03)	4.68 (0.11)
40	10	1000	**0.68 (0.25)**	**3.43 (0.19)**
		∞	**0.61 (0.21)**	**3.25 (0.29)**

Table 3: KSD for SGLD on exchange rate data. Mean and SD over 5 chains each. Results are shown after running each method for a fixed computational time.

	$\log_{10}$KSD	
Method	SVM	GARCH
Full	4.03 (0.14)	2.84 (0.30)
Weekly	3.87 (0.08)	2.81 (0.21)
No Buffer	4.48 (0.01)	**2.09 (0.09)**
Buffer	**3.56 (0.08)**	**2.19 (0.05)**

For the SVM, we see that buffering leads to more accurate MCMC samples, Table 3 (left). In particular, the samples from SGLD without buffering have smaller ϕ, τ^2 and a larger σ^2, indicating that its posterior is (inaccurately) centered around a SVM with larger latent state noise. We also again see that the full sequence and weekly segment methods perform poorly due to the limited number of samples that can be computed in a fixed runtime.

For the GARCH model, Table 3 (right), we see that the subsequence methods out perform the full sequence methods, but unlike in the SVM, buffering does not help with inference on the GARCH data. This is because the GARCH model that we recover on the exchange rate data (for all gradient methods) is close to white noise $\beta \approx 0$. Therefore the model believes the observations are close to independent, hence no buffer is necessary.

6 Discussion

In this work, we developed a particle buffered stochastic gradient estimators for nonlinear SSMs. Our key contributions are (i) extending buffered stochastic gradient MCMC with particle filtering for nonlinear SSMs, (ii) analyzing the error of our proposed particle buffered stochastic gradient g_θ^{PF} (Theorem 2) and its affect on our SGLD Algorithm 2 (Theorem 1), and (iii) generalizing the geometric decay bound for buffering to nonlinear SSMs with log-concave likelihoods (Theorem 3). We evaluated our proposed gradient estimator with SGLD on both synthetic data and EUR-USD exchange rate data. We find that buffering is necessary to control bias and that our stochastic gradient methods (Algorithm 2) are able to out perform batch methods on long sequences.

Possible future extensions of this work include relaxing the log-concave restriction of Theorem 3, extensions to Algorithm 2 as discussed at the end of Section 3.2, and applying our particle buffered stochastic gradient estimates to other applications than SGMCMC, such as maximising loglikelihoods or optimization in variational autoencoders for sequential data (Maddison et al., 2017; Naesseth et al., 2018).

Supplementary Material

See the Supplement (Aicher et al., 2023) for all additional material. In Supplement A, we provide additional details and proofs for the error analysis of Section 4. In particular, we provide the proof of Theorem 1 in Supplement A.1, the proof of Theorem 2 in Supplement A.2, the proof of Theorem 3 in Supplement A.3 and applications of Theorem 3 for LGSSM and SVM in Supplement A.4. In Supplement B, we provide additional particle filter and gradient details for the models in Section 5.1. In Supplement C, we provide additional details and figures of experiments.

References

Aicher, C., Ma, Y.-A., Foti, N. J., and Fox, E. B. (2019). "Stochastic Gradient MCMC for State Space Models." *SIAM Journal on Mathematics of Data Science*, 1(3): 555–587. 2, 6, 8

Aicher, C., Putcha, S., Nemeth, C., Fearnhead, P., and Fox, E. B. (2023). "Supplement to 'Stochastic Gradient MCMC for Nonlinear State Space Models.'." 9, 10, 12, 13, 15, 16, 17, 19

Andrieu, C., Doucet, A., and Holenstein, R. (2010). "Particle Markov chain Monte Carlo methods." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342. 4

Baker, J., Fearnhead, P., Fox, E. B., and Nemeth, C. (2019). "Control variates for stochastic gradient MCMC." *Statistics and Computing*, 29(3): 599–615. 5

Bollerslev, T. (1986). "Generalized autoregressive conditional heteroskedasticity." *Journal of Econometrics*, 31(3): 307–327. 12

