SIAM J. MATH. DATA SCI.
Vol. 1, No. 3, pp. 555–587

Stochastic Gradient MCMC for State Space Models*

Christopher Aicher†, Yi-An Ma‡, Nicholas J. Foti§, and Emily B. Fox†§

Abstract. State space models (SSMs) are a flexible approach to modeling complex time series. However, inference in SSMs is often computationally prohibitive for long time series. Stochastic gradient Markov chain Monte Carlo (SGMCMC) is a popular method for scalable Bayesian inference for large independent data. Unfortunately, when applied to dependent data, such as in SSMs, SGMCMC's stochastic gradient estimates are biased, as they break crucial temporal dependencies. To alleviate this, we propose stochastic gradient estimators that control this bias by performing additional computation in a "buffer" to reduce breaking dependencies. Furthermore, we derive error bounds for this bias and show a geometric decay under mild conditions. Using these estimators, we develop novel SGMCMC samplers for discrete, continuous, and mixed-type SSMs with analytic message passing. Our experiments on real and synthetic data demonstrate the effectiveness of our SGMCMC algorithms compared to batch MCMC, allowing us to scale inference to long time series with millions of time points.

Key words. stochastic gradient, Markov chain Monte Carlo, Bayesian inference, state space models, hidden Markov models, time series, exponential forgetting

AMS subject classifications. 60J05, 62F15, 62M10, 65C40

DOI. 10.1137/18M1214780

1. Introduction.

State space models (SSMs) are ubiquitous in the analysis of time series in fields as diverse as biology [76], finance and economics [44, 81], and systems and control [30]. As a defining feature, SSMs augment the observed time series with a *latent state sequence* to model complex time series dynamics with a latent Markov chain dependence structure. Given a time series, inference of model parameters involves sampling or marginalizing this latent state sequence. Unfortunately, both the runtime and memory required scale with the length of the time series, which is prohibitive for long time series (e.g., high frequency stock prices [37], genome sequences [29], or neural impulse recordings [19]). In practice, given a long time series, one could "segment" or "downsample" to reduce length; however, this preprocessing can destroy or change important signals, and computational considerations should ideally not limit scientific modeling.

*Received by the editors October 15, 2018; accepted for publication (in revised form) July 5, 2019; published electronically September 24, 2019. This work is an extension of [52].

https://doi.org/10.1137/18M1214780

Funding: This work was supported in part by ONR grants N00014-15-1-2380 and N00014-18-1-2862 and NSF CAREER award IIS-1350133. The third author's research was supported by a Washington Research Foundation Innovation Postdoctoral Fellowship in Neuroengineering and Data Science.

†Department of Statistics, University of Washington, Seattle, WA 98195-4322 (aicherc@uw.edu, ebfox@uw.edu).

‡Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA 94720-1776 (yianma@berkeley.edu).

§Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA 98195-2350 (nfoti@uw.edu).

To help scale inference in SSMs, we consider stochastic gradient Markov chain Monte Carlo (SGMCMC), a popular method for scaling Bayesian inference to large datasets [15, 51, 74]. The key idea of SGMCMC is to employ stochastic gradient estimates based on subsets or "minibatches" of data, avoiding costly computation of gradients on the full dataset, such that the resulting dynamics produce samples from the posterior distribution over SSM parameters. This approach has found much success in *independent* data models, where the stochastic gradients are *unbiased* estimates of the true gradients. However, when applying SGMCMC to SSMs, naive stochastic gradients are *biased*, as subsampling the data breaks dependencies in the SSM's latent state sequence. This bias can destroy the dynamics of SGMCMC, causing it to fail when applied to SSMs. The challenge is to correct these stochastic gradients for SSMs while maintaining the computational benefits of SGMCMC.

In this work, we develop computationally efficient stochastic gradient estimators for inference in general discrete-time SSMs. To control the bias of stochastic gradients, we marginalize the latent state sequence in a *buffer* around each subsequence, propagating critical information from outside each subsequence to its local gradient estimate while avoiding costly full-chain computations. Similar buffering ideas have previously been considered for belief propagation [36], variational inference [31], and in our earlier work on SGMCMC for hidden Markov models (HMMs) [52], but all are limited to discrete latent states. Here, we present buffering as an approximation to *Fisher's identity* [11], allowing us to naturally extend the buffering trick to continuous and mixed-type latent states.

We further develop analytic bounds on the bias of our proposed gradient estimator that, under mild conditions, decay geometrically in the buffer size. To obtain these bounds, we prove that the latent state sequence posterior distribution has an *exponential forgetting* property [11, 20]. However, unlike classic results which prove a geometric decay between the approximate and exact marginal posterior distributions in total variation distance, we use Wasserstein distance [72] to allow the analysis of continuous and mixed-type latent state SSMs. Our approach is similar to proofs of Wasserstein ergodicity in homogeneous Markov chains [28, 53, 64]; however, we extend these ideas to the *nonhomogeneous* Markov chains defined by the latent state sequence posterior distribution. These geometrically decaying bounds guarantee that we only need a small buffer size in practice, allowing scalable inference in SSMs.

Although our proposed gradient estimator can be generally applied to any stochastic gradient method, here we develop SGMCMC samplers for Bayesian inference in a variety of SSMs, such as HMMs, linear Gaussian SSMs (LGSSMs), and switching linear dynamical systems (SLDSs) [11, 33]. We also derive preconditioning matrices to take advantage of information geometry, which allows for more rapid mixing and convergence of our samplers [35, 58]. Finally, we validate our algorithms and theory on a variety of synthetic and real data experiments, finding that our gradient estimator can provide orders of magnitude runtime speed ups compared to batch sampling.

This paper significantly expands upon our initial work [52] by (i) connecting buffering to Fisher's identity, simplifying its presentation and analysis; (ii) nontrivially generalizing the approach to SSMs beyond the HMM, including continuous and mixed-type latent states; (iii) developing a general framework for bounding the error of buffered gradient estimators using Wasserstein distance; and (iv) providing extensive validation on a number of real and synthetic datasets.

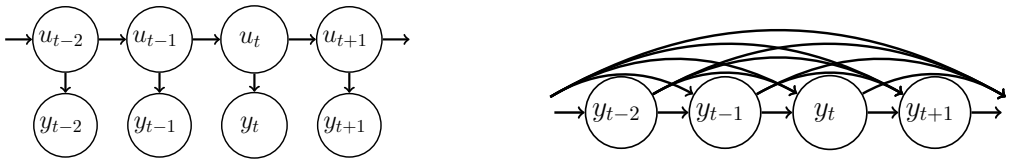

Figure 1. *Graphical model of an SSM: (left) the joint process u, y, (2.1), and (right) y marginalizing out u, (2.2). The parameters θ are not shown but connect to all nodes.*

The paper is organized as follows. First, we review background on SSMs and SGMCMC methods in section 2. We then present our framework of constructing buffered gradient estimators to extend SGMCMC to SSMs in section 3. We prove the geometrically decaying bounds for our proposed buffered gradient estimate in section 4. We apply our framework and error bounds to discrete, continuous, and mixed-type latent state SSMs in section 5. Finally, we investigate our algorithms on both synthetic and real data in section 6.

2. Background.

2.1. State space models for time series. State space models (SSMs) for time series are a class of discrete-time bivariate stochastic process $\{u_t, y_t\}_{t\in\mathcal{T}}$, $\mathcal{T} = \{1, \dots, T\}$, consisting of a latent state sequence $u := u_{1:T}$ generated by a homogeneous Markov chain and an observation sequence $y := y_{1:T}$ generated independently conditioned on u [11]. Examples of state space models include HMMs, LGSSMs, and SLDSs (see section 5 for details). For a generic SSM, the joint distribution of y and u factorizes as

$$p(y, u \,|\, \theta) = \prod_{t=1}^{T} p(y_t \,|\, u_t, \theta) p(u_t \,|\, u_{t-1}, \theta) \cdot p_0(u_0) \ , \tag{2.1}$$

where θ are model-specific parameters, $p(y_t|u_t, \theta)$ is the *emission density*, $p(u_t|u_{t-1}, \theta)$ is the *transition density*, and $p_0(u_0)$ is a prior for the latent states. As the latent state sequence u is unobserved, the likelihood of θ given only the observations y (marginalizing u) is

$$p(y \,|\, \theta) = \int \prod_{t=1}^{T} p(y_t \,|\, u_t, \theta) p(u_t \,|\, u_{t-1}, \theta) \cdot p_0(u_0) \ du \ . \tag{2.2}$$

Unconditionally, the observations y are not independent and the graphical model of this *marginal likelihood*, (2.2), has many long term dependencies; see Figure 1 (right). In contrast, when conditioned on u the observations y are independent and the *complete-data likelihood*, (2.1), has a simpler chain structure; see Figure 1 (left).

To infer θ given y, we can maximize the marginal likelihood $p(y \,|\, \theta)$ or, given a prior $p(\theta)$, sample from the posterior $p(\theta \,|\, y) \propto p(y \,|\, \theta)p(\theta)$. However, traditional inference methods for θ, such as expectation maximization (EM), variational inference, or Gibbs sampling, take advantage of the conditional independence structure in $p(y, u \,|\, \theta)$, (2.1), rather than working directly with $p(y \,|\, \theta)$, (2.2) [6, 65]. To use $p(y, u \,|\, \theta)$ with unobserved u, these methods rely on sampling or taking expectations of u from the posterior $\gamma(u) := p(u \,|\, y, \theta)$. As an example,

gradient-based methods take advantage of *Fisher's identity* [11]

$$\nabla \log p(y \,|\, \theta) = \mathbb{E}_{u|y,\theta}[\nabla \log p(y, u \,|\, \theta)] = \mathbb{E}_{u\sim\gamma}[\nabla \log p(y, u \,|\, \theta)] \ , \tag{2.3}$$

which allows gradients of (2.2) to be computed in terms of (2.1). To compute the posterior $\gamma(u)$, these methods use the well-known *forward-backward algorithm* [11, 65]. The algorithm works by recursively computing a sequence of forward messages $\alpha_t(u_t)$ and backward messages $\beta_t(u_t)$ which are used to compute the pairwise marginals of γ. More specifically,

$$\alpha_t(u_t) := p(u_t, y_{\leq t} \,|\, \theta) = \int p(y_t, u_t \,|\, u_{t-1}, \theta)\alpha_{t-1}(u_{t-1})\, du_{t-1} \ , \tag{2.4}$$

$$\beta_t(u_t) := p(y_{>t} \,|\, u_t, \theta) = \int p(y_{t+1}, u_{t+1} \,|\, u_t, \theta)\beta_{t+1}(u_{t+1})\, du_{t+1} \ , \tag{2.5}$$

$$\gamma_{t-1:t}(u_{t-1}, u_t) := p(u_{t-1}, u_t \,|\, y, \theta) \propto \alpha_{t-1}(u_{t-1})p(y_t, u_t \,|\, u_{t-1}, \theta)\beta_t(u_t) \ . \tag{2.6}$$

When message passing is tractable (i.e., when (2.4)–(2.5) involve discrete or conjugate likelihoods), the forward-backward algorithm can be calculated in closed form. When message passing is intractable, the messages can be approximated using Monte Carlo sampling methods (e.g., blocked Gibbs sampling [12, 32] or particle methods [2, 9, 25, 67]). In both cases, when the length of the time series $|\mathcal{T}|$ is much larger than the dimension of θ, the forward-backward algorithm (running over the entire sequence) requires $O(|\mathcal{T}|)$ time and memory at *each iteration*.

The SSM challenge is to scale inference of model parameters θ to long time series when the computation and storage per iteration $O(|\mathcal{T}|)$ is prohibitive.

