# Stochastic Gradient MCMC for Nonlinear State Space Models

Christopher Aicher[1], Srshti Putcha[2], Christopher Nemeth[3], Paul Fearnhead[3], and Emily B. Fox[1,4]

[1]Department of Statistics, University of Washington, Seattle, WA
[2]STOR-i Centre for Doctoral Training, Lancaster University, Lancaster, UK
[3]Department of Mathematics and Statistics, Lancaster University, Lancaster, UK
[4]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA

**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 inference for hidden Markov models (HMMs) 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 SGMCMC for inference on both long sequential synthetic and minute-resolution financial returns data, demonstrating the importance of this class of methods.

## 1 Introduction

Nonlinear *state space models* (SSMs) are widely used in many scientific domains for modeling time series and sequential data. 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 or *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 pass through the data scales linearly with the length of the sequence, and (ii) the number of particles (and hence the computation per data point) required to control the variance of the particle filter scales with the length of the sequence.

Stochastic gradient Markov chain Monte Carlo (SGMCMC) is a popular method for scaling Bayesian inference to large data sets, which replace full data gradients with stochastic gradient estimates based on subsets of data [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., 2018]. To correct for this, Ma et al. [2017] and Aicher et al. [2018] have developed

*buffered* stochastic gradient estimators that control the bias. The latent state sequence is marginalized in a buffer around each subsequence, allowing fewer dependencies to be broken. However, the work so far has been limited to SSMs where analytic marginalization is possible (e.g. HMMs and linear dynamical systems).

In this work, we propose *particle buffered* gradient estimators that generalize the buffered gradient estimators to nonlinear SSMs. In particular, we show how buffering in nonlinear SSMs can be approximated with a modified particle filter. Beyond the regular speedup gains from using a subsequence over a batch, our method also reduces the number of particles required to control the variance of the particle filter. We provide an error analysis of our proposed estimators by decomposing the error into buffering error and particle filter error. We also extend the buffering error bounds of Aicher et al. [2018] 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.

This paper is organized as follows. First, we review background on particle filtering in nonlinear SSMs and SGMCMC for analytic SSMs in Section 2. We then present our particle buffered stochastic gradient estimator and its error analysis in Section 3. Finally, we test our estimator for nonlinear SSMs on both synthetic and EUR-US exchange rate data in Section 4.

# 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, $Y_t$, are therefore 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 $p(x_t|x_{t-1}, \theta)$ is the *transition density* and $p(y_t|x_t, \theta)$ is the *emission density*.

Examples of nonlinear SSMs include the *stochastic volatility model* (SVM) [Shephard, 2005] or the *generalized autoregressive conditional heteroskedasticity* (GARCH) model [Bollerslev, 1986]. For a review of applications of state space modeling, see Langrock [2011].

For an arbitrary sequence $\{a_i\}$, we use $a_{i:j}$ to denote the sequence $(a_i, a_{i+1}, \ldots, a_j)$. To infer the model parameters $\theta$, a quantity of interest is the *score function*, or the gradient of the marginal loglikelihood, $\nabla_\theta \log p(y_{1:T}|\theta)$. Using the score function, the loglikelihood can for instance 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],

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

However, 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.

### 2.1.1 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* (SIR) [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,

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

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 $p(x_t|x_{t-1},\theta)$, SIR is also known as the *bootstrap particle filter* [Gordon et al., 1993]. By using the transition density for proposals, the importance weight recursion in Eq. (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 $h_t(x_t,x_{t-1}) = \nabla_\theta \log p(y_t,x_t \,|\, x_{t-1},\theta)$ – then we only need to keep track of the partial sum $H_t = \sum_{s=1}^t h_s(x_s,x_{s-1})$ rather than the full list of $x_{1:t}$ during SIR. The complete particle filtering scheme is detailed in Algorithm 1.

**Algorithm 1** Particle Filter

**Input:** number of particles, $N$, pairwise statistics, $h_{1:T}$, observations $y_{1:T}$, proposal density $q$,
Draw $x_0^{(i)} \sim \nu(x_0|\theta)$, set $w_0^{(i)} = \frac{1}{N}$, and $H_0^{(i)} = 0\ \forall i$.
**for** $t = 1,\ldots,T$ **do**
  Resample ancestor indices $\{a_1,\ldots,a_N\}$.
  Propagate particles $x_t^{(i)} \sim q(\cdot|x_{t-1}^{(a_i)},y_t,\theta)$.
  Update each $w_t^{(i)}$ according to Eq. (3).
  Update statistics $H_t^{(i)} = H_{t-1}^{(a_i)} + h_t(x_t^{(i)},x_{t-1}^{(a_i)})$.
**end for**
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 et al. [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 $\theta$ 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.

A fundamental method within the SGMCMC family is the *stochastic gradient Langevin dynamics* (SGLD) algorithm [Welling and Teh, 2011]:

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

where $\epsilon^{(k)}$ is the stepsize. 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 [2017] provide non-asymptotic bounds on Wasserstein distance to the posterior after $K$ steps of SGLD 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 $\widehat{g}_\theta$ [Nagapetyan et al., 2017, Baker et al., 2018, Chatterji et al., 2018] and augmented dynamics to improve mixing [Ma et al., 2015] such as SGHMC [Chen et al., 2014], SGNHT [Ding et al., 2014], and SGRLD [Girolami and Calderhead, 2011, Patterson and Teh, 2013].

### 2.2.1 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 Eq. (2) is

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

Although Eq. (5) requires only 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 Eq. (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, Eq. (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 reduce the bias in stochastic gradients while avoiding accessing the full sequence, previous work on SGMCMC for SSMs proposed *buffered* stochastic gradients [Ma et al., 2017, Aicher et al., 2018][1]:

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

[1] Previous work on inference in general SSMs has shown that the Markov chain displays a forgetting property (see Chapter 3 of Cappé et al. [2005]). Therefore, conditional on the current value of $t$, it is sensible to use buffering, as we expect distant time points to have negligible impact.

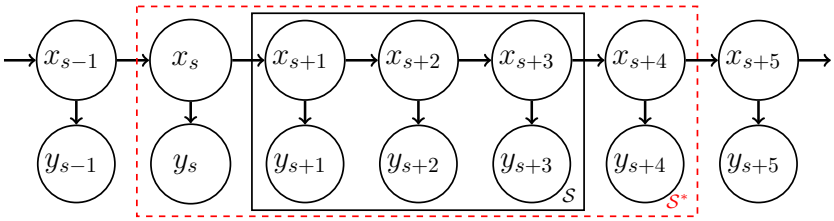


Figure 1: Graphical model of $\mathcal{S}^*$ with $S = 3$ and $B = 1$.

where $\mathcal{S}^* = \{s + 1 - B, \ldots, s + S + B\}$ is the *buffered* subsequence for $\mathcal{S}$ (see Figure 1). Note Eq. (5) is $\widehat{g}(S, T)$ and Eq. (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 SSM model and gradient both satisfy a Lipschitz property, the bias decays geometrically in buffer size $B$ (see Theorem 4.1 of Aicher et al. [2018]). Specifically,

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

where $L_\theta$ is a bound for the Lipschitz constants of the *forward and backward smoothing kernels*[2]

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

The bound provided in Eq. (8) ensures that only a modest buffer size $B$ is required (e.g. $\mathcal{O}(\log \delta^{-1})$ for an accuracy of $\delta$). 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 Eq. (7). In addition, we extend the error bounds of Aicher et al. [2018] to the nonlinear SSM case, guaranteeing that the error decays geometrically in $B$, without requiring an explicit form for the smoothing kernels. We also analyze the approximation error by decomposing the buffering error and the particle filter error.

## 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 Eq. (7) using Algorithm 1. In particular, the complete data loglikelihood, $\log p(y_{\mathcal{S}}, x_{\mathcal{S}}, \theta)$, in Eq. (7) decomposes into a sum of pairwise statistics $H = \sum_{t \in \mathcal{S}^*} h_t(x_t, x_{t-1})$ where

$$h_t(x_t, x_{t-1}) = \begin{cases} \dfrac{\nabla_\theta \log p(x_t, y_t \,|\, x_{t-1}, \theta)}{\Pr(t \in \mathcal{S})} & \text{if } t \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

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 Eq. (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 $S + 2B \ll T$.

[2]We follow Aicher et al. [2018] and consider Lipschitz constant for a kernel $\Psi$ is measured in terms of the $p$-Wasserstein distance between distributions of $x, x'$ and $\Psi(x), \Psi(x')$. See the Supplement for additional details.

### 3.2 SGMCMC Algorithm

Using $g_\theta^{\mathrm{PF}}(S, B, N)$ as our stochastic gradient estimate in SGLD, Eq. (4), gives us Algorithm 2.[3]

**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 $g_\theta^{\mathrm{PF}}$ using Algorithm 1 on Eq. (10).
  Set $\theta^{(k+1)} \leftarrow \theta^{(k)} + \epsilon \cdot g_\theta^{\mathrm{PF}} + \mathcal{N}(0, 2\epsilon)$
**end for**
Return $\theta^{(K+1)}$

Algorithm 2 can be extended and improved 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 2.1.1, and (iii) using more advanced SGMCMC schemes such as those listed in Section 2.2.

### 3.3 Error Analysis

Although defining the particle variant of SGMCMC is relatively straightforward by building on Aicher et al. [2018], the error analysis presents new challenges. To analyze the error of the SGMCMC sampler, it is sufficient to bound the bias and variance of our stochastic gradient estimator to the exact full-data gradient [Dalalyan and Karagulyan, 2017]. We link the error between the full gradient $g_\theta$ and $g_\theta^{\mathrm{PF}}(S, B, N)$ through $\widehat{g}_\theta(S, B)$ and $\widehat{g}_\theta(S, T)$,

$$g_\theta \;\Leftrightarrow\; \widehat{g}_\theta(S,T) \;\Leftrightarrow\; \widehat{g}_\theta(S,B) \;\Leftrightarrow\; g_\theta^{\mathrm{PF}}(S,B,N). \tag{11}$$

Therefore there are three error sources to consider in (11)

(I) *Subsequence Error*, $g_\theta \;\Leftrightarrow\; \widehat{g}_\theta(S,T)$: the error in approximating Fisher's identity with a stochastic subsequence. The error in this term follows the standard stochastic gradient literature, which depends on the subsequence length $S$ and how subsequences are sampled. For a random minibatch of size $S$ sampling without replacement, the variance scales $\mathcal{O}(1/S)$; However, for a random *contiguous* subsequences of size $S$, the variance scales $\mathcal{O}(\frac{1+\rho}{S(1-\rho)})$ where $\rho$ is a bound on the autocorrelation between terms (see the Supplement for details).