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer. 3

Chatterji, N. S., Flammarion, N., Ma, Y.-A., Bartlett, P. L., and Jordan, M. I. (2018). "On the Theory of Variance Reduction for Stochastic Gradient Monte Carlo." In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 764–773. PMLR. 5

Chen, C., Ding, N., and Carin, L. (2015). "On the Convergence of Stochastic Gradient

MCMC Algorithms with High-Order Integrators." In *Advances in Neural Information Processing Systems*, volume 28, 2278–2286. 8, 9

Chen, T., Fox, E., and Guestrin, C. (2014). "Stochastic Gradient Hamiltonian Monte Carlo." In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 1683–1691. PMLR. 5

Colombo, M., Figalli, A., and Jhaveri, Y. (2017). "Lipschitz changes of variables between perturbations of log-concave measures." *Annali Scuola Normale Superiore - Classe Di Scienze*, 17(4): 1491–1519. 12

Dahlin, J., Lindsten, F., and Schön, T. B. (2015). "Particle Metropolis–Hastings using gradient and Hessian information." *Statistics and Computing*, 25(1): 81–92. 4

Dalalyan, A. S. and Karagulyan, A. G. (2019). "User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient." *Stochastic Processes and their Applications*, 129(12): 5278–5311. 5, 9

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. (2014). "Bayesian Sampling Using Stochastic Gradient Thermostats." In *Advances in Neural Information Processing Systems*, volume 27, 3203–3211. 5

Doucet, A. and Johansen, A. M. (2009). "A tutorial on particle filtering and smoothing: Fifteen years later." *Handbook of Nonlinear Filtering*, 12(3): 656–704. 3, 4

Dukic, V., Lopes, H. F., and Polson, N. G. (2012). "Tracking Epidemics With Google Flu Trends Data and a State-Space SEIR Model." *Journal of the American Statistical Association*, 107(500): 1410–1426. 1

Fearnhead, P. and Künsch, H. R. (2018). "Particle Filters and Data Assimilation." *Annual Review of Statistics and Its Application*, 5: 421–449. 3

Gelman, A., Carlin, J. B., Rubin, D. B., Vehtari, A., Dunson, D. B., and Stern, H. S. (2013). *Bayesian Data Analysis*. CRC Press, third edition. 16

Girolami, M. and Calderhead, B. (2011). "Riemann manifold Langevin and Hamiltonian Monte Carlo methods." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2): 123–214. 5

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). "Novel approach to nonlinear/non-Gaussian Bayesian state estimation." *IEE Proceedings F - Radar and Signal Processing*, 140(2): 107–113. 1, 4

Gorham, J. and Mackey, L. (2017). "Measuring Sample Quality with Kernels." In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1292–1301. PMLR. 16

Gorham, J., Raj, A., and Mackey, L. (2020). "Stochastic Stein Discrepancies." In *Advances in Neural Information Processing Systems*, volume 33, 17931–17942. 16

Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems." *ASME Journal of Basic Engineering*, 82: 35–45. 14

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. (2015). "On Particle Methods for Parameter Estimation in State-Space Models." *Statistical Science*, 30(3): 328–351. 2, 10

Kitagawa, G. (1996). "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models." *Journal of Computational and Graphical Statistics*, 5(1): 1–25. 4

Kitagawa, G. and Sato, S. (2001). "Monte Carlo Smoothing and Self-Organising State-Space Model." In *Sequential Monte Carlo Methods in Practice*, 177–195. Springer New York. 4

Liu, J. S. and Chen, R. (1998). "Sequential Monte Carlo Methods for Dynamic Systems." *Journal of the American Statistical Association*, 93(443): 1032–1044. 4

Liu, Q., Lee, J., and Jordan, M. (2016). "A Kernelized Stein Discrepancy for Goodness-of-fit Tests." In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, 276–284. PMLR. 16

Ma, Y.-A., Chen, T., and Fox, E. (2015). "A Complete Recipe for Stochastic Gradient MCMC." In *Advances in Neural Information Processing Systems*, volume 28, 2917–2925. 2, 5