2.2. Stochastic gradient MCMC. One popular method for scalable Bayesian inference is *stochastic gradient* Markov chain Monte Carlo (SGMCMC) [15, 51, 74]. The idea behind gradient-based MCMC is to simulate continuous dynamics for a *potential energy* function $U(\theta) \propto -\log p(y, \theta)$ such that the dynamics generate samples from the posterior distribution $p(\theta \,|\, y)$. For example, the Langevin diffusion over $U(\theta)$ is given by the stochastic differential equation (SDE)

$$d\theta_s = g(\theta)ds + \sqrt{2}dW_s \ , \tag{2.7}$$

where dW_s is Brownian motion, $g(\theta) = -\nabla U(\theta) = \nabla_\theta \log p(y, \theta)$, and s indexes continuous time. As $s \to \infty$, the distribution of θ_s converges to the SDE's stationary distribution, which by the Fokker–Planck equation is the posterior $p(\theta \,|\, y)$ [51]. Because we cannot perfectly simulate (2.7), in practice we use a discretized numerical approximation. One straightforward approximation is the Euler–Maruyama discretization

$$\theta^{(s+1)} \leftarrow \theta^{(s)} + hg(\theta^{(s)}) + \mathcal{N}(0, 2h) \ , \tag{2.8}$$

where h is the stepsize and s indexes discrete time steps. This recursive update defines the Langevin Monte Carlo (LMC) algorithm. Typically, a Metropolis–Hastings correction step is added to account for the discretization error [62, 61].

For large datasets, computing $g(\theta)$ at every step in (2.8) is computationally prohibitive. To alleviate this, the key ideas of *stochastic gradient* Langevin dynamics (SGLD) are to replace

$g(\theta)$ with a quick-to-compute unbiased estimator $\hat{g}(\theta)$ and to use a decreasing stepsize $h^{(s)}$ to avoid costly Metropolis–Hastings correction steps [74]:

$$\theta^{(s+1)} \leftarrow \theta^{(s)} + h^{(s)}\hat{g}(\theta^{(s)}) + \mathcal{N}(0, 2h^{(s)}) \ . \tag{2.9}$$

For i.i.d. data, an example of $\hat{g}(\theta)$ is to use a random minibatch $\mathcal{S} \subset \mathcal{T}$, $|\mathcal{S}| \ll |\mathcal{T}|$:

$$\hat{g}(\theta) = -\frac{1}{\Pr(\mathcal{S})} \sum_{t \in \mathcal{S}} \nabla \log p(y_t \,|\, \theta) - \nabla \log p(\theta) \ , \tag{2.10}$$

which only requires $O(|\mathcal{S}|)$ time to compute. When $\hat{g}(\theta)$ is unbiased and with an appropriate decreasing stepsize schedule $h^{(s)}$, the distribution of $\theta^{(s)}$ asymptotically converges to the posterior distribution [15, 68]. However, in practice one uses a small, finite stepsize for greater efficiency, which introduces a small bias [18].

A Riemannian extension of SGLD (SGRLD) simulates the Langevin diffusion over a Riemannian manifold with metric $D(\theta)^{-1}$ by preconditioning the gradient and noise of (2.9) by $D(\theta)$. By incorporating geometric information about structure of θ, SGRLD aims for a diffusion which mixes more rapidly. Suggested examples of the metric $D(\theta)^{-1}$ are the Fisher information matrix $\mathcal{I}(\theta) = \mathbb{E}_y[\nabla^2 \log p(y \,|\, \theta)]$ or a noisy Hessian estimate $\widehat{\nabla^2 \log p}(y \,|\, \theta)$ [35, 58]. Given $D(\theta)$, each step of SGRLD is

$$\theta^{(s+1)} \leftarrow \theta^{(s)} + h\left[D(\theta^{(s)}) \cdot \hat{g}(\theta^{(s)}) + \Gamma(\theta^{(s)})\right] + \mathcal{N}\left(0, 2hD(\theta^{(s)})\right) \ , \tag{2.11}$$

where the vector $\Gamma(\theta)$ is a correction term $\Gamma(\theta)_i = \sum_j \frac{\partial D(\theta)_{ij}}{\partial \theta_j}$ to ensure the dynamics converge to the target posterior [51, 77]. Many extensions to SGMCMC have been proposed, such as using control variates to reduce the variance of $\hat{g}(\theta)$ [5, 14, 55] or augmented dynamics to improve mixing [15, 16, 23, 47]. Although our ideas extend to these formulations as well, we focus on the popular SGLD and SGRLD algorithms.

To apply SGMCMC to SSMs, we must choose whether to use the complete-data loglikelihood or the marginal data loglikelihood in the potential $U(\theta)$. If we use the complete-data loglikelihood, then we treat (u, θ) as the parameters. Although the observations y conditioned on (u, θ) are independent, we must calculate gradients for $u_{-T:T}$ at each iteration, which is prohibitive for long sequences $|\mathcal{T}|$ and intractable for discrete or mixed-type u. On the other hand, if we use the marginal loglikelihood, then we only need to take gradients in θ. However, the observations y conditioned on θ alone are *not* independent, and therefore the minibatch gradient estimator (2.10) breaks crucial dependencies, causing it to be biased. Our SGMCMC challenge is correcting the bias in stochastic gradient estimates $\nabla\widetilde{U}(\theta)$ when applied to SSMs.

3. General framework.

We now present our framework for scalable Bayesian inference in SSMs with long observation sequences. Our approach is to extend SGMCMC to SSMs by developing a gradient estimator that ameliorates the issue of broken temporal dependencies. In particular, we develop a computationally efficient gradient estimator that uses a *buffer* to avoid breaking crucial dependencies, only breaking weak dependencies. We first present a (computationally prohibitive) unbiased estimator of $g(\theta) = \nabla \log p(y \,|\, \theta)$ for SSMs using Fisher's identity. We then derive a general computationally efficient gradient estimate $\tilde{g}(\theta)$

that accounts for the dependence in observations using a buffer. We also propose preconditioning matrices for SGRLD with SSMs. Finally, we present our general SGMCMC pseudocode for SSMs.

3.1. Unbiased gradient estimate. The main challenge in constructing an efficient estimate $\tilde{g}(\theta)$ of $g(\theta)$ for SSMs is handling the lack of independence (marginally) in y. Because the observations in SSMs are not independent, we cannot produce an unbiased estimate of $g(\theta)$ with a randomly selected subset of data points as in (2.10). For example, a naive estimate is to take the gradient of a random contiguous *subsequence* $\mathcal{S} = \{t_1, \dots, t_S\} \subset \mathcal{T}$ with $t_i = t_{i-1} + 1$:

$$\hat{g}(\theta) = -\frac{1}{\Pr(\mathcal{S})} \nabla \log p(y_{\mathcal{S}} \,|\, \theta) - \nabla \log p(\theta) \ , \tag{3.1}$$

where $p(y_{\mathcal{S}} \,|\, \theta)$ is computed with $p(u_{t_0}) = p_0(u_{t_0})$. This estimate only requires $O(S)$ time compared to $O(T)$ for $g(\theta)$. However, because the marginal likelihood does not factorize as in the independent observations case, this estimate is biased: $\mathbb{E}_{\mathcal{S}}[\hat{g}(\theta)] \neq g(\theta)$. In addition, as $\mathcal{S}$ are contiguous subsequences of $\mathcal{T}$, the scaling factor $\Pr(\mathcal{S})^{-1}$ is no longer correct, as time points in the center of $\mathcal{T}$ are sampled more frequently than the endpoints; instead, each time point should be scaled pointwise.

To obtain an unbiased estimate for $g(\theta)$, we use Fisher's identity (2.3) to rewrite $g(\theta)$ in terms of the complete-data loglikelihood as a sum over time points:

$$\begin{aligned} g(\theta) &= -\nabla \log p(y \,|\, \theta) - \nabla \log p(\theta) \\ &= -\mathbb{E}_{u|y,\theta} \left[\nabla \log p(y, u \,|\, \theta)\right] - \nabla \log p(\theta) \\ &= -\sum_{t \in \mathcal{T}} \mathbb{E}_{u|y,\theta} \left[\nabla \log p(y_t, u_t \,|\, u_{t-1}, \theta)\right] - \nabla \log p(\theta) \ . \end{aligned} \tag{3.2}$$

From this, we straightforwardly identify an unbiased estimator for a subsequence $\mathcal{S}$:

$$\bar{g}(\theta) = -\sum_{t \in \mathcal{S}} \frac{1}{\Pr(t \in \mathcal{S})} \mathbb{E}_{u|y,\theta} \left[\nabla \log p(y_t, u_t \,|\, u_{t-1}, \theta)\right] - \nabla \log p(\theta) \ , \tag{3.3}$$

where $\Pr(t \in \mathcal{S})$ is the probability that t is in the random subsequence $\mathcal{S}$.

Although (3.3) reduces the number of gradient terms to compute from T to S, the summation terms require calculating expectations of $u \,|\, y, \theta$. More specifically, (3.3) requires expectations with respect to the pairwise marginal posteriors $p(u_t, u_{t-1} \,|\, y_{\mathcal{T}})$ for $t \in \mathcal{S}$. Recall that computing these marginals takes $O(T)$ time to pass messages over the entire sequence $\mathcal{T}$. This defeats the purpose of using a subsequence. If we instead only pass messages over the subsequence $\mathcal{S}$, then the pairwise marginals are $p(u_t, u_{t-1} \,|\, y_{\mathcal{S}})$ and we return to a biased gradient estimator:

$$\hat{g}(\theta) = -\sum_{t \in \mathcal{S}} \frac{1}{\Pr(t \in \mathcal{S})} \mathbb{E}_{u|y_{\mathcal{S}},\theta} [\nabla \log p(y_t, u_t \,|\, u_{t-1}, \theta)] - \nabla \log p(\theta) \ . \tag{3.4}$$

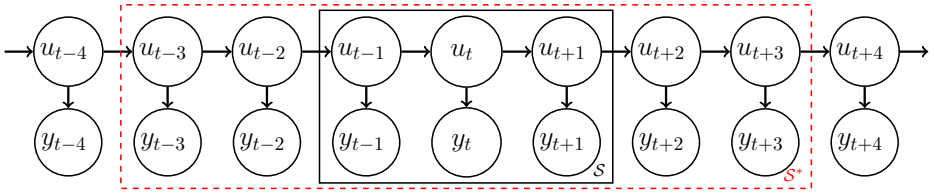

Figure 2. *Graphical model of a buffered subsequence with $S = 3$ and $B = 2$.*

3.2. Approximate gradient estimate. We instead propose passing messages over a *buffered* subsequence $\mathcal{S}^* := \{t_{-B}, \ldots, t_{S+B}\}$ for some positive buffer size B, with $\mathcal{S} \subset \mathcal{S}^* \subset \mathcal{T}$ (see Figure 2). The idea is that there exists a large enough B such that $p(u_\mathcal{S} \,|\, y_{\mathcal{S}^*}, \theta) \approx p(u_\mathcal{S} \,|\, y_\mathcal{T}, \theta)$. Our *buffered gradient estimator* sums only over $\mathcal{S}$ but takes expectations over $u_\mathcal{S} \,|\, y_{\mathcal{S}^*}, \theta$ instead of $u_\mathcal{S} \,|\, y_\mathcal{T}, \theta$:

$$\tilde{g}(\theta) = -\sum_{t \in \mathcal{S}} \frac{1}{\Pr(t \in \mathcal{S})} \mathbb{E}_{u|y_{\mathcal{S}^*},\theta} \left[\nabla \log p(y_t, u_t \,|\, u_{t-1}, \theta)\right] - \nabla \log p(\theta) \ , \tag{3.5}$$

where $p(u_{t_{-B-1}}) = p_0(u_{t_{-B-1}})$. When $B = 0$, this is equivalent to the biased estimator $\hat{g}(\theta)$ of (3.4). When $B = T$, this is equivalent to the unbiased estimator $\bar{g}(\theta)$ of (3.3).