(II) *Buffering Error*, $\widehat{g}_\theta(S,T) \;\Leftrightarrow\; \widehat{g}_\theta(S,B)$: the error in approximating the latent state posterior $p(x_{1:T} \,|\, y_{1:T})$ with $p(x_{1:T} \,|\, y^*_{\mathcal{S}})$. If the smoothing kernels $\{\vec{\Psi}_t, \overleftarrow{\Psi}_t\}$ are contractions for all $t$ (i.e. $L_\theta < 1$), then from Eq. (8) the error in this term scales as $\mathcal{O}((L_\theta)^B/S)$ [Aicher et al., 2018]. In Section 3.3.1, we show sufficient conditions for $L_\theta < 1$.

(III) *Particle Error*, $\widehat{g}_\theta(S,B) \;\Leftrightarrow\; g_\theta^{\mathrm{PF}}(S,B,N)$: the error from the particle smoother Monte-Carlo approximation. This error depends on the number of particles $N$ and the length of sequence $|\mathcal{S}^*| = S + 2B$. For the particle filter, Algorithm 1, the asymptotic variance in this term scales as $\mathcal{O}((S+2B)^2/N)$ [Poyiadjis et al., 2011].

The error term (I-III) that dominates depends on the regime $(S, B, N)$. For example, increasing $B$, decreases the error in term (II), but increases the error in term (III); therefore, increasing $B$ to reduce buffering bias will not be effective if $N$ is not sufficiently large to avoid particle degeneracy.

[3] Python code for Algorithm 2 and experiments of Section 4 is available at `https://github.com/aicherc/sgmcmc_ssm_code`.

### 3.3.1 Buffering Error for Nonlinear SSMs

To obtain a bound for the buffering error term (II), we require the Lipschitz constant $L_\theta$ 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. Instead, we show that we 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* $\vec{\Psi}_t^{(1)}, \overleftarrow{\Psi}_t^{(1)}$

$$\begin{aligned} \vec{\Psi}_t^{(0)} &:= p(x_t \,|\, x_{t-1}, \theta) & \vec{\Psi}_t^{(1)} &:= p(x_t \,|\, x_{t-1}, y_t, \theta), \\ \overleftarrow{\Psi}_t^{(0)} &:= p(x_t \,|\, x_{t+1}, \theta) & \overleftarrow{\Psi}_t^{(1)} &:= p(x_t \,|\, x_{t+1}, y_t, \theta). \end{aligned} \tag{12}$$

The prior kernels $\vec{\Psi}_t^{(0)}, \overleftarrow{\Psi}_t^{(0)}$ are defined by the model and therefore usually available. When $\vec{\Psi}_t^{(1)}, \overleftarrow{\Psi}_t^{(1)}$ are also available, they can be used to obtain even tighter bounds.

We now present our results for the forward kernels $\vec{\Psi}_t$; similar arguments can be made for the backward kernels $\overleftarrow{\Psi}_t$. These results rely on the transition and emission densities being *log-concave* in $x_t, x_{t-1}$.

**Theorem 1** (Lipschitz Bound for Log-Concave Models)**.** *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*

$$\|\vec{\Psi}_t\|_{Lip} \;\leq\; \|\vec{\Psi}_t^{(1)}\|_{Lip} \;\leq\; \|\vec{\Psi}_t^{(0)}\|_{Lip}. \tag{13}$$

This theorem lets us bound $L_\theta$ with the Lipschitz constant of either the prior kernels or filtered kernels. The proof of Theorem 1 is provided in the Supplement and is based on Caffarelli's log-concave perturbation theorem [Villani, 2008, Colombo et al., 2015]. Examples of SSMs that are log-concave include the LGSSM, the stochastic volatility model, or any linear SSM with log-concave transition or emission noise. Examples of SSMs that are not log-concave include the GARCH model or any linear SSM with a transition or emission noise distribution that is not log-concave (e.g. Student's $t$).

# 4 Experiments

We first introduce the three models: (i) linear Gaussian SSM (LGSSM), a case where analytic buffering is possible, to assess the impact of the particle filter; (ii) the SVM, where the emissions are non-Gaussian; and (iii) a GARCH model, where the latent transitions are nonlinear. We then empirically test the gradient error of our particle buffered gradient estimator on synthetic data for fixed $\theta$. Finally, we evaluate the performance of our proposed SGLD algorithm (Algorithm 2) on both real and synthetic data.

## 4.1 Models

### 4.1.1 Linear Gaussian SSM

The *linear Gaussian SSM* (LGSSM) is

$$X_t \,|\, (X_{t-1} = x_{t-1}), \theta \sim \mathcal{N}(x_t \,|\, \phi x_{t-1}\,,\, \sigma^2), \tag{14}$$

$$Y_t \,|\, (X_t = x_t), \theta \sim \mathcal{N}(y_t \,|\, x_t\,,\, \tau^2), \tag{15}$$

with $X_0 \sim \mathcal{N}(x_0 | 0, \frac{\phi^2}{1-\sigma^2})$ and parameters $\theta = (\phi, \sigma, \tau)$.

The transition and emission distributions are both Gaussian and log-concave in $x$, allowing Theorem 1 to apply. In the Supplement, we show that the filtered kernels $\{\vec{\Psi}_t^{(1)}, \overleftarrow{\Psi}_t^{(1)}\}$ 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 Eq. (11) can be calculated analytically.

#### 4.1.2 Stochastic Volatility Model

The *stochastic volatility model* (SVM) is

$$X_t \,|\, (X_{t-1} = x_{t-1}), \theta \sim \mathcal{N}(x_t \,|\, \phi x_{t-1}\,,\, \sigma^2), \tag{16}$$

$$Y_t \,|\, (X_t = x_t), \theta \sim \mathcal{N}(y_t \,|\, 0\,,\, \exp(x_t)\tau^2), \tag{17}$$

with parameters $\theta = (\phi, \sigma, \tau)$.

For the SVM, the transition and emission distributions are log-concave in $x$, allowing Theorem 1 to apply. In the Supplement, 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 $|\phi| < 1$.

#### 4.1.3 GARCH Model

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

$$X_t \,|\, (X_{t-1} = x_{t-1}), \sigma_t^2, \theta \sim \mathcal{N}(x_t \,|\, 0,\, \sigma_t^2), \tag{18}$$

$$\sigma_t^2(x_{t-1}, \sigma_{t-1}^2, \theta) = \alpha + \beta x_{t-1}^2 + \gamma \sigma_{t-1}^2, \tag{19}$$

$$Y_t \,|\, (X_t = x_t), \theta \sim \mathcal{N}(y_t \,|\, x_t\,,\, \tau^2), \tag{20}$$

with 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 is *not* log-concave and therefore our theory (Theorem 1) 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.

### 4.2 Stochastic Gradient Error

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 $T$ 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.

Figure 2 displays the mean squared error (MSE) between our particle buffered stochastic gradient $g_\theta^{\mathrm{PF}}(S, B, N)$ and $\widehat{g}_\theta(S, T)$ averaged over 100,000 replications. We evaluate the gradients at $\theta$ equal to the data generating parameters. We vary the buffer size $B \in [0, 8]$ and the number of samples $N \in \{100, 1000, 10000\}$. For the LGSSM, we also consider $N = \infty$, calculating $g_\theta^{\mathrm{PF}}(S, B, \infty)$ using the Kalman filter, which is tractable in the linear setting. We calculate $\widehat{g}_\theta(S, T)$ using the Kalman filter for the LGSSM, and use $\widehat{g}_\theta(S, T) \approx g_\theta^{\mathrm{PF}}(S, 16, 10^7)$ for the SVM and the GARCH model, assuming that $N = 10^7$ particles and $B = 16$ is sufficient for a highly accurate approximation.

Figure 2 demonstrates the trade-off between the buffering error (II) and the particle error (III) from Section 3.3. 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 MSE 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 error continues to decreases exponentially with large $B$ as there is no particle filter error when using the Kalman filter.

Figure 2 also shows that buffering cannot be ignored in these three example models: there is high MSE for $B = 0$. In general, buffering has diminishing returns when $B$ is excessively large relative to $N$.

### 4.3 SGLD Experiments

Having examined the stochastic gradient error, we now consider using our stochastic gradient estimators in SGLD.

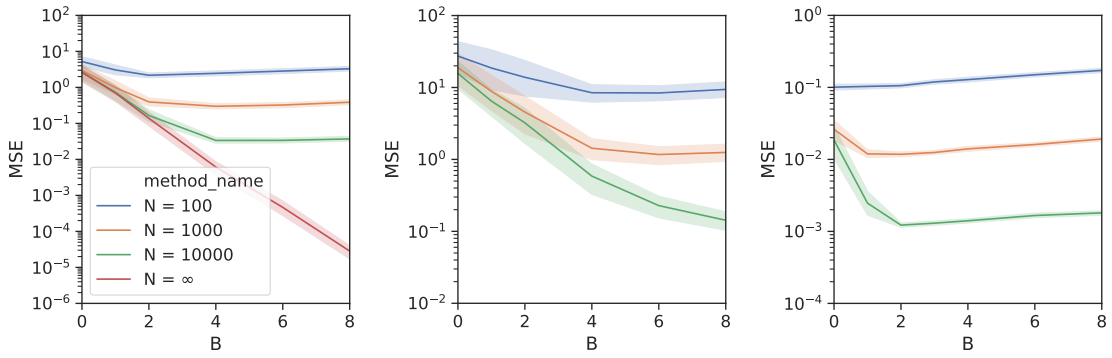


Figure 2: Buffered Stochastic Gradient Estimate Error Plots. (left) LGSSM $\phi$, (middle) SVM $\phi$, (right) GARCH $\beta$

#### 4.3.1 SGLD Evaluation Methods

We assess the performance of our samplers given a fixed computation budget, by measuring both the heldout and predictive loglikelihoods on a test sequence. Given a sampled parameter value $\theta^{(k)}$ the heldout loglikelihood is

$$\sum_{t=1}^{T} \log p(y_t \mid y_{<t}, \theta) \approx \sum_{t=1}^{T} \sum_{i=1}^{N} w_{t-1}^{(i)} \log p(y_t \mid x_{t-1}^{(i)}, \theta), \tag{21}$$

and the $r$-step ahead predictive loglikelihood is

$$\sum_{t=1}^{T} \log p(y_{t+r} \mid y_{<t}, \theta) \approx \sum_{t=1}^{T} \sum_{i=1}^{N} w_{t-1}^{(i)} \log p(y_{t+r} \mid x_{t-1}^{(i)}, \theta), \tag{22}$$

where $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ are obtained from the particle filter on the test sequence. For synthetic data, we also measure the mean-squared error (MSE) of the posterior sample average $\hat{\theta}^{(s)} = \sum_{k \le s} \theta^{(k)}/s$ to the true parameters $\theta^*$.