Ma, Y.-A., Foti, N. J., and Fox, E. B. (2017). "Stochastic Gradient MCMC Methods for Hidden Markov Models." In *Proceedings of the 34th International Conference on Machine Learning*, 2265–2274. PMLR. 2, 6

Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. (2017). "Filtering Variational Objectives." In *Advances in Neural Information Processing Systems*, volume 30, 6573–6583. 19

Naesseth, C., Linderman, S., Ranganath, R., and Blei, D. (2018). "Variational Sequential Monte Carlo." In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, 968–977. PMLR. 19

Nagapetyan, T., Duncan, A. B., Hasenclever, L., Vollmer, S. J., Szpruch, L., and Zygalakis, K. (2017). "The true cost of stochastic gradient Langevin dynamics." *arXiv preprint arXiv:1706.02692*. 5

Nemeth, C. and Fearnhead, P. (2021). "Stochastic gradient Markov chain Monte Carlo." *Journal of the American Statistical Association*, 116(533): 433–450. 5

Nemeth, C., Fearnhead, P., and Mihaylova, L. (2016). "Particle Approximations of the Score and Observed Information Matrix for Parameter Estimation in State–Space Models With Linear Computational Cost." *Journal of Computational and Graphical Statistics*, 25(4): 1138–1157. 4

Olsson, J. and Westerborn, J. (2017). "Efficient particle-based online smoothing in general hidden Markov models: The PaRIS algorithm." *Bernoulli*, 23(3): 1951–1996. 4, 10

Ou, R., Young, A. L., and Dunson, D. B. (2018). "Clustering-Enhanced Stochastic

Gradient MCMC for Hidden Markov Models with Rare States." *arXiv preprint arXiv:1810.13431*. 7

Patterson, S. and Teh, Y. W. (2013). "Stochastic Gradient Riemannian Langevin dynamics on the Probability Simplex." In *Advances in Neural Information Processing Systems*, volume 26, 3102–3110. 5

Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). "Particle approximations of the score and observed information matrix in state space models with application to parameter estimation." *Biometrika*, 98(1): 65–80. 4

Robbins, H. and Monro, S. (1951). "A stochastic approximation method." *The Annals of Mathematical Statistics*, 400–407. 3

Schmidt, M., Babanezhad, R., Ahmed, M., Defazio, A., Clifton, A., and Sarkar, A. (2015). "Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields." In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38, 819–828. PMLR. 7

Shephard, N. (2005). *Stochastic Volatility: Selected Readings*. Oxford University Press. 1, 12

Teh, Y. W., Thiery, A. H., and Vollmer, S. J. (2016). "Consistency and Fluctuations For Stochastic Gradient Langevin Dynamics." *Journal of Machine Learning Research*, 17(7): 1–33. 5

Villani, C. (2008). *Optimal Transport: Old and New*, volume 338 of *A Series of Comprehensive Studies in Mathematics*. Springer Science & Business Media, first edition. 12

Vollmer, S. J., Zygalakis, K. C., and Teh, Y. W. (2016). "Exploration of the (Non-) Asymptotic Bias and Variance of Stochastic Gradient Langevin Dynamics." *Journal of Machine Learning Research*, 17(159): 1–48. 8

Welling, M. and Teh, Y. W. (2011). "Bayesian Learning via Stochastic Gradient Langevin Dynamics." In *Proceedings of the 28th International Conference on Machine Learning*, 681–688. 2, 5

Acknowledgments

We would like to thank Nicholas Foti for helpful discussions. This work was supported in part by: ONR Grants N00014-15-1-2380, N00014-18-1-2862, and N00014-22-1-2110; NSF CAREER Award IIS-1350133; AFOSR Grant FA9550-21-1-0397; and, EPSRC Grants EP/L015692/1, EP/S00159X/1, EP/V022636/1, EP/R01860X/1, EP/R018561/1 and EP/R034710/1.