The trade-off between accuracy (bias) and runtime depends on the size of the buffer B and current model parameters $\theta^{(s)}$. Intuitively, when $\theta^{(s)}$ produces pairwise marginals that are similar to i.i.d. data, we can use a small buffer B. When $\theta^{(s)}$ produces strongly dependent pairwise marginals, we must use a larger buffer B. In section 4, we analyze, for a fixed value of θ, how quickly the bias between $\bar{g}(\theta)$ and $\tilde{g}(\theta)$ decays with increasing B. We show a geometric decay

$$\mathbb{E}_\mathcal{S} \|\bar{g}(\theta) - \tilde{g}(\theta)\|_2 \leq C_\theta \rho_\theta^{-B} \quad \text{for some } C_\theta > 0 \ , \tag{3.6}$$

where ρ_θ is large for i.i.d. data and small for strongly dependent data. The term C_θ depends on the smoothness of $g(\theta)$ and how accurately $p_0(u_{t_{-B-1}})$ approximates $p(u_{t_{-B-1}} \,|\, y_{\mathcal{T}\setminus\mathcal{S}^*})$. For a gradient accuracy of ϵ, we only need a logarithmic buffer size $O(\log \epsilon^{-1})$.[1] Therefore, our buffered gradient estimator reduces the computation time from $O(T)$ to $O(S + \log \epsilon^{-1})$. By using buffered stochastic gradients $\tilde{g}$ with an appropriate buffer size B in SGMCMC (see (2.9) or (2.11)), we can generate samples $\theta^{(s)}$ that are close to the samples that would be generated if we were to use the unbiased (but intractable) stochastic gradients $\bar{g}$. In our experiments (section 6), we find that modest buffers significantly correct for bias.

Our approach is similar to fixed-lag smoothing methods in the particle filter literature [13, 21, 56], which approximate $p(u_t \,|\, y_{1:T}, \theta)$ using a right buffer $p(u_t \,|\, y_{1:t+B}, \theta)$ in a streaming fashion. However, our approach, (3.5), differs by using both a left and a right buffer $p(u_t \,|\, y_{1:T}, \theta) = p(u_t \,|\, y_{t-B:t+B})$, which allow us to avoid a full pass over the data.

3.3. Preconditioning and Fisher information. The desirable properties for the preconditioning matrix $D(\theta)$ for SGRLD are (i) the resulting dynamics take advantage of the geometric

[1] As $\epsilon \geq C_\theta \rho_\theta^{-B} \Rightarrow B \geq -\log \epsilon / \log \rho_\theta + \log C_\theta / \log \rho_\theta \Rightarrow B$ is $O(\log \epsilon^{-1})$.

structure of θ, (ii) both $D(\theta)$ and $\Gamma(\theta)$ can be efficiently computed, and (iii) neither $D(\theta)g(\theta)$ nor $\Gamma(\theta)$ is numerically unstable.

The *expected Fisher information* $\mathcal{I}_y$ is the Riemannian metric proposed in [35]:

$$D^{-1}(\theta) = \mathcal{I}_y = \mathbb{E}_{y\,|\,\theta}\left[\nabla^2 \log p(y\,|\,\theta)\right] \quad . \tag{3.7}$$

Unfortunately for SSMs, the lack of independence in the marginal likelihood requires a double sum over $\mathcal{T}$ to compute $\mathcal{I}_y$, which is computationally intractable for long time series. We instead replace I_y with the *complete data Fisher information* $I_{u,y}$:

$$\mathcal{I}_{u,y} = \mathbb{E}_{u,y\,|\,\theta}\left[\nabla^2 \log p(y, u \;\; \theta)\right] = T \cdot \mathbb{E}_{u,y\,|\,\theta}\left[\nabla^2 \log p(y_t, u_t\,|\,u_{t-1}, \theta)\right] \quad . \tag{3.8}$$

Because $I_{u,y}$ can be calculated analytically for the SSMs we consider (section 5), we use $D(\theta) = I_{u,y}^{-1}$ when possible or approximations of $I_{u,y}^{-1}$ when not (see the supplementary materials (M121478_01.pdf [local/web 1.18MB]), linked from the main article webpage, for details). In our experiments, we find that in practice, using preconditioning works well and outperforms vanilla SGLD.

3.4. Algorithm pseudocode. Algorithms 3.1 and 3.2 summarize our generic SGMCMC method for SSMs.[2]

Algorithm 3.1. SGRLD.

Input: data y, parameters $\theta^{(0)}$, stepsize h, subsequence length S, error tolerance ϵ
for $s = 0, 1, 2, \ldots, N_{\text{steps}} - 1$ **do**
  $\tilde{g}(\theta^{(s)}) = \texttt{NoisyGradient}(y, \theta^{(s)}, S, \epsilon)$ // Algorithm 3.2 or 5.1
  $D^{(s)}, \Gamma^{(s)} = \texttt{GetPreconditioner}(\theta^{(s)})$ // e.g., (3.7)
  $\theta^{(s+1)} \leftarrow \theta^{(s)} + h^{(s)}\left[D^{(s)}\tilde{g}(\theta^{(s)}) + \Gamma^{(s)}\right] + \mathcal{N}\left(0, 2h^{(s)}D^{(s)}\right)$ // (2.11)
end for
Return $\theta^{(N_{\text{steps}})}$

Algorithm 3.2. `NoisyGradient` for analytic message passing.

Input: data y, parameters θ, subsequence length S, error tolerance ϵ
$B = \texttt{BufferSize}(\theta, S, \epsilon)$
$\mathcal{S}, \mathcal{S}^* = \texttt{GetBufferedSubsequence}(y, S, B)$
$p(u_{\mathcal{S}}\,|\,y_{\mathcal{S}^*}, \theta) = \texttt{ForwardBackward}(y, \mathcal{S}^*, \theta)$ // Message Passing
$\tilde{g}(\theta) = -\sum_{t\in\mathcal{S}} \frac{1}{\Pr(t\in\mathcal{S})}\, \mathbb{E}_{u_{\mathcal{S}}|y_{\mathcal{S}^*},\theta}[\nabla_\theta \log p(y_t, u_t|u_{t-1})]$ // (3.5)
Return $\tilde{g}(\theta)$

To select the buffer size B in Algorithm 3.2, we choose B large enough such that the error using B and a larger buffer size B^* is small:

$$B = \min\left\{\hat{B} \in [0, B^*] \;:\; \mathbb{E}_{\mathcal{S}}\|\,\tilde{g}(\theta, \mathcal{S}, \hat{B}) - \tilde{g}(\theta, \mathcal{S}, B^*)\| < \epsilon\right\} \quad , \tag{3.9}$$

[2]Python code for our method is available online from https://github.com/aicherc/sgmcmc_ssm_code.

where $\tilde{g}(\theta, \mathcal{S}, B) = \mathbb{E}_{u|y_{\mathcal{S}^*},\theta}[\nabla \log p(y_{\mathcal{S}}, u_{\mathcal{S}} \,|\, \theta)]$ and the expectation over $\mathcal{S}$ is approximated with an empirical average over N_S subsequences. Equation (3.9) uses $\tilde{g}(\theta, \mathcal{S}, B^*)$ as a proxy for $\tilde{g}(\theta, \mathcal{S}, T)$. As the error decays geometrically (section 4), we found that using $B^* = 100$ was conservative in practice. Calculating B using (3.9) at every iteration for a new $\theta^{(s)}$ is impractical; therefore, for our experiments, we use a fixed B, estimated using θ from a pilot run with $B = B^*$ and $N_S = 1000$. In addition, instead of evaluating each $\hat{B}$ in $[0, B^*]$, we can estimate the required B for a target error tolerance ϵ after estimating the error $\hat{\epsilon}$ of a single $\hat{B}$ by taking advantage of the geometric error scaling rate, (3.6), to obtain $B = \hat{B} + \log_{\rho_\theta}(\hat{\epsilon}/\epsilon)$, where ρ_θ is a bound on the geometric decay rate from theory.

4. Buffered gradient estimator error bounds. In this section, we establish a bound on the expected error between the unbiased gradient $\bar{g}(\theta)$ and our buffered gradient estimator $\tilde{g}(\theta)$, (3.5). Given such a bound, we can control the overall error in our SGLD or SGRLD scheme when the SGMCMC dynamics possess a contraction property [40]. Specifically, if we can uniformly bound $\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2 < \delta$, then the difference in a single step of SGMCMC, (2.11), using the unbiased and approximate gradients $\bar{g}$ and $\tilde{g}$ is bounded by δh. Therefore, we can apply Theorem 1.11 of [40], which states that the sample average of a test function evaluated on samples of the approximate-gradient $\tilde{g}$ chain, $\sum_{i<s} \varphi(\theta^{(i)})/s$, converges to the posterior expected value of the unbiased-gradient $\bar{g}$ chain, $\mathbb{E}_\theta[\varphi(\theta)]$, plus an additional error term proportional to δh. For our analysis, we first consider the simple case of uniformly sampling a single sequence from T/S separate subsequences (i.e., $\Pr(t \in \mathcal{S}) = S/T$ for all t) and assume the prior p_0 is stationary (i.e., $p_0(u_t) = \int p(u_t|u_{t-1})p_0(u_{t-1})du_{t-1}$).

Our approach is to bound $\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2$ in terms of the Wasserstein distance between the exact posterior $\gamma_t(u_t) = p(u_t \,|\, y_{\mathcal{T}}, \theta)$ and our approximate posterior $\widetilde{\gamma}_t(u_t) = p(u_t \,|\, y_{\mathcal{S}^*}, \theta)$ and then show that this Wasserstein distance decays geometrically. To bound the Wasserstein distance, we follow existing work on bounding Markov processes in Wasserstein distance [28, 53, 64]. However, unlike previous work that focuses on the homogeneous Markov process of the joint model $\{u, y \,|\, \theta\}$, we instead focus on the induced *nonhomogeneous* Markov process of the conditional model $\{u \,|\, y, \theta\}$. To do so, we use the forward (f_t) and backward (b_t) *random maps* of $\{u \,|\, y, \theta\}$ [22]:

$$u_t \sim p(u_t \,|\, y, \theta) \;\Rightarrow\; (f_t(u_t), u_t) \sim p(u_{t+1}, u_t \,|\, y, \theta)\ , \tag{4.1}$$

$$u_t \sim p(u_t \,|\, y, \theta) \;\Rightarrow\; (b_t(u_t), u_t) \sim p(u_{t-1}, u_t \,|\, y, \theta)\ . \tag{4.2}$$

If f_t and b_t satisfy a contractive property, then we can bound the Wasserstein distance between $\gamma_t, \widetilde{\gamma}_t$ in terms of $\gamma_{t-1}, \widetilde{\gamma}_{t-1}$ and $\gamma_{t+1}, \widetilde{\gamma}_{t+1}$, respectively. Bounding the error of the induced nonhomogeneous Markov process has been previously studied in the SSM literature using total variation (TV) distance [11, 20, 46, 69]. These works bound the error in TV distance by quantifying how quickly the smoothed posterior forgets the initial condition. However, these bounds typically require stringent regularity conditions, which are hard to prove outside of finite or compact spaces.[3] In particular, these bounds are not immediately applicable for LGSSMs. In contrast, we bound the error in Wasserstein distance by proving contraction

[3]These bounds have been extended to noncompact spaces for the *filtered* posterior, when the SSM satisfies a multiplicative drift condition [75].

properties of f_t and b_t, allowing us to handle continuous and mixed-type SSMs such as the LGSSM (section 5.3.1).

Our main result is that if, for each fixed θ, the gradient of $\log p(y, u \mid \theta)$ satisfies a Lipschitz condition and the random maps $\{f_t, b_t\}_{t \in \mathcal{S}^*}$ all satisfy a contraction property, then the error $\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2$ decays geometrically in the buffer size B.