We measure the sample quality of our MCMC chains $\{\theta^{(k)}\}_{k=1}^K$ using the *kernel Stein discrepancy* (KSD) given equal computation 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. 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

$$\mathrm{KSD}(\hat{p}, p) = \sum_{d=1}^{\dim(\theta)} \sqrt{\sum_{k,k'=1}^{\tilde{K}} \mathcal{K}_0^d(\theta^{(k)}, \theta^{(k')})}, \tag{23}$$

where

$$\mathcal{K}_0^d(\theta, \theta') = \frac{1}{p(\theta|y)p(\theta'|y)} \nabla_{\theta_d} \nabla_{\theta'_d} (p(\theta|y)\mathcal{K}(\theta, \theta')p(\theta'|y)) \tag{24}$$

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 Eq. (24) requires full gradient evaluations of $\log p(\theta|y)$ that are computationally intractable, we replace these terms with corresponding stochastic estimates using $g_\theta^{\mathrm{PF}}$.

### 4.3.2 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 $10^6$ and test sequences of length $T = 10^3$ using the same parametrization as Section 4.2.

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 $\epsilon$ and selected the method with smallest KSD to the posterior on the training set. We present the KSD results (for the best $\epsilon$) in Table 1 and trace plots of the metrics in Figure 3.

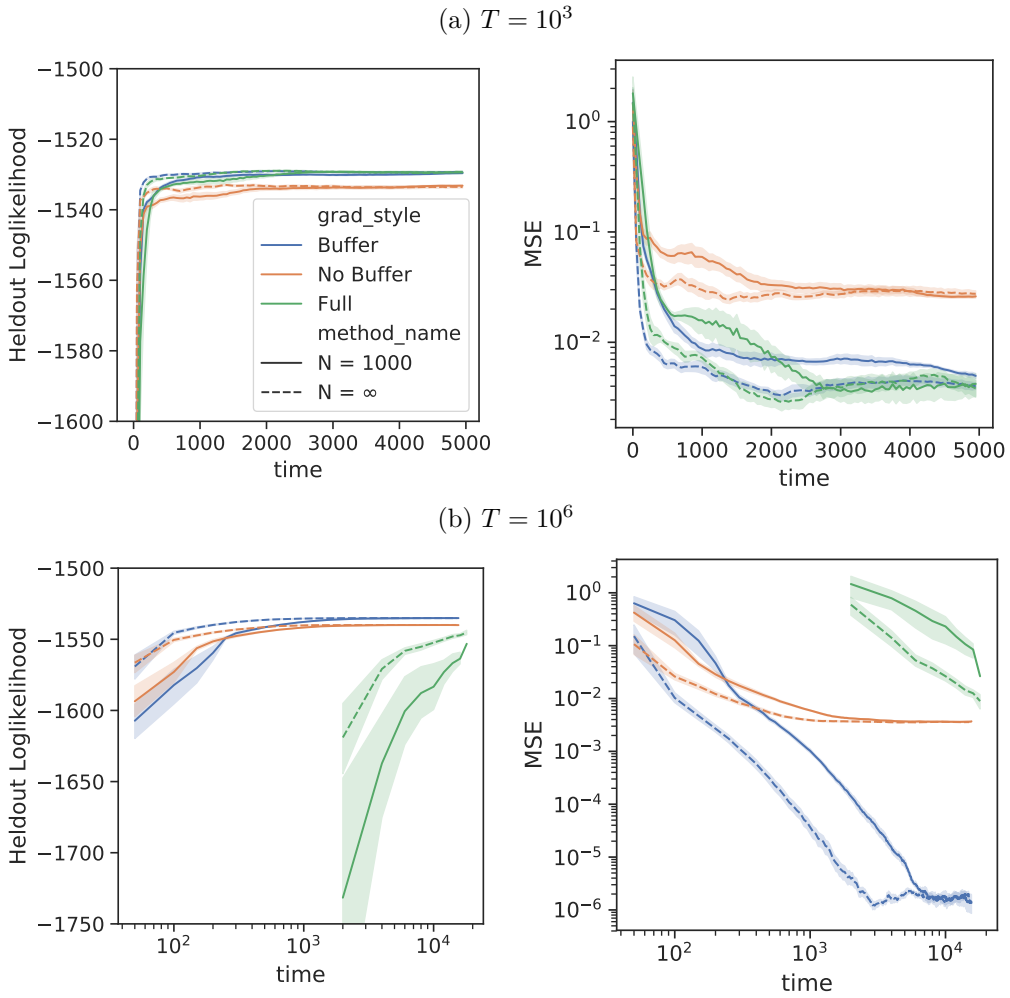


Figure 3: SGLD on Synthetic LGSSM data. (top) $T = 10^3$, (bottom) $T = 10^6$. (left) heldout-loglikelihood, (right) MSE of estimated posterior mean of $\hat{\phi}^{(k)}$ to true $\phi = 0.9$.

From Figure 3, we see that the methods without buffering ($B = 0$) have lower heldout loglikelihoods on the test sequence and 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 1 presents the mean and standard deviation on our estimated $\log_{10}$ KSD for $\theta$. Tables of the marginal KSD for individual components of $\theta$ can be

found in the Supplement. 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.

Table 1: KSD for Synthetic LGSSM. Mean and SD.

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

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

#### 4.3.3 SGLD on Exchange Rate Log-Returns

We now consider fitting the SVM and the GARCH model to EUR-US 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. Note that our method (Algorithm 2) easily scales to the unsegmented series; however the abrupt changes between starts of weeks are not adequately modeled by Eqs. (16)-(17)

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 $\epsilon$ and selected the method with smallest KSD. We present the KSD results in Table 2. Figure 4 are trace plots of the heldout and predictive loglikelihood for the four different SGLD methods, each averaged over 5 chains.

Table 2: KSD for SGLD on exchange rate data. Mean and SD over 5 chains each.

| | $\log_{10}$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 improves performance on both heldout and predictive loglikelihoods, Figure 4(top), and also leads to more accurate MCMC samples, Table 2(left). In

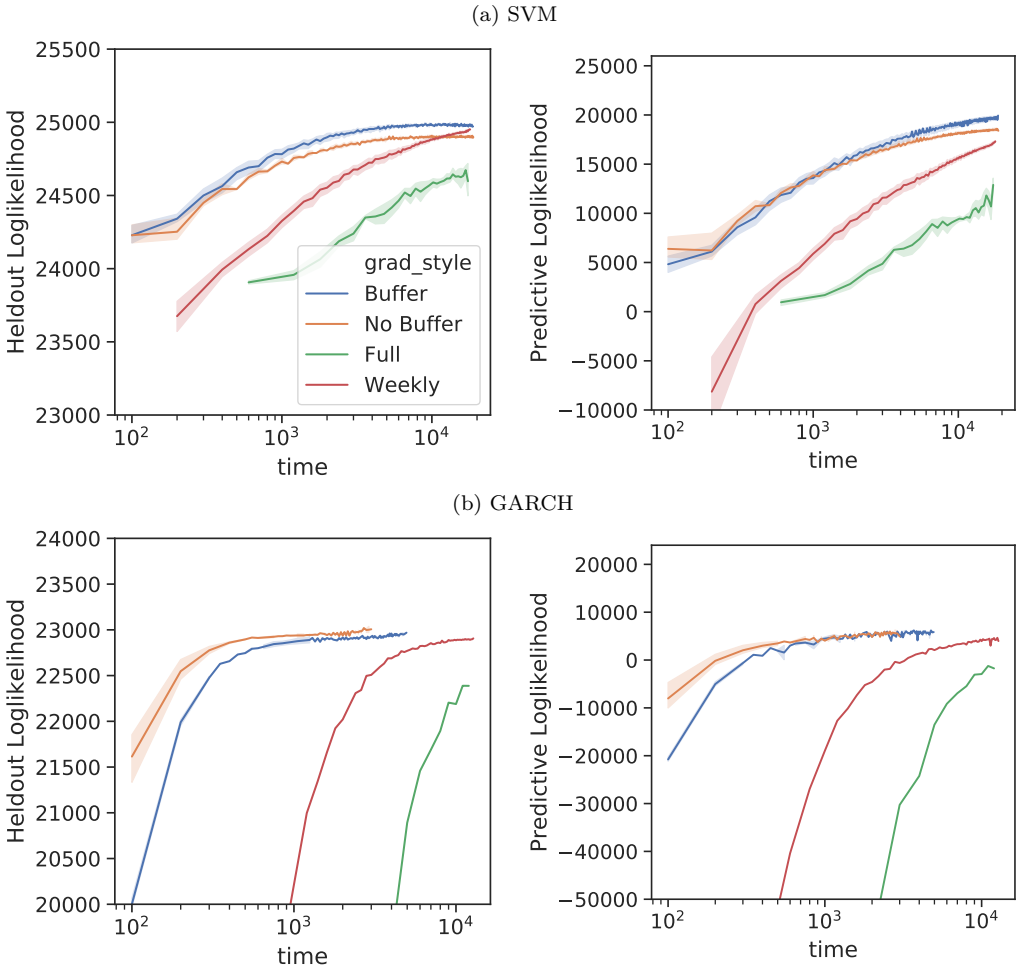


Figure 4: SGLD plots on exchange rate data. (top) SVM, (bottom) GARCH, (left) heldout-loglikelihood, (right) 3-step ahead predictive loglikelihood.

particular, the samples from SGLD without buffering have smaller $\phi, \tau^2$ and a larger $\sigma^2$, 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, Figure 4(bottom) and Table 2(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. Although buffering performs worse on a runtime scale, here, it is leading to a more accurate posterior estimate (less bias) in *all* settings.