Theorem 4.1. *Let $\epsilon_{\rightarrow}$ and $\epsilon_{\leftarrow}$ be the 1-Wasserstein distances between γ_t and $\widetilde{\gamma}_t$ at the left and right ends of $\mathcal{S}^*$, respectively. Let $\epsilon_1 = \max_{\mathcal{S}^* \subset \mathcal{T}} \{\epsilon_{\rightarrow}, \epsilon_{\leftarrow}\}$. If the gradients of $\log p(y_t, u_t \mid u_{t-1}, \theta)$ are all Lipschitz in $u_{t-1:t}$ with constant L_U, and random maps f_t and b_t are all Lipschitz[4] in u_t with constant $L < 1$, then we have*

$$\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2 \leq T \cdot L_U \cdot \frac{1+L}{1-L} \cdot \frac{1-L^S}{S} \cdot L^B \cdot 2\epsilon_1 \quad . \tag{4.3}$$

A similar result for when the gradient of the complete data loglikelihood is Lipschitz in uu^T instead of u (as needed for LGSSM) will be proved in section 4.3.

As $L < 1$, Theorem 4.1 states that the error of the buffered gradient estimator decays geometrically as $O(L^B)$. Therefore, the required buffer size B for an error tolerance of δ scales logarithmically as $O(\log \delta^{-1})$. In contrast, the error of the gradient estimator decays only linearly in the subsequence length, $O(S^{-1})$; therefore, much longer subsequences, $O(\delta^{-1})$, are required to reduce bias. This agrees with the intuition that the bias is dominated by the error at the endpoints of subsequence.

Theorem 4.1 requires bounding the Lipschitz constants of the gradient of the complete data loglikelihood and the random maps f_t, b_t given the parameters θ and observations $y_{\mathcal{T}}$. We show examples of these bounds for specific models in section 5.1.1 (HMMs) and 5.3.1 (LGSSMs). Theorem 4.1 also depends on the maximum Wasserstein distance ϵ_1 between γ_t and $\widetilde{\gamma}_t$ for all $\mathcal{S}^* \subset \mathcal{T}$ and $t \in \mathcal{T}$, which is finite.

The remainder of this section is as follows. First, in section 4.1, we show how to bound the error in $\bar{g}, \tilde{g}$ in terms of Wasserstein distances between $\gamma, \widetilde{\gamma}$. Second, in section 4.2, we show that these Wasserstein distances decay geometrically in B. Finally, in section 4.3, we prove our main results, Theorems 4.1 and 4.5, and discuss relaxations of the assumptions on the sampling of subsequences $\mathcal{S}$ and the prior p_0. To keep the presentation clean, we leave proofs of the lemmas to the supplementary materials.

4.1. Functional bound in terms of Wasserstein.

We first review the definition of Wasserstein distance. Let $\mathcal{W}_p(\gamma, \widetilde{\gamma})$ be the p-Wasserstein distance:

$$\mathcal{W}_p(\gamma, \widetilde{\gamma}) := \left[\inf_{\xi} \int \|u - \widetilde{u}\|_2^p \, d\xi(u, \widetilde{u}) \right]^{1/p} \quad , \tag{4.4}$$

where ξ is a joint measure or *coupling* over $(u, \widetilde{u})$ with marginals $\int_{\widetilde{u}} d\xi(u, \widetilde{u}) = d\gamma(u)$ and $\int_u d\xi(u, \widetilde{u}) = d\widetilde{\gamma}(\widetilde{u})$. Wasserstein distance satisfies all the properties of a metric. A useful property of the 1-Wasserstein distance is the following Kantorovich–Rubinstein duality

[4]The random mapping ψ is Lipschitz with constant L if $\mathbb{E}_{\psi} \|\psi(u) - \psi(u')\|_2 \leq L \|u - u'\|_2$ for all u, u'.

formula for the difference of expectations of Lipschitz functions [72]:

$$\mathcal{W}_1(\gamma,\widetilde{\gamma}) = \sup_{\|\psi\|_{Lip}\le 1}\left\{\int \psi\, d\gamma - \int \psi\, d\widetilde{\gamma}\right\} \;\Rightarrow\; |\mathbb{E}_\gamma[\psi] - \mathbb{E}_{\widetilde{\gamma}}[\psi]| \le \|\psi\|_{Lip}\cdot \mathcal{W}_1(\gamma,\widetilde{\gamma})\ , \tag{4.5}$$

where $\|\psi\|_{Lip}$ denotes the Lipschitz constant of ψ.

We connect the error $\|\bar{g}-\tilde{g}\|_2$ to the Wasserstein distances between $\gamma,\widetilde{\gamma}$, by applying this duality formula (4.5) to the difference of (3.3) and (3.5):

$$\bar{g}(\theta) - \tilde{g}(\theta) = \frac{T}{S}\sum_{t\in\mathcal{S}} \mathbb{E}_{\gamma_{t-1:t}}\left[\nabla \log p(y_t,u_t|u_{t-1},\theta)\right] - \mathbb{E}_{\widetilde{\gamma}_{t-1:t}}\left[\nabla \log p(y_t,u_t|u_{t-1},\theta)\right]. \tag{4.6}$$

Applying the triangle inequality gives Lemma 4.2.

Lemma 4.2. *If $\nabla \log p(y_t,u_t|u_{t-1},\theta)$ are Lipschitz in $u_{t-1:t}$ with constant L_U, then*

$$\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2 \le \frac{T}{S}\cdot L_U \cdot \sum_{t\in\mathcal{S}} \mathcal{W}_1(\gamma_{t-1:t},\widetilde{\gamma}_{t-1:t}). \tag{4.7}$$

If $\nabla \log p(y_t,u_t\,|\,u_{t-1},\theta)$ is not Lipschitz in $u_{t-1:t}$, but is Lipschitz in $u_{t-1:t}u_{t-1:t}^T$ (as in LGSSMs), then the following lemma lets us bound the 1-Wasserstein distance of uu^T in terms of the 2-Wasserstein distance of u.

Lemma 4.3. *Let γ' be the distribution of uu^T. Let $\widetilde{\gamma}'$ be the distribution of $\widetilde{u}\widetilde{u}^T$. Let $M = \mathbb{E}_\gamma[\|u\|_2^2] < \infty$. (Note $\mathcal{W}_2(\gamma,\widetilde{\gamma}) < \infty$ implies $\mathbb{E}_\gamma[\|u\|_2^2] < \infty$.) Then,*

$$\mathcal{W}_1(\gamma',\widetilde{\gamma}') \le (2\sqrt{M}+1)\cdot \max\left\{\mathcal{W}_2(\gamma,\widetilde{\gamma})^{1/2}, \mathcal{W}_2(\gamma,\widetilde{\gamma})\right\}\ .$$

4.2. Geometric Wasserstein decay. We first review why contractive random maps induce Wasserstein bounds. If two distributions γ_t,γ_t' have identically distributed random maps f_t, f_t', that is, there exists a random function f_t satisfying

$$u\sim\gamma_t \text{ and } u'\sim\gamma_t' \;\Rightarrow\; f_t(u)\sim\gamma_{t+1} \text{ and } f_t(u')\sim\gamma_{t+1}'\ , \tag{4.8}$$

then we can bound the Wasserstein distance of $\gamma_{t+1},\gamma_{t+1}'$ in terms of the Wasserstein distance of γ_t,γ_t' given a bound on the random map's Lipschitz constant $\|f_t\|_{Lip} < L$:

$$\begin{aligned}\mathcal{W}_p(\gamma_{t+1},\gamma_{t+1}')^p &= \inf_{\xi_{t+1}}\int \|u_{t+1}-u_{t+1}'\|_2^p\, d\xi_{t+1}(u_{t+1},u_{t+1}')\\ &\le \inf_{\xi_t}\int \|f_t(u_t)-f_t(u_t')\|_2^p\, d\xi_t(u_t,u_t')df_t\\ &\le \inf_{\xi_t}\int L^p\cdot\|u_t-u_t'\|_2^p\, d\xi_t(u_t,u_t') = L^p\cdot\mathcal{W}_p(\gamma_t,\gamma_t')^p\ .\end{aligned} \tag{4.9}$$

Unfortunately for SSMs, (4.9) does not apply, as the random maps f_t, b_t of γ and $\widetilde{f}_t,\widetilde{b}_t$ of $\widetilde{\gamma}$ are *not identically* distributed. To see this, we first review the conditional probability

distributions used to define f_t, b_t. The forward random map f_t draws $u_{t+1} \mid u_t$ from the *forward smoothing kernel*

$$\mathcal{F}_t(u_{t+1} \mid u_t) := p(u_{t+1} \mid u_t, y_{>t}) = p(u_{t+1} \mid u_t)p(y_{t+1} \mid u_{t+1})\beta_{t+1}(u_{t+1})/\beta_t(u_t) \tag{4.10}$$

and the backward random map b_t draws $u_{t-1} \mid u_t$ from the *backward smoothing kernel*

$$\mathcal{B}_t(u_{t-1} \mid u_t) := p(u_{t-1} \mid u_t, y_{\geq t}) = p(u_t \mid u_{t-1})p(y_t \mid u_t)\alpha_{t-1}(u_{t-1})/\alpha_t(u_t) \ . \tag{4.11}$$

Because $\widetilde{\gamma}$ uses different forward and backward messages $\widetilde{\alpha}$, $\widetilde{\beta}$ in (4.10) and (4.11), the kernels $\widetilde{\mathcal{F}}_t, \widetilde{\mathcal{B}}_t$ are not identical to $\mathcal{F}_t, \mathcal{B}_t$ (and the random maps are *not* identically distributed). This is unlike homogeneous Markov chains, where the kernels are identical at each time t (and the random maps are identically distributed).

Instead of connecting γ to $\widetilde{\gamma}$ directly, we use the triangle inequality to connect them through an intermediate distribution $\widehat{\gamma} := p(u \mid y_{t \geq t_{-B}}, \theta)$:

$$\mathcal{W}_p(\gamma, \widetilde{\gamma}) \leq \mathcal{W}_p(\gamma, \widehat{\gamma}) + \mathcal{W}_p(\widehat{\gamma}, \widetilde{\gamma}) \ . \tag{4.12}$$

Introducing this particular intermediate distribution $\widehat{\gamma}$ is the key step for our Wasserstein bounds between γ and $\widetilde{\gamma}$. Because $\widehat{\gamma}$ conditions on all y_t after $y_{\mathcal{S}^*}$, $\widehat{\gamma}$ and γ have identical backward messages β_t and therefore identically distributed forward random maps f_t. Similarly, because $\widehat{\gamma}$ does not condition on y_t before $y_{\mathcal{S}^*}$, $\widehat{\gamma}$ and $\widetilde{\gamma}$ have identical forward messages $\widetilde{\alpha}_t$ and identically distributed backward random maps $\widetilde{b}_t$.

Therefore, we can bound $\mathcal{W}_p(\gamma, \widehat{\gamma})$ using f_t and bound $\mathcal{W}_p(\widehat{\gamma}, \widetilde{\gamma})$ using $\widetilde{b}_t$ with the contraction trick (4.9), giving us Lemma 4.4.

Lemma 4.4. *If there exist $L_f, L_b < 1$ such that for all $t \in \mathcal{S}^*$, $\|f_t\|_{Lip} < L_f$ and $\|\widetilde{b}_t\|_{Lip} < L_b$, then for all $t \in \mathcal{S}$ we have*

$$\begin{aligned} \mathcal{W}_p(\gamma_{t-1:t}, \widehat{\gamma}_{t-1:t}) &\leq (1 + L_f^p)^{1/p} \cdot \mathcal{W}_p(\gamma_{t-1}, \widehat{\gamma}_{t-1}) \\ &\leq (1 + L_f^p)^{1/p} \cdot L_f^{t-1-t_{-B}} \cdot \mathcal{W}_p(\gamma_{t_{-B}}, \widehat{\gamma}_{t_{-B}}) \ , \end{aligned} \tag{4.13}$$

$$\begin{aligned} \mathcal{W}_p(\widehat{\gamma}_{t-1:t}, \widetilde{\gamma}_{t-1:t}) &\leq (1 + L_b^p)^{1/p} \cdot \mathcal{W}_p(\widehat{\gamma}_t, \widetilde{\gamma}_t) \\ &\leq (1 + L_b^p)^{1/p} \cdot L_b^{t_{S+B}-t} \cdot \mathcal{W}_p(\widehat{\gamma}_{t_{S+B}}, \widetilde{\gamma}_{t_{S+B}}) \ . \end{aligned} \tag{4.14}$$

We show sufficient conditions for the random maps to be contractions (i.e., $L_f, L_b < 1$) for specific models in section 5.1.1 (HMMs) and 5.3.1 (LGSSMs).