# 5 Discussion

In this work, we developed a particle buffered stochastic gradient estimators for nonlinear SSMs. Our key contributions are (i) combining buffered stochastic gradient MCMC with particle filtering

for nonlinear SSM (Algorithm 1), (ii) decomposing the error of our proposed gradient estimator into parts due to buffering and particle filtering, and (iii) generalizing the geometric decay error bound for buffering to nonlinear SSMs with log-concave likelihoods (Theorem 1). We evaluated our proposed gradient estimator with SGLD for three models (LGSSM, SVM, GARCH) on both synthetic data and EUR-US exchange rate data. We find 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 1, 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 optimization in variational autoencoders for sequential data [Maddison et al., 2017, Naesseth et al., 2018].

## Acknowledgements

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

## References

Christopher Aicher, Yi-An Ma, Nicholas J. Foti, and Emily B. Fox. Stochastic Gradient MCMC for State Space Models. *arXiv preprint arXiv:1810.09098*, 2018.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342, 2010.

Jack Baker, Paul Fearnhead, Emily B Fox, and Christopher Nemeth. Control variates for stochastic gradient MCMC. *Statistics and Computing*, Aug 2018. ISSN 1573-1375.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307 – 327, 1986. ISSN 0304-4076.

Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*. Springer, 2005.

Niladri S Chatterji, Nicolas Flammarion, Yi-An Ma, Peter L Bartlett, and Michael I Jordan. On the Theory of Variance Reduction for Stochastic Gradient Monte Carlo. In *Proceedings of the 35th International Conference on Machine Learning*, pages 764–773, 2018.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

Maria Colombo, Alessio Figalli, and Yash Jhaveri. Lipschitz changes of variables between perturbations of log-concave measures. *arXiv preprint arXiv:1510.03687*, 2015.

Johan Dahlin, Fredrik Lindsten, and Thomas B Schön. Particle Metropolis–Hastings using gradient and Hessian information. *Statistics and Computing*, 25(1):81–92, 2015.

Arnak S Dalalyan and Avetik G Karagulyan. User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. *arXiv preprint arXiv:1710.00095*, 2017.

Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems*, pages 3203–3211, 2014.

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

Vanja Dukic, Hedibert F Lopes, and Nicholas G Polson. Tracking epidemics with Google flu trends data and a state-space SEIR model. *Journal of the American Statistical Association*, 107(500): 1410–1426, 2012.

Paul Fearnhead and Hans R. Künsch. Particle Filters and Data Assimilation. *Annual Review of Statistics and Its Application*, 5(1):421–449, 2018.

Andrew Gelman, John B Carlin, Donald B Rubin, Aki Vehtari, David B Dunson, and Hal S Stern. *Bayesian Data Analysis*. CRC Press, 2013.

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

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

Jackson Gorham and Lester Mackey. Measuring Sample Quality with Kernels. *arXiv preprint arXiv:1703.01717*, 2017.

Gregor Kastner. Dealing with stochastic volatility in time series using the R package stochvol. *Journal of Statistical Software*, 69(5):1–30, 2016. doi: 10.18637/jss.v069.i05.

Genshiro Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

Genshiro Kitagawa and Seisho Sato. Monte Carlo smoothing and self-organising state-space model. In *Sequential Monte Carlo Methods in Practice*, pages 177–195. Springer, 2001.

Roland Langrock. Some applications of nonlinear and non-Gaussian state–space modelling by means of hidden Markov models. *Journal of Applied Statistics*, 38(12):2955–2970, 2011.

Jun S. Liu and Rong Chen. Sequential Monte Carlo methods for Dynamic Systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning*, pages 276–284, 2016.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, pages 2917–2925, 2015.

Yi-An Ma, Nicholas J Foti, and Emily B Fox. Stochastic Gradient MCMC Methods for Hidden Markov Models. In *International Conference on Machine Learning*, pages 2265–2274, 2017.

Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6573–6583, 2017.

Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational Sequential Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 968–977, 2018.

Tigran Nagapetyan, Andrew B Duncan, Leonard Hasenclever, Sebastian J Vollmer, Lukasz Szpruch, and Konstantinos Zygalakis. The true cost of stochastic gradient Langevin dynamics. *arXiv preprint arXiv:1706.02692*, 2017.

Christopher Nemeth, Paul Fearnhead, and Lyudmila Mihaylova. 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, 2016.

Jimmy Olsson, Johan Westerborn, et al. Efficient particle-based online smoothing in general hidden Markov models: the PaRIS algorithm. *Bernoulli*, 23(3):1951–1996, 2017.

Rihui Ou, Alexander L Young, and David B Dunson. Clustering-Enhanced Stochastic Gradient MCMC for Hidden Markov Models with Rare States. *arXiv preprint arXiv:1810.13431*, 2018.

Sam Patterson and Yee Whye Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, pages 3102–3110, 2013.

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

Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, 09 1951.

Adrien Saumard and Jon A Wellner. Log-concavity and strong log-concavity: a review. *Statistics surveys*, 8:45, 2014.

Mark Schmidt, Reza Babanezhad, Mohamed Ahmed, Aaron Defazio, Ann Clifton, and Anoop Sarkar. Non-uniform stochastic average gradient method for training conditional random fields. In *Artificial Intelligence and Statistics*, pages 819–828, 2015.

Neil Shephard. *Stochastic Volatility: Selected Readings*. Oxford University Press, 2005.

Yee Whye Teh, Alexandre H Thiery, and Sebastian J Vollmer. Consistency and fluctuations for stochastic gradient Langevin dynamics. *The Journal of Machine Learning Research*, 17(1):193–225, 2016.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

# Supplement

This Supplement is organized as follows. In Section A, we provide additional details for the error analysis. In particular, we provide the proof of Theorem 1 in Section A.2.3 and applications of it to the LGSSM and SVM in Section A.3. In Section B, we provide additional particle filter and gradient details for the models in Section 4.1. In Section C, we provide additional details and figures of experiments.

# A Error Analysis Supplement

## A.1 Stochastic Subsequence Error

We are interested in the (mean-squared) error between the full gradient $g_\theta$ and the unbiased stochastic gradient estimate $\widehat{g}_\theta(S,T)$, specifically for the case of randomly sampling a *contiguous* subsequence $\mathcal{S}$. Because $\widehat{g}_\theta(S,T)$ is unbiased, this reduces to calculating the variance of $\widehat{g}_\theta(S,T)$ with respect to the sampling distribution of the subsequence $\mathcal{S}$. For simplicity, we consider the 1-D case and assume $\Pr(t \in \mathcal{S}) = S/T$.

Let $f_t = \mathbb{E}_{x_{1:T}|y_{1:T},\theta}[\nabla \log p(y_t, x_t \,|\, x_{t-1}, \theta)]$, thus

$$g_\theta = \mathbb{E}_{x_{1:T}|y_{1:T},\theta}\left[\sum_{t=1}^{T} \nabla \log p(y_t, x_t \,|\, x_{t-1}, \theta)\right] = \sum_{t=1}^{T} f_t, \tag{A.1}$$

$$\widehat{g}_\theta(S,T) = \mathbb{E}_{x_{1:T}|y_{1:T},\theta}\left[\sum_{t\in\mathcal{S}} \Pr(t \in \mathcal{S})^{-1} \cdot \nabla \log p(y_t, x_t \,|\, x_{t-1}, \theta)\right] = \tfrac{T}{S}\sum_{t\in\mathcal{S}} f_t. \tag{A.2}$$

Consider the uniform random variable $\hat{t}$ over $\{1,\dots,T\}$. To bound the variance of $\widehat{g}_\theta(S,T)$, we additionally assume $|\operatorname{Corr}(f_{\hat{t}}, f_{\hat{t}+s})| \le \rho^s$, that is the correlation between gradients decays with time. This assumption is reasonable when both the observations $Y_{1:T}$ and posterior latent states $X_{1:T}|Y_{1:T}$ are *ergodic* (i.e. exhibit a exponential forgetting property) [Cappé et al., 2005]. Let $V = \operatorname{Var}(f_{\hat{t}})$ be the variance and recall the following covariance formula $\operatorname{CoV}(X,Y) \le |\operatorname{Corr}(X,Y)|\sqrt{\operatorname{Var}(X)\operatorname{Var}(Y)}$. Then we have

$$\operatorname{Var}(\widehat{g}_\theta(S,T)) = \frac{T^2}{S^2}\cdot \operatorname{Var}\left[\sum_{t\in\mathcal{S}} f_t\right], \tag{A.3}$$

$$= \frac{T^2}{S^2}\cdot\left[S\cdot\operatorname{Var}(f_{\hat{t}}) + \sum_{s=1}^{S-1} 2(S-s)\operatorname{CoV}(f_{\hat{t}}, f_{\hat{t}+s})\right], \tag{A.4}$$

$$\le \frac{T^2}{S^2}\cdot\left[S\cdot V + \sum_{s=1}^{S-1} 2(S-s)\cdot V\rho^s\right], \tag{A.5}$$

$$= \frac{T^2}{S^2}\left[S\cdot V + V\cdot 2\frac{\rho(\rho^S + S(1-\rho) - 1)}{(1-\rho)^2}\right] = \mathcal{O}\left(\frac{T^2\cdot V}{S}\cdot\frac{1+\rho}{1-\rho}\right). \tag{A.6}$$

Note that without the decaying correlation assumption (i.e. if $\rho = 1$), there is no decay in the covariance terms, and thus the variance of $\widehat{g}_\theta(S,T)$ does not necessarily decay with increasing $S$.

## A.2 Proof of Theorem 1

Theorem 1 states that if the prior distribution for $x_0$, the transition distribution $p(x_t \,|\, x_{t-1}, \theta)$ and the emission distribution $p(y_t \,|\, x_t)$ are log-concave, then we can bound the Lipschitz constant of $\vec{\Psi}_t$ in terms of $\vec{\Psi}_t^{(0)}$ and $\vec{\Psi}_t^{(1)}$.

We first briefly review Wasserstein distance, random mappings, and Lipschitz constants of kernels [Aicher et al., 2018, Villani, 2008]. Then we review *Caffarelli's log-concave perturbation theorem*, the main tool we use in our proof. Finally, we present the proof in Section A.2.3.