4.3. Proofs of main theorems. Putting together the results of the previous two subsections gives us our geometric error bounds: Theorem 4.1 when the gradient terms are Lipschitz in u and Theorem 4.5 when the gradient terms are Lipschitz in uu^T. Both theorems require the random maps of the forward and backward smoothing kernels to be contractions. We first prove Theorem 4.1.

Proof of Theorem 4.1. Combining Lemmas 4.2 and 4.4 with some algebra,

$$\begin{aligned}\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2 &\le \frac{T}{S} \cdot L_U \cdot \sum_{t \in \mathcal{S}} \mathcal{W}_1(\gamma_{t-1:t}, \widetilde{\gamma}_{t-1:t}) \\ &\le \frac{T}{S} \cdot L_U \cdot \sum_{t \in \mathcal{S}} \mathcal{W}_1(\gamma_{t-1:t}, \widehat{\gamma}_{t-1:t}) + \mathcal{W}_1(\widehat{\gamma}_{t-1:t}, \widetilde{\gamma}_{t-1:t}) \\ &\le \frac{T}{S} \cdot L_U \cdot \sum_{t=1}^{S} (1 + L_f) L_f^{B+t-1} \epsilon_1 + (1 + L_b) L_b^{B+S-t} \epsilon_1 \\ &\le T \cdot L_U \cdot \frac{1+L}{1-L} \cdot \frac{1 - L^S}{S} \cdot L^B \cdot 2\epsilon_1 \ ,\end{aligned}$$

where $\max_{\mathcal{S}^* \subset \mathcal{T}} \{\mathcal{W}_1(\gamma_{t_-B}, \widehat{\gamma}_{t_-B}), \mathcal{W}_1(\widehat{\gamma}_{t_{S+B}}, \widetilde{\gamma}_{t_{S+B}})\} = \max_{\mathcal{S}^* \subset \mathcal{T}} \{\epsilon_{\rightarrow}, \epsilon_{\leftarrow}\} = \epsilon_1$. ∎

We now prove a similar result for when $\nabla \log p(y, u_t \,|\, u_{t-1}\theta)$ is Lipschitz in uu^T.

Theorem 4.5. *Let* $\epsilon_2 = \max_{\mathcal{S}^* \subset \mathcal{T}} \{\mathcal{W}_2(\gamma_{t_-B}, \widehat{\gamma}_{t_-B}), \mathcal{W}_2(\widehat{\gamma}_{t_{S+B}}, \widetilde{\gamma}_{t_{S+B}})\}$. *If the gradients* $\nabla \log p(y_t, u_t \,|\, u_{t-1}, \theta)$ *are Lipschitz in* uu^T *with constant* L'_U, *and there exist* $L_f, L_b < 1$ *for Lemma* 4.4, *then with* $L = \max\{L_f, L_b\}$ *and* $L_U = (2\sqrt{\mathbb{E}_\gamma \|u\|_2^2} + 1) L'_U$ *we have*

$$\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2 \le T \cdot L_U \cdot \frac{\sqrt{1+L^2}}{1 - L^{1/2}} \cdot \frac{1 - L^{S/4}}{S/2} \cdot L^{B/2} \cdot \max_{r \in \{1/2, 1\}} (2\epsilon_2)^r \, .$$

Similar to Theorem 4.1, Theorem 4.5 states that the squared error of the buffered gradient estimator decays geometrically if the complete-data loglikelihood is Lipschitz in uu^T instead of u. However, the price we pay is a square root: the error decays $O(L^{B/2})$ instead of $O(L^B)$.

Proof of Theorem 4.5. Applying Lemmas 4.3 and 4.4, we have

$$\begin{aligned}\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2 &\le \frac{T}{S} \cdot L_U \cdot \sum_{t \in \mathcal{S}} \max_{r \in \{1/2, 1\}} [\mathcal{W}_2(\gamma_{t-1:t}, \widehat{\gamma}_{t-1:t}) + \mathcal{W}_2(\widehat{\gamma}_{t-1:t}, \widetilde{\gamma}_{t-1:t})]^r \\ &\le \frac{T}{S} \cdot L_U \cdot \sum_{t=1}^{S} \max_{r \in \{1/2, 1\}} \left[(L^{B+t-1} + L^{B+S-t}) \sqrt{1+L^2} \epsilon_2 \right]^r \\ &\le \frac{T}{S} \cdot L_U \cdot \sum_{t=1}^{S} L^{(B + \min\{t-1, S-t\})/2} \cdot \sqrt{1+L^2} \cdot \max_{r \in \{1/2, 1\}} (2\epsilon_2)^r \\ &\le \frac{T}{S} \cdot L_U \cdot 2 \cdot \frac{1 - L^{S/4}}{1 - L^{1/2}} \cdot L^{B/2} \cdot \sqrt{1+L^2} \cdot \max_{r \in \{1/2, 1\}} (2\epsilon_2)^r \ .\end{aligned}$$

∎

Our error analysis (Theorems 4.1 and 4.5) indicates that only a logarithmic buffer size is required to control the bias to a fixed error tolerance δ.

4.3.1. Relaxations of assumptions. We now briefly discuss relaxations of the assumptions on $\Pr(t \in \mathcal{S})$ and p_0.

If the contiguous subsequences are not sampled from a strict partition (i.e., $\Pr(t \in \mathcal{S}) \neq S/T$ for all t), then we can replace the factor of T/S in Theorems 4.1 and 4.5 with

$\max_t \Pr(t \in \mathcal{S})^{-1}$. Additional details on different sampling methods for $\mathcal{S}$ can be found in the supplementary materials.

If the initial distribution for $u_{t_{-B-1}}$ of our buffered stochastic gradient, p_0, is not stationary, then our approximate posterior over the latent states $\tilde{\gamma}_t(u_t)$ is not equal to $p(u_t \,|\, y_{\mathcal{S}^*}, \theta)$. However, Theorems 4.1 and 4.5 will still apply; the choice of initial distribution only affects the Wasserstein distance between $\gamma_t, \tilde{\gamma}_t$ and therefore the terms ϵ_1, ϵ_2 in the theorems. In fact, the optimal initial distribution is $p(u_{t_{-B}} | y_{\mathcal{T} \setminus \mathcal{S}^*})$, which minimizes the Wasserstein distance of $\gamma, \tilde{\gamma}$.

5. Example models.

In this section, we provide examples of how to apply the generic framework of section 3 and the bounds of section 4 to common SSMs.

5.1. Gaussian HMM.

We consider discrete latent state HMMs with Gaussian emissions. The complete-data likelihood of a Gaussian HMM is as follows:

$$p(y, z \,|\, \theta) = \prod_{t=1}^{T} \Pi_{z_{t-1}, z_t} \cdot \mathcal{N}(y_t \,|\, \mu_{z_t}, \Sigma_{z_k}) \ , \tag{5.1}$$

where $y_t \in \mathbb{R}^m$ are the observations, $u_t \equiv z_t \in \{1, \dots, K\}$ are the discrete latent variables, and $\theta = \{\Pi, \mu, \Sigma\}$ are the parameters with $\Pi_k \in \Delta^K$ (simplex over K states), $\mu_k \in \mathbb{R}^m$, and $\Sigma_k \in \mathbb{S}_+^m$ (positive definite matrices) for $k = 1, \dots, K$. In practice, we use the *expanded mean* parameters of Π instead of Π (as in [58]) and the *Cholesky decomposition* of Σ_k^{-1} instead of Σ_k to ensure positive definiteness. As the latent states are discrete over a finite space, the forward backward algorithm for an HMM can be done in closed form; thus, pairwise latent marginals $\gamma_{t-1:t}(z_{t-1}, z_t)$, gradients $\nabla U(\theta)$, and preconditioning terms $D(\theta)$ and $\Gamma(\theta)$ are straightforward to calculate. Complete details are provided in the supplementary materials.

5.1.1. Error bound coefficients.

In the finite discrete variable case, conditions for bounding the Lipschitz coefficient of the smoothing kernels $\mathcal{F}_t, \mathcal{B}_t$ (as needed for section 4.2) are equivalent to conditions for bounding their *Dobrushin coefficients* [11, 20]. The Dobrushin coefficient for a transition kernel $\mathcal{Q}$ is

$$\delta(\mathcal{Q}) = \sup_{z, z'} \frac{1}{2} \|\mathcal{Q}(z, \cdot) - \mathcal{Q}(z', \cdot)\|_{TV} = \frac{\|\mathcal{Q}(z, \cdot) - \mathcal{Q}(z', \cdot)\|_{TV}}{\|\delta_z - \delta_{z'}\|_{TV}} \ . \tag{5.2}$$

The final term of (5.2) shows the connection between Dobrushin coefficients and Lipschitz coefficients: it is the ratio of the distance of between kernels $\mathcal{Q}(z, \cdot), \mathcal{Q}(z', \cdot)$ with the distance between point masses at z and z'. Therefore, for discrete latent states, $L_f = \max_t \delta(\mathcal{F}_t)$ and $L_b = \max_t \delta(\mathcal{B}_t)$.

In the discrete case, sufficient conditions for $L_f, L_b < 1$ are well known (see [11, Chapter 4.3]). If the transition matrix Π satisfies the *strong mixing condition*, that is, there exist constants σ^- and σ^+ with $0 < \sigma^- \leq \sigma^+$ and a probability distribution $\kappa \in \Delta^K$ over z such that $\sigma^- \kappa(z') \leq \Pi_{z,z'} \leq \sigma^+ \kappa(z')$ and $\mathbb{E}_\kappa[p(y \,|\, z)] < \infty$, then the Dobrushin coefficients are bounded by $L = 1 - \sigma^-/\sigma^+$. Relaxations of this condition can be found in [11, 20]. Alternatively, we can obtain tighter bounds for HMMs via estimating the Lyapunov exponents for the underlying random dynamical systems defined by random maps f_t and b_t [79, 52].

Finally, the Lipschitz constant L_U for Lemma 4.2 is

$$L_U = \max_{t \in \mathcal{S}, z_t, z_t'} \|\nabla \log p(y_t, z_t \,|\, z_{t-1}, \theta) - \nabla \log p(y_t, z_t' \,|\, z_{t-1}', \theta)\| \ . \tag{5.3}$$

This is easy to compute since at each iteration y and $\theta = \theta^{(s)}$ are fixed. Given these bounds on L_U and L, we can use Theorem 4.1 to select the buffer size B to ensure approximate convergence to the stationary distribution.

5.2. Autoregressive HMM. We now consider autoregressive hidden Markov models (ARHMMs), a generalization of the discrete state HMM where each observation depends not only on the latent state but also on the last p observations. Specifically, the discrete latent state z_t determines which AR(p) process models the dynamics of y at time t. The complete-data likelihood of an ARHMM is as follows:

$$p(y, z \,|\, \theta) = \prod_{t=1}^{T} \Pi_{z_{t-1}, z_t} \cdot \mathcal{N}(y_t \,|\, A_{z_t} \overline{y_t}, Q_{z_k}) \ , \tag{5.4}$$

where $y_t \in \mathbb{R}^m$ are the observations, $\overline{y_t} = y_{t-1:t-p}$ are the p-lagged observations, $u_t \equiv z_t \in \{1, \ldots, K\}$ are the discrete latent variables, and $\theta = \{\Pi, A, Q\}$ are the parameters with $\Pi_k \in \Delta^K$, $A_k \in \mathbb{R}^{m \times mp}$, and $Q_k \in \mathbb{S}_+^m$ for $k = 1, \ldots, K$. From (5.4), we see that the ARHMM is a time-dependent mixture of K AR processes of order p. The pairwise latent marginals, gradients, and preconditioning terms for an ARHMM are calculated similarly to the Gaussian HMM. Further details are provided in the supplementary materials. The theory and constants for the error bounds of section 4 are identical to those presented for the Gaussian HMM.