### A.2.1 Wasserstein Distance and Random Mappings

The *p-Wasserstein distance* with respect to Euclidean distance is

$$\mathcal{W}_p(\gamma, \widetilde{\gamma}) := \left[\inf_{\xi} \int \|x - \widetilde{x}\|_2^p \, d\xi(x, \widetilde{x})\right]^{1/p} \tag{A.7}$$

where $\xi$ is a joint measure or *coupling* over $(x, \widetilde{x})$ with marginals $\int_{\widetilde{x}} d\xi(x, \widetilde{x}) = d\gamma(x)$ and $\int_x d\xi(x, \widetilde{x}) = d\widetilde{\gamma}(x)$.

To bound the Wasserstein distance, we first must introduce the concept of a *random mapping* associated with a transition kernel.

Let $\Psi : \mathcal{U} \to \mathcal{V}$ be a transition kernel for random variables $u$ and $v$, then for any measure $\mu(u)$ over $\mathcal{U}$, we define the induced measure $(\mu\Psi)(v)$ over $\mathcal{V}$ as $(\mu\Psi)(v) = \int \Psi(u, v)\mu(du)$.

A *random mapping* $\psi$ is a random function that maps $\mathcal{U}$ to $\mathcal{V}$ such that if $u \sim \mu$ then $\psi(u) \sim \mu\Psi$. For example, if $\Psi(u, v) = \mathcal{N}(v \,|\, u, 1)$, then a random mapping for $\Psi$ is the identity function plus Gaussian noise $\psi(u) = u + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$. Note that if $\psi$ is deterministic $(\mu\Psi)(v)$ is the *push-forward* measure of $\mu$ through the mapping $\psi$; otherwise it is the average (or marginal) over $\psi$ of push-forward measures [Villani, 2008].

We say the kernel (and random mapping) has Lipschitz constant $L$ with respect to Euclidean distance if

$$\|\Psi\|_{Lip} = \|\psi\|_{Lip} = L \Leftrightarrow \sup_{u,u'} \left\{ \frac{\mathbb{E}_{\psi}[\|\psi(u) - \psi(u')\|_2]}{\|u - u'\|_2} \right\} \leq L \tag{A.8}$$

Note that is $L$ is an upper-bound on the *expected value* of Lipschitz constants for random instances of $\psi$.

These definitions are useful for proving bounds in Wasserstein distance. For example, we can show the kernel $\Psi$ induces a contraction in $p$-Wasserstein distance if $\|\Psi\|_{Lip} < 1$. That is $\mathcal{W}_p(\mu\Psi, \tilde{\mu}\Psi) \leq \|\Psi\|_{Lip} \cdot \mathcal{W}_p(\mu, \tilde{\mu})$

*Proof.*

$$\begin{aligned}
\mathcal{W}_p(\mu\Psi, \tilde{\mu}\Psi)^p &= \inf_{\xi(\mu\Psi, \tilde{\mu}\Psi)} \int \|v - \tilde{v}\|_2^p \, d\xi(v, \tilde{v}) \\
&\leq \inf_{\xi(\mu, \tilde{\mu})} \int \|\psi(u) - \psi(\tilde{u})\|_2^p \, d\xi(u, \tilde{u}) df_K \\
&\leq \inf_{\xi(\mu, \tilde{\mu})} \int \|\Psi\|_{Lip}^p \cdot \|u - \tilde{u}\|_2^p \, d\xi(u, \tilde{u}) = \|\Psi\|_{Lip}^p \cdot \mathcal{W}_p(\mu, \tilde{\mu})^p \ .
\end{aligned} \tag{A.9}$$

□

### A.2.2 Caffarelli's Log-Concave Perturbation Theorem

Caffarelli's log-concave perturbation theorem allows us to connect Lipschitz constants between kernels that are log-concave perturbations of one another.

**Theorem A.1** (Caffarelli's)**.** *Let $\gamma(x)$ be a log-concave measure for $x$ and suppose $\ell(x)$ is a log-concave function such that $\gamma'(x) = \ell(x)\gamma(x)$ is a probability measure over $x$. Then there exists a 1-Lipschitz mapping $T : \mathcal{X} \to \mathcal{X}$ such that if $x \sim \gamma(x)$ then $T(x) \sim \gamma'(x)$.*

We can think of $\gamma(x)$ as a prior distribution $p(x)$, $\ell(x)$ as a normalized conditional likelihood $p(y|x)/p(y)$ and $\gamma'(x)$ as the posterior $p(x|y)$. Because $\ell(x)$ is log-concave, we call $\gamma'(x)$ a *log-concave perturbation* of $\gamma$.

The original version of Caffarelli's theorem [Colombo et al., 2015, Saumard and Wellner, 2014] requires the prior $\gamma(x)$ to be *strongly* log-concave (e.g. a Gaussian) to show that the mapping $T$ is a *strict* contraction $\|T\|_{Lip} < 1$; however this weaker version, Theorem A.1 in [Villani, 2008], is sufficient for our purposes.

### A.2.3 Proof of Theorem 1

Using Theorem A.1, we can now prove Theorem 1 from Section 3.3.1.

*Proof of Theorem 1.* Let $\vec{\psi}_t, \vec{\psi}_t^{(0)}, \vec{\psi}_t^{(1)}$ be random mappings associated with the forward kernels $\vec{\Psi}_t, \vec{\Psi}_t^{(0)}, \vec{\Psi}_t^{(1)}$ respectively. Because the transition and emission distributions are log-concave, $p(y_{t:T}, x_{t:T} \,|\, x_t)$ and $p(y_{t:T} \,|\, x_t)$ are also log-concave (recall log-concavity is preserved under products and marginalization [Saumard and Wellner, 2014]).

Since $p(y_{t:T} \,|\, x_t)$ is log-concave, we can write $\vec{\Psi}_t$ as a log-concave perturbation of $\vec{\Psi}_t^{(0)}$,

$$\vec{\Psi}_t = p(x_t \,|\, x_{t-1}, y_{t:T}, \theta) \propto p(y_{t:T} \,|\, x_t) p(x_t \,|\, x_{t-1}, \theta) = p(y_{t:T} \,|\, x_t) \cdot \vec{\Psi}_t^{(0)}. \tag{A.10}$$

Therefore, there exists $T_t^{(0)}$ with $\|T_t^{(0)}\|_{Lip} \leq 1$ such that $\vec{\psi}_t = (T_t^{(0)} \circ \vec{\psi}_t^{(0)})$. Thus,

$$\|\vec{\Psi}_t\|_{Lip} = \|T_t^{(0)}\|_{Lip} \cdot \|\vec{\Psi}_t^{(0)}\|_{Lip} \leq \|\vec{\Psi}_t^{(0)}\|_{Lip}. \tag{A.11}$$

Similarly, we can write $\vec{\Psi}_t$ as a log-concave perturbation of $\vec{\Psi}_t^{(1)}$ using $p(y_{>t} \,|\, x_t)$, thus $\|\vec{\Psi}_t\|_{Lip} \leq \|\vec{\Psi}_t^{(1)}\|_{Lip}$. □

Note the assumptions for equivalent results in the backward smoothers $\overleftarrow{\Psi}_t$ are almost identical. Note that log-concavity in $p(x_t \,|\, x_{t+1}, \theta)$ is implied from log-concavity inboth $p(x_t \,|\, x_{t-1}, \theta)$ and the prior $p(x_t)$.

## A.3 Bounds for Specific Models

We now provide specific bounds for the buffering error for models we consider in Section 4.

For both the LGSSM and SVM, we assume the prior $\nu(x_0|\theta) = \mathcal{N}(0, \sigma^2/(1-\phi^2))$. Then the latent state transitions are $p(x_t \,|\, x_{t-1}, \theta) = \mathcal{N}(x_t \,|\, \phi x_t, \sigma^2)$ and $p(x_t \,|\, x_{t+1}, \theta) = \mathcal{N}(x_t \,|\, \phi x_{t+1}, \sigma^2)$, which are both Gaussian and therefore log-concave.

Similarly, the emissions for the LGSSM and SVM are also log-concave:

- For the LGSSM, $p(y_t \,|\, x_t, \theta) \propto \exp(-(y_t - x_t)^2/(2\sigma^2)$ is log-concave,
- For the SVM, $p(y_t \,|\, x_t, \theta) \propto \exp(-y_t^2/(2\sigma^2) \cdot \exp(-x) - x/2)$ is log-concave (as $\exp(-x) + x$ is convex).

### A.3.1 Contraction Bound for LGSSM

We assume the prior $\nu(x_0|\theta) = \mathcal{N}(0, \sigma^2/(1-\phi^2))$. For the LGSSM, the filtered kernels are

$$\vec{\Psi}_t^{(1)}(x_t \,|\, x_{t-1}) = p(x_t \,|\, x_{t-1}, y_t, \theta) \propto \mathcal{N}(x_t|\phi x_{t-1}, \sigma^2) \cdot \mathcal{N}(y_t|x_t, \tau^2), \tag{A.12}$$

$$\overleftarrow{\Psi}_t^{(1)}(x_t \,|\, x_{t+1}) = p(x_t \,|\, x_{t+1}, y_t, \theta) \propto \mathcal{N}(x_t|0, \sigma^2/(1-\phi^2)) \cdot \mathcal{N}(y_t|x_t, \tau^2) \cdot \mathcal{N}(x_{t+1}|\phi x_t, \sigma^2). \tag{A.13}$$

Therefore,

$$\vec{\Psi}_t^{(1)}(x_t \,|\, x_{t-1}) = \mathcal{N}\left(x_t \,\middle|\, \frac{\sigma^2 y_t + \phi\tau^2 x_{t-1}}{\sigma^2 + \tau^2}, \frac{\sigma^2\tau^2}{\sigma^2 + \tau^2}\right), \tag{A.14}$$

$$\overleftarrow{\Psi}_t^{(1)}(x_t \,|\, x_{t+1}) = \mathcal{N}\left(x_t \,\middle|\, \frac{\sigma^2 y_t + \phi\tau^2 x_{t+1}}{\sigma^2 + \tau^2}, \frac{\sigma^2\tau^2}{\sigma^2 + \tau^2}\right). \tag{A.15}$$

The associated random mapping are,

$$\vec{\psi}_t^{(1)}(x_t \,|\, x_{t-1}) = \frac{\sigma^2 y_t}{\sigma^2+\tau^2} + \frac{\phi\tau^2}{\sigma^2+\tau^2} \cdot x_{t-1} + \mathcal{N}\left(0\,,\, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\right), \tag{A.16}$$

$$\overleftarrow{\psi}_t^{(1)}(x_t \,|\, x_{t+1}) = \frac{\sigma^2 y_t}{\sigma^2+\tau^2} + \frac{\phi\tau^2}{\sigma^2+\tau^2} \cdot x_{t+1} + \mathcal{N}\left(0\,,\, \frac{\sigma^2\tau^2}{\sigma^2+\tau^2}\right). \tag{A.17}$$

Since these maps are linear, we have $\|\vec{\Psi}_t^{(1)}\|_{Lip} = \|\overleftarrow{\Psi}_t^{(1)}\|_{Lip} = |\phi| \cdot \frac{\tau^2}{\sigma^2+\tau^2}$. Applying Theorem 1, we obtain

$$L_\theta \leq \max_t \{\|\vec{\Psi}_t^{(1)}\|, \|\overleftarrow{\Psi}_t^{(1)}\|\} = |\phi| \cdot (1+\sigma^2/\tau^2)^{-1}. \tag{A.18}$$

Therefore $L_\theta < 1$ whenever $|\phi| < 1 + \sigma^2/\tau^2$.