5.3. Linear Gaussian SSM. A linear Gaussian SSM (LGSSM), also called a linear dynamical system (LDS), consists of a latent Gaussian (vector) AR process over states $u_t \equiv x_t \in \mathbb{R}^n$ and conditionally Gaussian emissions $y_t \in \mathbb{R}^m$ [8, 50]. Specifically,

$$p(y, x \,|\, \theta) = \prod_{t=1}^{T} \mathcal{N}(x_t \,|\, A x_{t-1}, Q) \cdot \mathcal{N}(y_t \,|\, C x_t, R) \ , \tag{5.5}$$

where $A \in \mathbb{R}^{n \times n}$ is the latent state transition matrix, $Q \in \mathbb{S}_+^n$ is the transition noise covariance, $C \in \mathbb{R}^{m \times n}$ is the emission matrix, and $R \in \mathbb{S}_+^m$ is the emission noise covariance. Together, A, Q, C, R are the model parameters θ. The matrices A, C, and Q are unidentifiable without additional restriction, as applying an orthonormal transformation M gives an equivalent representation: $\tilde{A} = MAM^{-1}$, $\tilde{C} = CM^{-1}$, $\tilde{Q} = MQM^T$. To enforce identifiability, we choose to restrict the first $\min(n, m)$ rows and columns of C to the identity matrix. In practice, we use the Cholesky decompositions ψ_Q, ψ_R of Q^{-1}, R^{-1} (respectively) instead of Q, R. The recursions for the forward backward algorithm for LGSSMs is known as the Kalman smoother [11, 8, 33]. Because the transition and emission processes are linear Gaussian, all forward messages, backward messages, and pairwise latent marginals $\gamma_{t-1:t}(x_{t-1}, x_t)$ are Gaussian; therefore, the gradients and preconditioning matrix can be calculated analytically. Further details are provided in the supplementary materials.

5.3.1. Error bound coefficients. The random maps of an LGSSM are strict contractions under mild conditions (Lemmas 5.1 and 5.2), and the gradients are Lipschitz in xx^T (Lemma 5.3). Therefore, Theorem 4.5 applies.

Lemma 5.1. *The forward random maps of an LGSSM are Gaussian linear maps. Specifically, $f_t(x_t) = F_t^f x_t + \zeta_t^f$, where ζ_t^f is a Gaussian random intercept and F_t^f is a matrix function of θ and $y_{>t}$. As a linear map, the Lipschitz constant of f_t is*

$$\|f_t\|_{Lip} = \|F_t^f\|_2 \leq \|A(I_n + QC^T R^{-1} C)^{-1}\|_2 = L_f \ . \tag{5.6}$$

As $\|(I_n + QC^T R^{-1} C)^{-1}\|_2 < 1$, if $\|A\|_2 < 1$, then $\|f_t\|_{Lip} \leq L_f < 1$ for all t.

Lemma 5.2. *The backward random maps of an LGSSM are Gaussian linear maps. Specifically, $b_t(x_t) = F_t^b x_t + \zeta_t^b$, where ζ_t^b is a Gaussian random intercept and F_t^b is a matrix function of θ and $y_{<t}$. As a linear map, the Lipschitz constant of b_t is*

$$\|b_t\|_{Lip} = \|F_t^b\|_2 \leq \|A(QA^T Q^{-1} A + QC^T R^{-1} C)^{-1}\|_2 = L_b \ . \tag{5.7}$$

If $\|A\|_2 < \|(QA^T Q^{-1} A + QC^T R^{-1} C)^{-1}\|_2$, then $\|f_t\|_{Lip} \leq L_f < 1$ for all t. In addition, when the variance of the prior $p_0(x)$ is less than the steady state variance $V_\infty = Q + AV_\infty A^T$ and A commutes with Q, we obtain a tighter bound:

$$\|b_t\|_{Lip} = \|F_t^b\|_2 \leq \|A(I_n + QC^T R^{-1} C)^{-1}\|_2 = L_b \ . \tag{5.8}$$

In this case, if $\|A\|_2 < 1$, then $\|b_t\|_{Lip} \leq L_b < 1$ for all t.

Lemmas 5.1 and 5.2 agree with intuition: when $\|A\|_2 \approx 0$ (no connection between x_{t-1} and x_t) or $\|Q\|_2 \gg \|R\|_2$ (transition noise is much larger than emission noise), then $L_f, L_b \approx 0$ (observations can be treated independently). Conversely, when $\|A\|_2 \approx 1$ and $\|Q\|_2 \ll \|R\|_2$, then $L_f, L_b \approx 1$ and buffering is necessary.

Lemma 5.3. *As x, y are jointly Gaussian in the LGSSM, the gradient of the complete-data loglikelihood is a quadratic form in xx^T with matrices*

$$\begin{aligned} \Omega = \{ & I_n \otimes Q^{-1}, I_n \otimes Q^{-1} A, Q^{-1/2} \otimes I_n, Q^{-1/2} A \otimes I_n, Q^{-1/2} \otimes A, Q^{-1/2} A \otimes A, \\ & I_n \otimes R^{-1}, I_n \otimes R^{-1} C, R^{-1/2} C \otimes C \} \ , \end{aligned} \tag{5.9}$$

where $Q^{-1/2} = \psi_Q$ and $R^{-1/2} = \psi_R$. Therefore, a bound for the Lipschitz constant is $L'_U = \max_{\omega \in \Omega} \|\omega\|_2$. This bound grows in $\|A\|, \|C\|, \|Q\|^{-1}, \|R\|^{-1}$.

The proofs can be found in the supplementary materials.

5.4. Switching linear dynamical system (SLDS). Switching linear dynamical systems (SLDSs) are an example of an SSM with both discrete and continuous latent variables. The form of SLDS models that we consider is

$$p(y, x, z \,|\, \theta) = \prod_{t=1}^{T} \Pi_{z_{t-1}, z_t} \cdot \mathcal{N}(x_t \,|\, A_{z_t} x_{t-1}, Q_{z_t}) \cdot \mathcal{N}(y_t \,|\, C x_t, R) \ , \tag{5.10}$$

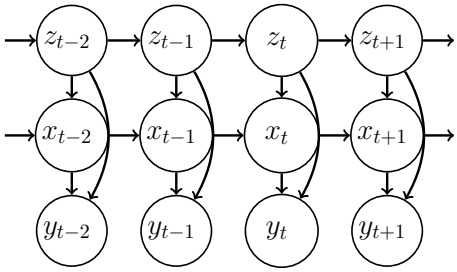

Figure 3. *Graphical model of an SLDS.*

where $y_t \in \mathbb{R}^m$ are the observations, $u_t \equiv (x_t, z_t) \in \mathbb{R}^n \times \{1, \dots, K\}$ are the mixed-type latent state sequence, and $\theta = \{\Pi, A, Q, C, R\}$ are the model parameters with $\Pi_k \in \Delta^K$, $A_k \in \mathbb{R}^{n \times n}$, $Q_k \in \mathbb{S}_+^n$ for $k = 1, \dots, K$, $C \in \mathbb{R}^{m \times n}$, and $R \in \mathbb{S}_+^m$ (see Figure 3). The SLDS of (5.10) can be viewed either as a latent AR(1)-HMM with conditional Gaussian emissions or as hidden Markov switches of an LGSSM. As an extension of the ARHMM, the latent continuous state sequence x_t can *smooth* noisy observations. As an extension of the LGSSM, the latent discrete state sequence z_t allows modeling of more complex dynamics by *switching* between different states (or regimes).

5.4.1. Gradient estimators. Unlike previous models, the forward-backward algorithm for the latent variables (x, z) in an SLDS does not have a closed form. Specifically, the transition kernel for x is a Gaussian mixture, and so the forward and backward messages of x are Gaussian mixtures with an exponentially increasing number of components (e.g., α_t has K^t components). Because the forward-backward algorithm is intractable for SLDSs, we rely on sampling (x, z) and forming a Monte Carlo estimate of the expectation in Fisher's identity (3.5). We consider various options of this Monte Carlo estimate below. To sample (x, z), we use a blocked Gibbs scheme as in [32], detailed in the supplementary materials.

Given a collection of N samples from blocked Gibbs $\{x^{(r)}, z^{(r)}\} \sim x, z \,|\, y, \theta$, we construct three different estimators for the marginal loglikelihood. The first estimator replaces the expectation in (3.5) with a Monte Carlo average:

$$\mathbb{E}_{x,z|y,\theta}[\nabla \log p(y, x, z \,|\, \theta)] \approx \frac{1}{N} \sum_{r=1}^{N} \nabla \log p(y, x^{(r)}, z^{(r)} \,|\, \theta) \ . \tag{5.11}$$

We construct two additional estimators by analytically integrating out either one of the two latent variables. These estimators are the *Rao–Blackwellization* of the naive Monte Carlo estimate [11]. Integrating out either x or z gives us

$$\mathbb{E}_{x,z|y,\theta}[\nabla \log p(y, x, z \,|\, \theta)] = \frac{1}{N} \sum_{r=1}^{N} \mathbb{E}_{x|y,z^{(r)},\theta}[\nabla \log p(y, x, z^{(r)} \,|\, \theta)] \ , \tag{5.12}$$

$$\mathbb{E}_{x,z|y,\theta}[\nabla \log p(y, x, z \,|\, \theta)] = \frac{1}{N} \sum_{r=1}^{N} \mathbb{E}_{z|y,x^{(r)},\theta}[\nabla \log p(y, x^{(r)}, z \,|\, \theta)] \ . \tag{5.13}$$

Because (5.12) integrates out x, it has lower variance for the gradient terms involving x (i.e., A, Q, R). Similarly, because (5.13) integrates out z, it has lower variance for the gradient terms involving z (i.e., Π).

Selecting one of the above Monte Carlo estimates of $\nabla U(\theta)$, we can deploy the same buffered subsampling estimator (3.5), obtaining Algorithm 5.1. Algorithm 5.1 replaces the forward-backward subroutine in Algorithm 3.2 with blocked Gibbs sampling over $\mathcal{S}^*$. Although this is more computationally costly than the exact forward-backward algorithms of the previous sections, it still provides memory saving and runtime speed ups compared to running a full blocked Gibbs sampler over $\mathcal{T}$. The explicit forms of (5.11)–(5.13), precondition matrix $D(\theta)$, and correction term $\Gamma(\theta)$ for SLDS used in Algorithm 3.1 are a combination of those for ARHMMs and LGSSMs. Complete details are provided in the supplementary materials.

Algorithm 5.1. `NoisyGradient` using blocked Gibbs (SLDS).

Input: data y, parameters θ, subsequence length S, error tolerance ϵ
$B = \texttt{BufferLength}(\theta, S, \epsilon)$ // From Theory or Adaptive
$\mathcal{S}, \mathcal{S}^* = \texttt{GetBufferedSubsequence}(y, S, B)$
$z_{\mathcal{S}^*}^{(0)} = \texttt{InitLatent}(\mathcal{S}^*, \theta)$ // With "burn-in"
for $r = 1, 2, \ldots, N$ **do**
  sample $x_{\mathcal{S}^*}^{(r)} \sim x_{\mathcal{S}^*} \,|\, y_{\mathcal{S}^*}, z_{\mathcal{S}^*}^{(r-1)}$ // Blocked Gibbs
  sample $z_{\mathcal{S}^*}^{(r)} \sim z_{\mathcal{S}^*} \,|\, y_{\mathcal{S}^*}, x_{\mathcal{S}^*}^{(r)}$
end for
calculate $\widetilde{U}(\theta)$ using a Monte Carlo estimate // (5.11), (5.12), or (5.13)
return $\nabla \widetilde{U}(\theta)$

5.4.2. Error bounds. There are two primary challenges for the error analysis of the SLDS: (i) the forward and backward smoothing kernels for the SLDS are mixtures, and (ii) the error from the finite-step blocked Gibbs sampler needs to be quantified. Conditions for contraction in the forward and backward smoothing random maps of switching models may follow from the conditions in [17]. Combining the convergence rate of the blocked Gibbs sampler with the error bound is an area we leave for future work. Our experiments in section 6.2 provide empirical evidence of the potential benefits of the algorithm.