### A.3.2 Contraction Bound for SVM

We assume the prior $\nu(x_0\theta) = \mathcal{N}(0, \sigma^2/(1-\phi^2))$. For the SVM, the prior kernels are,

$$\vec{\Psi}_t^{(0)}(x_t \,|\, x_{t-1}) = p(x_t \,|\, x_{t-1}, \theta) \propto \mathcal{N}(x_t|\phi x_{t-1}, \sigma^2), \tag{A.19}$$

$$\overleftarrow{\Psi}_t^{(0)}(x_t \,|\, x_{t+1}) = p(x_t \,|\, x_{t+1}, \theta) \propto \mathcal{N}(x_t|0, \sigma^2/(1-\phi^2)) \cdot \mathcal{N}(x_{t+1}|\phi x_t, \sigma^2). \tag{A.20}$$

Therefore,

$$\vec{\Psi}_t^{(0)}(x_t \,|\, x_{t-1}) = \mathcal{N}\left(x_t \,|\, \phi x_{t-1}\,,\, \sigma^2\right), \tag{A.21}$$

$$\overleftarrow{\Psi}_t^{(0)}(x_t \,|\, x_{t+1}) = \mathcal{N}\left(x_t \,|\, \phi x_{t+1}\,,\, \sigma^2\right). \tag{A.22}$$

The associated random mapping are

$$\vec{\psi}_t^{(0)}(x_t \,|\, x_{t-1}) = \phi \cdot x_{t-1} + \mathcal{N}\left(0\,,\, \sigma^2\right), \tag{A.23}$$

$$\overleftarrow{\psi}_t^{(0)}(x_t \,|\, x_{t+1}) = \phi \cdot x_{t+1} + \mathcal{N}\left(0\,,\, \sigma^2\right). \tag{A.24}$$

Applying Theorem 1, we obtain $L_\theta \leq |\phi|$.

# B Model Details Supplement

## B.1 Linear Gaussian State Space Model (LGSSM)

The LGSSM used in this paper is a scalar AR(1) model,

$$X_t \,|\, (X_{t-1} = x_{t-1}), \theta \sim \mathcal{N}(x_t \,|\, \phi x_{t-1}\,,\, \sigma^2), \tag{B.1}$$

$$Y_t \,|\, (X_t = x_t), \theta \sim \mathcal{N}(y_t \,|\, x_t\,,\, \tau^2), \tag{B.2}$$

with parameters $\theta = (\phi, \sigma, \tau)$.

When applying the particle filter, Algorithm 1, to the LGSSM, we consider two proposal densities $q(\cdot|\cdot)$:

- The prior (transition) kernel

$$X_t \,|\, (X_{t-1} = x_{t-1}), \theta \sim \mathcal{N}(x_t \,|\, \phi x_{t-1}, \sigma^2), \tag{B.3}$$

  where the weight update, Eq. (3), is

$$w_t^{(i)} \propto \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(\frac{-(y_t - x_t^{(i)})^2}{2\tau^2}\right). \tag{B.4}$$

- The 'optimal instrumental kernel'

$$X_t \,|\, (X_{t-1} = x_{t-1}, Y_t = y_t), \theta \sim \mathcal{N}\left(x_t \,\middle|\, \frac{\tau^2 \phi x_{t-1} + \sigma^2 y_t}{\sigma^2 + \tau^2}, \frac{\sigma^2 \tau^2}{\sigma^2 + \tau^2}\right), \tag{B.5}$$

  where the weight update, Eq. (3), is

$$w_t^{(i)} \propto \frac{1}{\sqrt{2\pi(\sigma^2 + \tau^2)}} \exp\left(\frac{-(y_t - \phi x_{t-1}^{(a_i)})^2}{2(\sigma^2 + \tau^2)}\right). \tag{B.6}$$

In our experiments with the LGSSM, we use the optimal instrumental kernel.

For this model, the (elementwise) complete data loglikelihood is

$$\log p(y_t, x_t \,|\, x_{t-1}, \theta) = \log(2\pi) - \log(\sigma) - \frac{(x_t - \phi x_{t-1})^2}{2\sigma^2} - \log(\tau) - \frac{(y_t - x_t)^2}{2\tau^2}. \tag{B.7}$$

The gradient of the complete data loglikelihood is then,

$$\nabla_\phi \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{(x_t - \phi x_{t-1}) \cdot x_{t-1}}{\sigma^2}, \tag{B.8}$$

$$\nabla_\sigma \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{(x_t - \phi x_{t-1})^2 - \sigma^2}{\sigma^3}, \tag{B.9}$$

$$\nabla_\tau \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{(y_t - x_t)^2 - \tau^2}{\tau^3}. \tag{B.10}$$

We reparametrize the gradients with $\sigma^{-1}$ and $\tau^{-1}$ to obtain,

$$\nabla_{\sigma^{-1}} \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{\sigma^2 - (x_t - \phi x_{t-1})^2}{\sigma}, \tag{B.11}$$

$$\nabla_{\tau^{-1}} \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{\tau^2 - (y_t - x_t)^2}{\tau}. \tag{B.12}$$

To complete the SGMCMC scheme, the prior distributions of the parameters $\theta$ are given as follows: $\phi \sim \mathcal{N}(0, 100 \cdot \sigma^2), \sigma^{-1} \sim Gamma(1 + 100, \frac{1}{1+100})$, and $\tau^{-1} \sim Gamma(1 + 100, \frac{1}{1+100})$. The initial parameter values for synthetic experiments were drawn from: $\phi \sim \mathcal{N}(0, 1 \cdot \sigma^2)$, $\sigma^{-1} \sim Gamma(2, 0.5)$ and $\tau^{-1} \sim Gamma(2, 0.5)$.

## B.2 Stochastic Volatility Model (SVM)

The SVM used in this paper is given by,

$$X_t \,|\, (X_{t-1} = x_{t-1}), \theta \sim \mathcal{N}(x_t \,|\, \phi x_{t-1}\,,\, \sigma^2), \tag{B.13}$$

$$Y_t \,|\, (X_t = x_t), \theta \sim \mathcal{N}(y_t \,|\, 0\,,\, \exp(x_t)\tau^2), \tag{B.14}$$

with parameters $\theta = (\phi, \sigma, \tau)$. In this model, the observations, $y_{1:T}$, represent the logarithm of the daily difference in the exchange rate and $X$ is the unobserved volatility. We assume that the volatility process is stationary (such that $0 < \phi < 1$), where $\phi$ is the persistence in volatility and $\tau$ is the instantaneous volatility.

For the particle filter, we use the prior kernel as the proposal density $q$

$$X_t \,|\, (X_{t-1} = x_{t-1}), \theta \sim \mathcal{N}(x_t \,|\, \phi x_{t-1}, \sigma^2), \tag{B.15}$$

with weight update

$$w_t^{(i)} \propto \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(\frac{-y_t^2}{2\exp(x_t^{(i)})\tau^2}\right). \tag{B.16}$$

The elementwise complete data loglikelihood is given by,

$$\log p(y_t, x_t \,|\, x_{t-1}, \theta) = \log(2\pi) - \log(\sigma) - \frac{(x_t - \phi x_{t-1})^2}{2\sigma^2} - \log(\tau) - 0.5x_t - \frac{(y_t)^2}{2\exp(x_t)\tau^2}. \tag{B.17}$$

The gradient of the complete data loglikelihood is then,

$$\nabla_\phi \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{(x_t - \phi x_{t-1}) \cdot x_{t-1}}{\sigma^2}, \tag{B.18}$$

$$\nabla_\sigma \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{(x_t - \phi x_{t-1})^2 - \sigma^2}{\sigma^3}, \tag{B.19}$$

$$\nabla_\tau \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{y_t^2/\exp(x_t) - \tau^2}{\tau^3}. \tag{B.20}$$

We parametrize with $\sigma^{-1}$ and $\tau^{-1}$ to obtain,

$$\nabla_{\sigma^{-1}} \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{\sigma^2 - (x_t - \phi x_{t-1})^2}{\sigma}, \tag{B.21}$$

$$\nabla_{\tau^{-1}} \log p(y_t, x_t \,|\, x_{t-1}, \theta) = \frac{\tau^2 - y_t^2/\exp(x_t)}{\tau}. \tag{B.22}$$

The prior distributions and initializations of the parameters $\theta$ are taken to be the same as in the LGSSM case.

## B.3 Generalized Autoregressive Conditional Heteroskedasticity (GARCH) Model

The GARCH(1,1) model (with noise) used in this paper is given by,

$$X_t \,|\, (X_{t-1} = x_{t-1}), \sigma_t^2, \theta \sim \mathcal{N}(x_t \,|\, 0,\, \sigma_t^2), \tag{B.23}$$

$$\sigma_t^2(x_{t-1}, \sigma_{t-1}^2, \theta) = \alpha + \beta x_{t-1}^2 + \gamma\sigma_{t-1}^2, \tag{B.24}$$

$$Y_t \,|\, (X_t = x_t), \theta \sim \mathcal{N}(y_t \,|\, x_t\,,\, \tau^2), \tag{B.25}$$

where parameters are $\theta = (\log\mu, \operatorname{logit}\phi, \operatorname{logit}\lambda, \tau)$ for $\alpha = \mu(1-\phi)$, $\beta = \phi\lambda$, $\gamma = \phi(1-\lambda)$. Note that $\sigma_t^2 = \mu(1-\phi) + \phi(\lambda x_{t-1}^2 + (1-\lambda)\sigma_{t-1}^2)$.