6. Experiments.

We evaluate the performance of our proposed SGRLD algorithm (section 3) using both synthetic and real data. We organize our experiments by the corresponding models of section 5. Our evaluation focuses on the following three topics: (1) the computational speed up of SGMCMC over batch MCMC, (2) the effectiveness of buffering in correcting bias, and (3) the effectiveness of the complete-data Fisher information preconditioning of SGRLD over SGLD.

For batch MCMC, we consider block-Gibbs sampling (Gibbs) and unadjusted Langevin Monte Carlo—both with preconditioning (RLD) and without precondition (LD). Note that LD and RLD are SGLD and SGRLD with $S = T$.

To assess the performance of our samplers, we measure the marginal loglikelihood of

samples $\theta^{(s)}$ at different runtimes on a heldout test sequence. In synthetic data, where the true parameter θ^* is known, we also measure the mean-squared error (MSE) of the sample average $\hat{\theta}^{(s)} = \sum_{i \le s} \theta^{(i)}/s$ to θ^*. To assess the quality of our MCMC samples at approximating the posterior $\Pr(\theta \mid y)$, we measure the kernel Stein discrepancy (KSD) of each chain after burn-in given equal computation time [38, 49], rather than the effective sample size (ESS) [10, 34], as the KSD accounts for bias in the samples. As with all gradient-based methods, our SGMCMC methods require a hyperparameter search over the fixed stepsize tuning parameter h. We present results for the best stepsize as assessed via heldout loglikelihood on a validation set. As the potential $U(\theta)$ for SSMs is nonconvex, initialization is important. For the HMM and ARHMM, we initialize the parameters Π, A, Q using z given from K-means clustering of the observations y (or $y_{t-p:t}$). For the LGSSM, we initialize the parameters from the prior. For the mixed-type SLDS, we first sample R from the prior and initialize Π, A, Q using z from K-means. Finally, in our experiments, we use flat and noninformative priors for θ. For complete details, see the supplementary materials.

6.1. Gaussian HMM and ARHMM.

6.1.1. Synthetic ARHMM. We first consider synthetic data generated from a 2-state ARHMM in two dimensions: $m = 2$. The true model parameters θ^* are

$$\Pi = \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}, \quad Q_1 = Q_2 = 0.1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$A_1 = 0.9 \cdot \begin{bmatrix} \cos(-\vartheta) & -\sin(-\vartheta) \\ \sin(-\vartheta) & \cos(-\vartheta) \end{bmatrix}, \quad A_2 = 0.9 \cdot \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) \\ \sin(\vartheta) & \cos(\vartheta) \end{bmatrix}.$$

The model's two states are alternating rotations of $y \in \mathbb{R}^2$ with angle $\vartheta = \pi/4$, and the latent state sequence has a high transition rate $\Pr(z_t \neq z_{t-1}) = 0.9$. From this model, we generate time series of lengths $T = 10^4$ and 10^6.

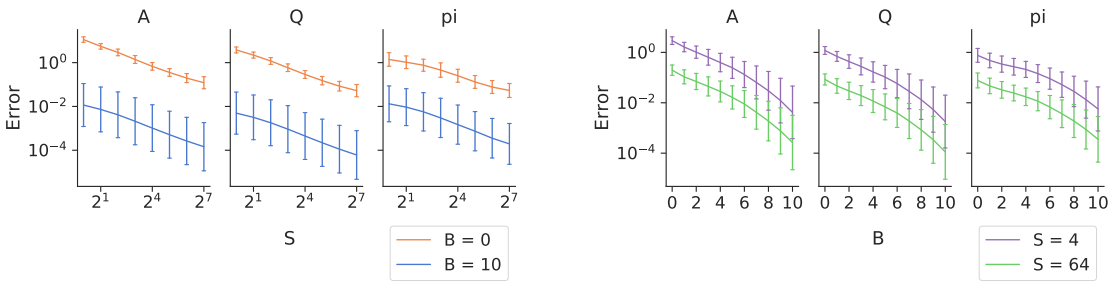

Figure 4. *Stochastic gradient error* $\mathbb{E}_{\mathcal{S}} \|\bar{g}(\theta) - \tilde{g}(\theta)\|_2$. *(Left) varying subsequence length* S *for no-buffer* $B = 0$ *and buffer* $B = 10$. *(Right) varying buffer size* B *for* $S = 4$ *and* $S = 64$ *subsequence lengths. Error bars are the standard deviation over* 100 *datasets.*

Figure 4 contains plots of the stochastic gradient error $\mathbb{E}_{\mathcal{S}} \|\bar{g}(\theta) - \tilde{g}(\theta)\|_2$ between the unbiased and buffered estimates evaluated at the true model parameters $\theta = \theta^*$. From Figure 4 (left), we see that the error decays $O(1/S)$ and that the errors in estimates without buffering $B = 0$ (orange) are orders of magnitude larger than the estimates with moderate buffering

$B = 10$ (blue). From Figure 4 (right), we see that the error decays geometrically in buffer size $O(L^B)$.

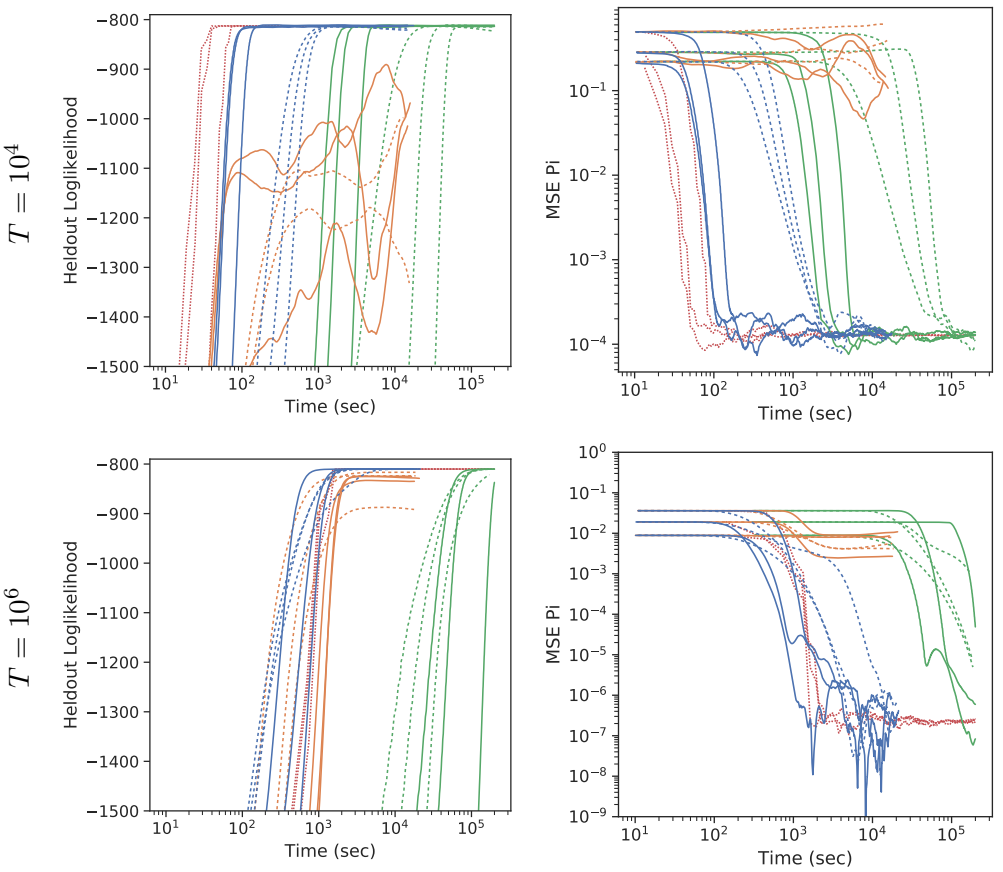

Figure 5. *Metrics vs. runtime on ARHMM data with $T = 10^4$ (top), $T = 10^6$ (bottom) for different methods: (Gibbs), (full), (no-buffer), and (buffer) SGMCMC. For SGMCMC methods, solid (—) and dashed (- -) lines indicate SGRLD and SGLD, respectively. The different metrics are (left) heldout loglikelihoood and (right) transition matrix estimation error $MSE(\hat{\Pi}^{(s)}, \Pi^*)$.*

In Figures 5 and 6, we compare subsequence-based MCMC methods, SGLD (no-buffer and buffer) and SGRLD (no-buffer and buffer), with full-sequence MCMC methods: LD, RLD, and Gibbs. We fit our samplers on one training sequence and evaluate performance on one test sequence. We consider two training sequences of lengths $T = 10^4$ and $T = 10^6$ and evaluate on the same test sequence of length $T = 10^4$. For the SGMCMC methods, we use a subsequence size of $S = 2$ and a buffer size of $B = 0$ (no-buffer) or $B = 2$ (buffer). We ran the subsequence methods for 6 hours and full-sequence methods for 144 hours.

From Figure 5, we see that our buffered SGMCMC (blue) helps convergence and mixing orders of magnitude faster than the full-sequence gradient MCMC (green). We also see that buffering is necessary to properly estimate Π, as the no-buffer SGMCMC methods (orange) do not properly learn Π. We also see that preconditioning helps convergence and mixing, as SGRLD (solid) outperforms SGLD (dashed). Although Gibbs outperforms SGMCMC for

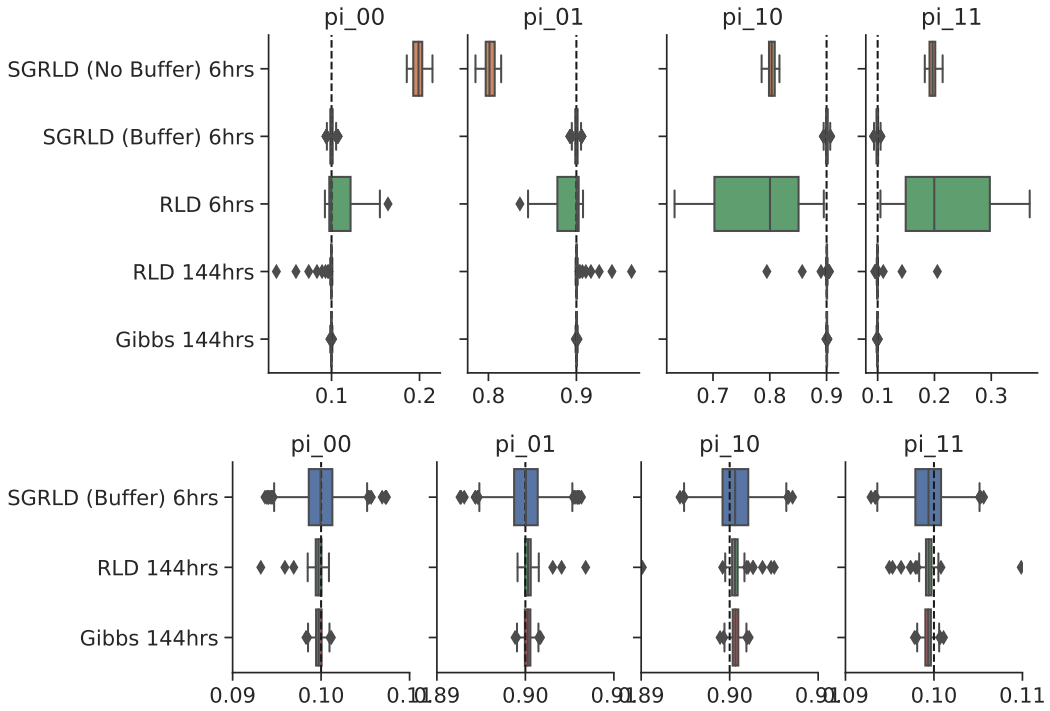

Figure 6. *Boxplot of MCMC samples for ARHMM data $T = 10^6$. (Top) comparison of all samplers, (bottom) zoom-in for top three. The half of each chain is discarded as burn-in. SGRLD with buffering in* 6 *hours is comparable to RLD or Gibbs in* 144 *hours.*

$T = 10^4$, Gibbs performs worse for $T = 10^6$, as each iteration requires a full pass over the dataset.