We consider two proposal densities $q(\cdot|\cdot)$ for the GARCH model:

- The prior kernel

$$\begin{bmatrix} X_t \\ \sigma_t^2 \end{bmatrix} \,\Big|\, \begin{bmatrix} X_{t-1} = x_{t-1} \\ \sigma_{t-1}^2 \end{bmatrix}, \theta \sim \begin{bmatrix} \mathcal{N}(x_t \,|\, 0, \alpha + \beta x_{t-1}^2 + \gamma\sigma_{t-1}^2) \\ \delta(\sigma_t^2 \,|\, \alpha + \beta x_{t-1}^2 + \gamma\sigma_{t-1}^2) \end{bmatrix}. \tag{B.26}$$

  where the weight update, Eq. (3), is

$$w_t^{(i)} \propto \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(\frac{-(y_t - x_t^{(i)})^2}{2\tau^2}\right). \tag{B.27}$$

- The optimal instrumental kernel

$$\begin{bmatrix} X_t \\ \sigma_t^2 \end{bmatrix} \Bigg| \begin{bmatrix} X_{t-1} = x_{t-1} \\ \sigma_{t-1}^2 \end{bmatrix}, (Y_t = y_t),\, \theta \sim \begin{bmatrix} \mathcal{N}(x_t \mid \sigma_t^2 y_t/(\sigma_t^2+\tau^2), \sigma_t^2\tau^2/(\sigma_t^2+\tau^2)) \\ \delta(\sigma_t^2 \mid \alpha + \beta x_{t-1}^2 + \gamma\sigma_{t-1}^2) \end{bmatrix}. \quad \text{(B.28)}$$

  where the weight update, Eq. (3), is

$$w_t^{(i)} \propto \frac{1}{\sqrt{2\pi((\sigma_t^{(i)})^2+\tau^2)}} \exp\left(\frac{-y_t^2}{2((\sigma_t^{(i)})^2+\tau^2)}\right). \quad \text{(B.29)}$$

In our experiments with the GARCH model, we use the optimal instrumental kernel.

The elementwise complete data loglikelihood is

$$\log p(y_t, x_t, \sigma_t^2 \mid x_{t-1}, \sigma_{t-1}^2, \theta) = -0.5\log(2\pi) - 0.5\log(\alpha+\beta x_{t-1}^2+\gamma\sigma_{t-1}^2) - \frac{x_t^2}{2(\alpha+\beta x_{t-1}^2+\gamma\sigma_{t-1}^2)}$$
$$- 0.5\log(2\pi) - \log(\tau) - \frac{(y_t-x_t)^2}{2\tau^2}. \quad \text{(B.30)}$$

Let $\mathcal{L}_t = \log p(y_t, x_t, \sigma_t^2 | x_{t-1}, \sigma_{t-1}^2, \theta)$. Then the gradient of the complete data log-likelihood $\nabla\mathcal{L}_t$ is

$$\nabla_\tau \mathcal{L}_t = \frac{(y_t-x_t)^2-\tau^2}{\tau^3}, \quad \text{(B.31)}$$

$$\nabla_{\log\mu} \mathcal{L}_t = \frac{x_t^2-\sigma_t^2}{2\sigma_t^4}\cdot(1-\phi)\cdot\mu, \quad \text{(B.32)}$$

$$\nabla_{\text{logit}\,\phi} \mathcal{L}_t = \frac{x_t^2-\sigma_t^2}{2\sigma_t^4}\cdot(\lambda x_{t-1}^2 + (1-\lambda)\sigma_{t-1}^2 - \mu)\cdot\phi(1-\phi), \quad \text{(B.33)}$$

$$\nabla_{\text{logit}\,\lambda} \mathcal{L}_t = \frac{x_t^2-\sigma_t^2}{2\sigma_t^4}\cdot(\phi x_{t-1}^2 - \phi\sigma_{t-1}^2)\cdot\lambda(1-\lambda). \quad \text{(B.34)}$$

The SGMCMC scheme is completed by setting the prior distributions for the parameters as follows: $(\phi+1)/2 \sim Beta(10, 1.5), \mu \sim U(0,2), (\lambda+1)/2 \sim Beta(20, 1.5)$ and $\tau^2 \sim \mathcal{IG}(2, 0.5)$.

# C Experiment Supplement

We first present additional SGLD results on synthetic data for the LGSSM in higher dimensions, the SVM and the GARCH models. We then present some additional details for the SGLD experiment on the EUR-US exchange rate data.

## C.1 SGLD on Synthetic Data

### C.1.1 LGSSM

Figure C.1 presents extra MSE plots for the parameters not presented in the main paper. Tables C.1 and C.2 present the full KSD results for each sampled variable.

### C.1.2 Higher Dimensional LGSSM

We generate synthetic LGSSM data for $X_t, Y_t \in \mathbb{R}^d$ using $\phi = 0.9\cdot\mathbb{I}_d$, $\sigma = 0.7\cdot\mathbb{I}_d$, and $\tau = \mathbb{I}_d$ for dimensions $d \in \{5, 10\}$. Figure C.2 presents the trace plot metrics for $d = 5$ and Figure C.3 for $d = 10$. Table C.3 presents the KSD tables for both.

We find that the Kalman filter $N = \infty$ is able to much more rapidly mix compared to the particle filter with $N = 1000$. This is both due to the increased particle filter variance in higher dimensions and the longer computation required for sampling particles in higher dimensions. However in both cases, we again see that buffering is necessary to avoid bias.

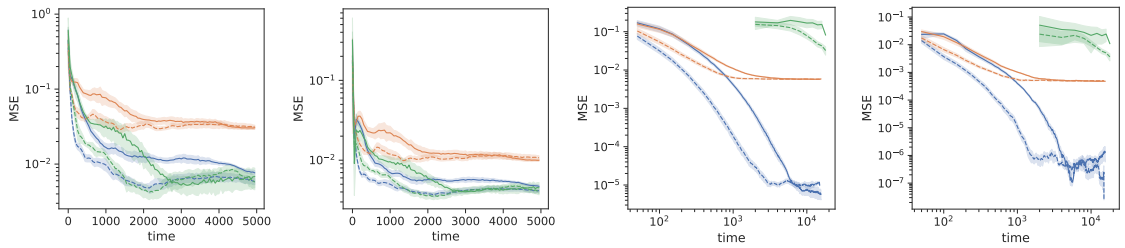


Figure C.1: Additional metrics for SGLD on LGSSM: (left) MSE of $\sigma$ for $T = 10^3$, (center-left) MSE of $\tau$ for $T = 10^3$, (center-right) MSE of $\sigma$ for $T = 10^6$, (right) MSE of $\tau$ for $T = 10^6$

Table C.1: KSD results for Synthetic LGSSM with $T = 10^3$

| $S$ | $B$ | METHOD | $\log_{10}$KSD $\phi$ | $\sigma$ | $\tau$ | TOTAL |
|---|---|---|---|---|---|---|
| $10^3$ | – | GIBBS | 0.09 (0.25) | -0.02 (0.01) | -0.16 (0.48) | **0.51 (0.13)** |
| | | KF | 0.01 (0.57) | 0.07 (0.09) | 0.20 (0.28) | **0.64 (0.17)** |
| | | PF | 0.38 (0.26) | 0.10 (0.16) | 0.44 (0.19) | 0.85 (0.08) |
| 40 | 0 | KF | 1.53 (0.03) | -0.08 (0.07) | -0.04 (0.16) | 1.55 (0.03) |
| | | PF | 1.55 (0.03) | -0.04 (0.13) | 0.10 (0.26) | 1.58 (0.03) |
| 40 | 10 | KF | 0.18 (0.27) | 0.02 (0.07) | 0.04 (0.44) | **0.61 (0.21)** |
| | | PF | 0.27 (0.46) | 0.09 (0.13) | -0.11 (0.53) | **0.68 (0.25)** |

Table C.2: KSD results for Synthetic LGSSM with $T = 10^6$

| $S$ | $B$ | METHOD | $\log_{10}$KSD $\phi$ | $\sigma$ | $\tau$ | TOTAL |
|---|---|---|---|---|---|---|
| $10^6$ | – | GIBBS | 3.91 (0.80) | 3.43 (1.07) | 3.52 (0.73) | 4.23 (0.74) |
| | | KF | 4.51 (0.48) | 4.21 (0.50) | 3.65 (0.55) | 4.85 (0.36) |
| | | PF | 4.77 (0.39) | 4.11 (0.57) | 3.55 (0.95) | 4.92 (0.40) |
| 40 | 0 | KF | 4.64 (0.14) | 3.25 (0.21) | 2.83 (0.61) | 4.68 (0.11) |
| | | PF | 4.64 (0.13) | 3.19 (0.35) | 3.12 (0.45) | 4.68 (0.10) |
| 40 | 10 | KF | 3.04 (0.39) | 1.57 (0.50) | 2.68 (0.20) | **3.25 (0.29)** |
| | | PF | 3.26 (0.17) | 1.70 (0.38) | 2.87 (0.33) | **3.43 (0.19)** |

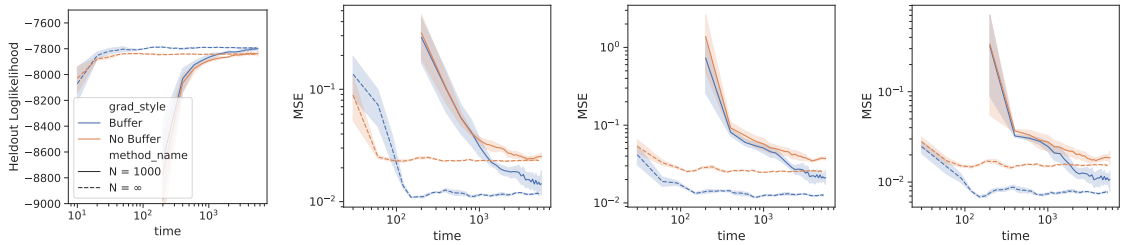


Figure C.2: SGLD Results for LGSSM $X \in \mathbb{R}^5$: (left) heldout loglikelihood, (center-left) MSE of $\phi$, (center-right) MSE of $\sigma$, (right) MSE of $\tau$.

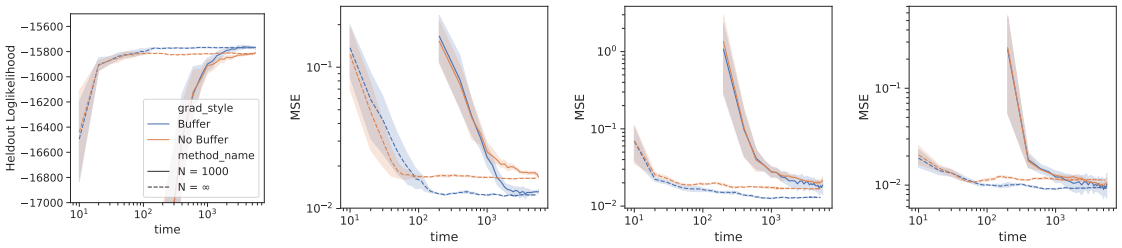


Figure C.3: SGLD Results for LGSSM $X \in \mathbb{R}^{10}$: (left) heldout loglikelihood, (center-left) MSE of $\phi$, (center-right) MSE of $\sigma$, (right) MSE of $\tau$.

Table C.3: KSD results for Synthetic LGSSM in higher dimensions

| | | | $\log_{10}$KSD | | | |
|---|---|---|---|---|---|---|
| Dim | Grad Est. | N | $\phi$ | $\sigma$ | $\tau$ | Total |
| 5 | No Buffer | 1000 | 1.78 (0.04) | 1.97 (0.26) | 1.44 (0.45) | 2.28 (0.20) |
| | | $\infty$ | 1.74 (0.01) | 2.09 (0.02) | 1.64 (0.02) | 2.35 (0.01) |
| | Buffer | 1000 | 1.18 (0.17) | 1.74 (0.25) | 1.44 (0.03) | **2.01 (0.13)** |
| | | $\infty$ | 0.84 (0.03) | 1.97 (0.03) | 1.40 (0.05) | **2.10 (0.03)** |
| 10 | No Buffer | 1000 | 1.84 (0.01) | 2.40 (0.06) | 2.26 (0.13) | 2.71 (0.06) |
| | | $\infty$ | 1.79 (0.01) | 2.13 (0.04) | 2.12 (0.01) | 2.52 (0.02) |
| | Buffer | 1000 | 1.60 (0.13) | 2.37 (0.04) | 2.20 (0.04) | 2.64 (0.04) |
| | | $\infty$ | 1.04 (0.06) | 2.08 (0.04) | 2.07 (0.01) | **2.39 (0.02)** |

### C.1.3 SVM

Figure C.4 presents the trace plot metrics for SGLD on the synthetic SVM data $T = 1000$ and Table C.4 presents the KSD for each sampled chain.

We find that buffering performs best (as measured by KSD). From Figure C.4 we see that not buffering leads to bias, while the full sequence method is nosier (fewer larger steps) compared to the buffer method.

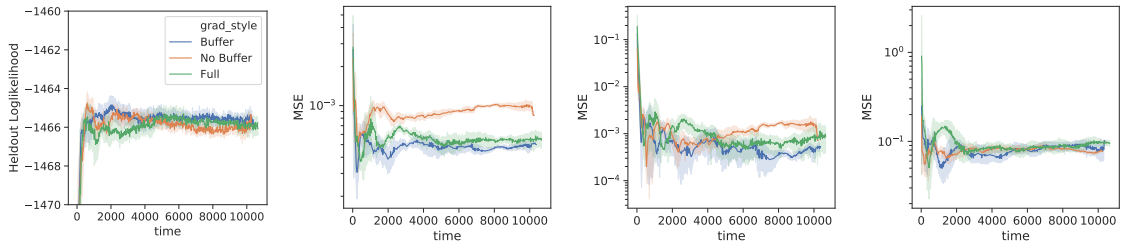


Figure C.4: SGLD results for synthetic SVM data: (left) heldout loglikelihood, (center-left) MSE of $\phi$, (center-right) MSE of $\sigma$, (right) MSE of $\tau$.

Table C.4: KSD results for Synthetic SVM

| | $\log_{10}$KSD | | | |
|---|---|---|---|---|
| Grad Est. | $\phi$ | $\sigma$ | $\tau$ | Total |
| Full | 0.68 (0.28) | 0.38 (0.40) | 0.44 (0.54) | 1.12 (0.22) |
| No Buffer | 1.49 (0.05) | -0.01 (0.23) | 0.09 (0.35) | 1.53 (0.05) |
| Buffer | 0.35 (0.33) | 0.23 (0.29) | 0.21 (0.40) | **0.81 (0.22)** |

### C.1.4 GARCH

Figure C.5 presents the trace plot metrics for SGLD on the synthetic GARCH data $T = 1000$ and Table C.5 presents the KSD for each sampled chain.

We again find that buffering performs best (as measured by KSD). From Figure C.5 we see that not buffering leads to bias in sampling $\mu$ and $\lambda$. The full sequence method encounters high particle error and therefore requires a much longer runtime with a much smaller stepsize to reduce bias.

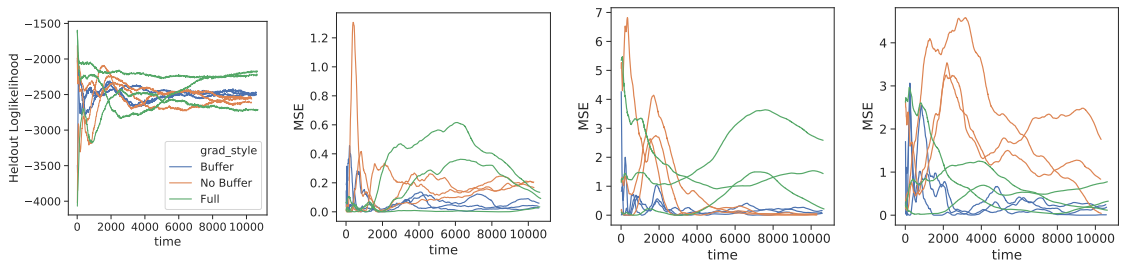


Figure C.5: SGLD results for synthetic SVM data: (left) heldout loglikelihood, (center-left) MSE of $\log(\mu)$, (center-right) MSE of logit $\phi$, (right) MSE of logit $\lambda$.

Table C.5: KSD results for Synthetic GARCH

| | $\log_{10}$KSD | | | | |
|---|---|---|---|---|---|
| Grad Est. | $\log \mu$ | $\text{logit}\, \lambda$ | $\text{logit}\, \phi$ | $\tau$ | Total |
| Full | 0.29 (0.59) | 0.04 (0.03) | 0.18 (0.34) | 0.55 (0.11) | 0.97 (0.05) |
| No Buffer | 0.07 (0.08) | -0.38 (0.09) | -0.15 (0.10) | 0.56 (0.10) | 0.77 (0.08) |
| Buffer | -0.27 (0.24) | -0.72 (0.19) | -0.69 (0.17) | 0.12 (0.19) | **0.39 (0.09)** |

## C.2 SGLD on Exchange Rate

The EUR-US exchange rate data was fulled from the `https://www.finam.ru` website for the time period from November 2017 to October 2018 at the minute resolution. The *demeaned log-returns* are calculated by taking the difference of the log-closing price (at each minute) and removing the mean, as done in the `stochvol` package in R [Kastner, 2016]

$$\tilde{y}_t = \log(y_t / y_{t-1}) - \frac{1}{T} \sum_{t'} \log(y_{t'} / y_{t'-1}) \ . \tag{C.1}$$

The data is plotted in Figure C.6.

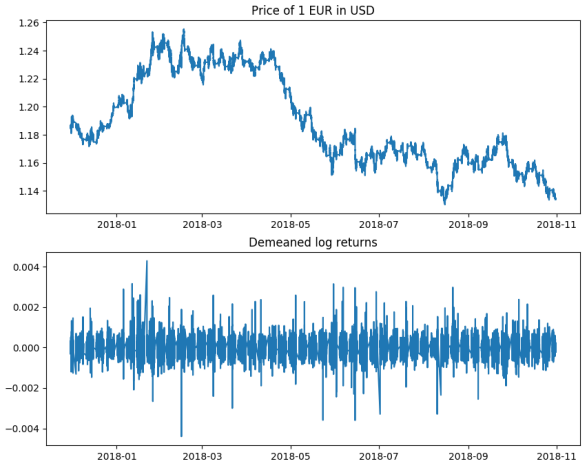


Figure C.6: EUR-US Exchange Rate Data (top) raw data (bottom) demeaned log-returns

### C.2.1 SVM

For the SVM, we initialized each chain at $\phi = 0.9$, $\sigma = 1.73$ and $\tau = 0.1$ for all SGLD methods. The full KSD results are presented in Table C.6.

### C.2.2 GARCH

For the GARCH model, we initialized each chain at $\log \mu = -0.4$, $\text{logit}\, \phi = 1.7$, $\text{logit}\, \lambda = 2.7$ and $\tau = 0.1$ for all SGLD methods. The full KSD results are presented in Table C.7.

Table C.6: KSD results for SVM on exchange rate data.

| | $\log_{10}$KSD | | | |
|---|---|---|---|---|
| Grad Est. | $\phi$ | $\sigma$ | $\tau$ | Total |
| Full | 3.63 (0.30) | 3.76 (0.07) | 1.46 (0.38) | 4.03 (0.14) |
| Weekly | 3.86 (0.08) | 2.18 (0.28) | 0.67 (0.39) | 3.87 (0.08) |
| No Buffer | 4.48 (0.01) | 1.84 (0.15) | 1.21 (0.14) | 4.48 (0.01) |
| Buffer | 3.53 (0.11) | 2.32 (0.13) | 1.23 (0.05) | **3.56 (0.10)** |

Table C.7: KSD results for GARCH on exchange rate data.

| | $\log_{10}$KSD | | | | |
|---|---|---|---|---|---|
| Grad Est. | $\log \mu$ | $\text{logit}\, \lambda$ | $\text{logit}\, \phi$ | $\tau$ | Total |
| Full | 2.18 (0.67) | 2.18 (0.07) | 2.19 (0.61) | 2.07 (0.06) | 2.84 (0.30) |
| Weekly | 2.17 (0.51) | 2.21 (0.03) | 2.31 (0.29) | 1.85 (0.19) | 2.81 (0.21) |
| No Buffer | 1.76 (0.06) | 1.43 (0.46) | 1.31 (0.09) | 1.58 (0.08) | **2.09 (0.09)** |
| Buffer | 1.76 (0.03) | 2.01 (0.08) | 1.11 (0.07) | 1.87 (0.07) | **2.19 (0.05)** |