Figure 6 contains boxplots comparing the marginal distribution for the different methods on the synthetic ARHMM data $T = 10^6$. From Figure 6, we see that SGRLD with buffering in 6 hours is comparable to RLD or Gibbs in 144 hours; however, SGRLD without buffering is biased and RLD in 6 hours has not had enough time to mix.

Table 1 displays the KSD of the samples to the posterior after discarding half the samples as burn-in. The standard deviation is over MCMC chains with different initializations. Although RLD and Gibbs perform well for $T = 10^4$, both perform worse for larger $T = 10^6$ due to the increased time between samples. We also see that the nonbuffered methods do poorly for all T due to sampling from the incorrect distribution. Although SGLD (buffer) and SGRLD (buffer) perform comparably after burn-in, Figure 5 suggests SGRLD converges more rapidly.

In the supplementary materials, we present a synthetic data experiment for the Gaussian HMM and find similar results.

6.1.2. Ion channel recordings. We investigate the behavior of SGMCMC samplers on ion channel recording data when fitting a Gaussian HMM. In particular, we consider a 1MHz recording of a single alamethicin channel [63]. This data was previously investigated using a

Table 1

$\log_{10}$*(KSD) by variable of ARHMM samplers at* 6 *hours. Mean and (standard deviation) over runs in Figure* 5.

	Sampler	π	A	Σ
$T = 10^4$	SGLD (No-Buffer)	3.15 (0.46)	2.47 (0.51)	2.33 (0.30)
	SGLD (Buffer)	0.99 (0.13)	1.60 (0.20)	1.80 (0.13)
	LD	1.77 (0.72)	1.86 (0.32)	2.12 (0.36)
	SGRLD (No-Buffer)	3.15 (0.39)	2.02 (0.24)	1.91 (0.24)
	SGRLD (Buffer)	0.89 (0.04)	1.53 (0.10)	1.60 (0.30)
	RLD	0.67 (0.27)	2.02 (0.14)	1.60 (0.18)
	Gibbs	0.36 (0.07)	1.30 (0.20)	0.61 (0.13)
$T = 10^6$	SGLD (No-Buffer)	4.73 (0.07)	4.07 (0.22)	3.67 (0.25)
	SGLD (Buffer)	2.62 (0.06)	3.30 (0.20)	2.77 (0.31)
	LD	3.59 (0.22)	4.73 (0.33)	4.78 (0.34)
	SGRLD (No-Buffer)	4.75 (0.15)	4.02 (0.06)	3.61 (0.12)
	SGRLD (Buffer)	2.27 (0.08)	3.38 (0.08)	2.89 (0.09)
	RLD	3.31 (0.05)	4.22 (0.12)	3.56 (0.07)
	Gibbs	3.17 (0.30)	4.18 (0.07)	3.30 (0.07)

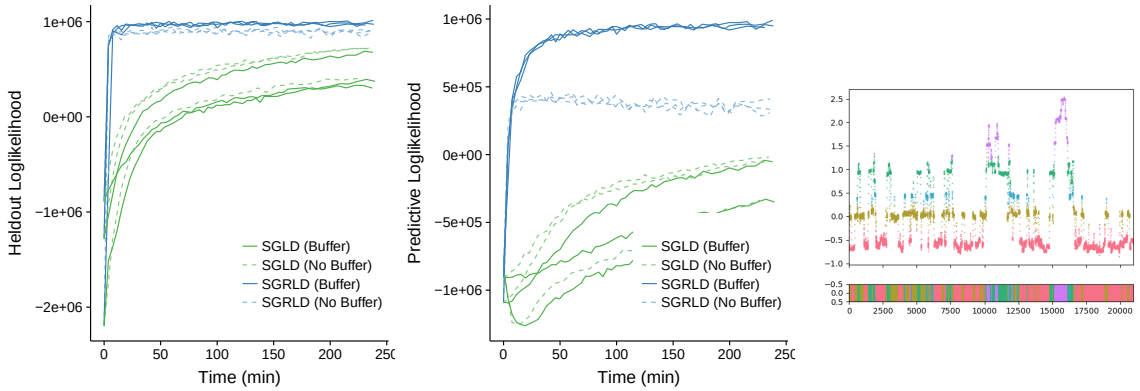

Figure 7. *Ion channel recordings: (Left) heldout loglikelihood vs. runtime. (Center)* 10*-step predictive loglikelihood* $\sum_t \log \Pr(y_{t+10} \mid \theta, y_{\leq t})$ *vs. runtime. (Right) segmentation by SGRLD (buffer).*

Bayesian nonparametric HMM in [57, 70]. In that work, the authors downsampled the data by a factor of 100 and only used 10,000 and 2,000 observations due to the challenge of scaling computations to the full sequence. We present the results on the data without downsampling (10 million observations), where Gibbs sampling runs into memory issues. Figure 7 presents our results, after applying a log-transform and normalizing the observations. We train on the first 90% and evaluate on the last 10%. For our SGMCMC methods, we use a subsequence size of $S = 10$ and a buffer size of $B = 0$ (no-buffer) or $B = 10$ (buffer). In addition to heldout loglikelihood, we also evaluate on 10-step ahead predictive loglikelihood $\sum_t \log \Pr(y_{t+10} \mid \theta, y_{\leq t})$, which is more sensitive to Π. We see that SGRLD quickly converges compared to SGLD. Although the buffered methods take longer to compute ($S + 2B = 30$ vs. $S = 10$), we see that buffering is necessary to perform well. In the supplementary materials, we present results comparing SGMCMC methods with Gibbs sampling on a downsampled version.

6.1.3. Canine seizure iEEG. We now consider applying SGMCMC samplers to intracranial EEG (iEEG) data. In particular, we consider data from a study on canines with epilepsy available at http://ieeg.org [19]. We focus on one canine, which over the course of 45.1 days was continuously monitored at 200Hz over 16 channels and recorded 90 seizures. This data was analyzed in prior work that compared a baseline ARHMM to nonparametric extensions using Gibbs sampling [76]. Following [76], we process the data into 4 minute windows around each seizure to focus on the seizure dynamics resulting in 90 time series of 48,000 points in $\mathbb{R}^{16}$. We use an ARHMM with $K = 5$ latent states and $p = 5$ lags treating each channel independently. We perform an 80-20 train-test split over 90 seizures, running inference on the training set and evaluating log-likelihood on the heldout test set. We compare SGLD and SGRLD samplers with $S = 100$ and $B = 10$ with the baseline Gibbs sampler on the full dataset. Because of the large data size, we also consider a *subset* Gibbs sampler that only uses 10% of the training set seizures.

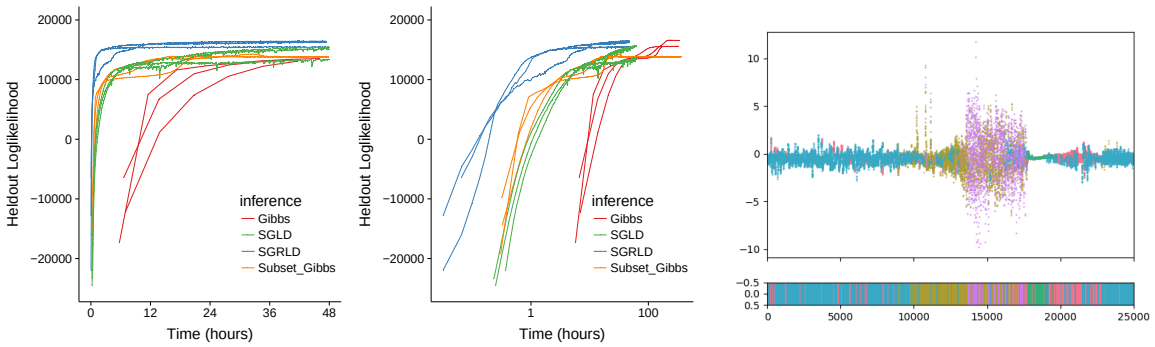

Figure 8. *ARHMM for canine seizure data: (left) heldout loglikelihood vs. time, (center) heldout loglikelihood vs. time on log-scale, (right) example segmentation of a test seizure channel by SLDS fit with SGRLD. The MCMC methods compared are Gibbs, subset Gibbs, SGLD, and SGLRD.*

In Figure 8, we see that SGRLD converges much more rapidly than the other methods. As each iteration of the Gibbs sampler takes 6 hours, it takes a couple of weeks for the Gibbs sampler to converge to the solution to which SGRLD converges in a few hours. Although the subset Gibbs sampler is 10 times faster than Gibbs, it does not converge to the full data posterior and its generalization error to the heldout test set is poorer than the other methods. From this experiment, we see that SGMCMC methods provide order of magnitude improvements (compared to subsetting the data).

6.2. LGSSM and SLDS. We first validate the LGSSM (SLDS with $K = 1$) on synthetic data. We then consider the SLDS sampler on a synthetic dataset and two real datasets: the seizure data of section 6.1.3 and a weather dataset.

6.2.1. Synthetic LGSSM. We consider synthetic data from an LGSSM with observations and latent state dimension $m = n = 2$. In particular, we consider a rotating state sequence with noisy observations. The true model parameters θ^* are

$$A = 0.7 \cdot \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) \\ \sin(\vartheta) & \cos(\vartheta) \end{bmatrix}, \quad Q = 0.1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where $\vartheta = \pi/4$. Because the transition error Q is smaller than the emission error R, inclusion of previous and future observations is necessary to accurately infer the continuous latent state x_t.

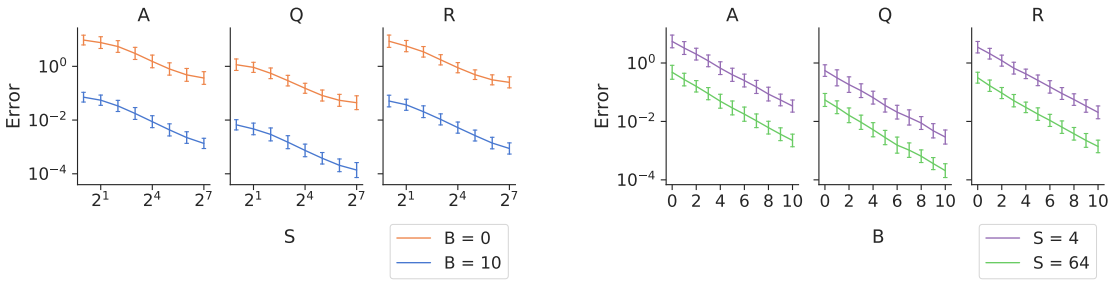

Figure 9. *Stochastic gradient error* $\mathbb{E}_{\mathcal{S}}\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2$. *(Left) varying subsequence length* S *for no-buffer* $B = 0$ *and buffer* $B = 10$. *(Right) varying buffer size* B *for* $S = 4$ *and* $S = 64$ *subsequence lengths. Error bars are the standard deviation over* 100 *datasets.*

Figure 9 contains plots of the stochastic gradient error $\mathbb{E}_{\mathcal{S}}\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2$ between the unbiased and buffered estimates evaluated at the true model parameters $\theta = \theta^*$. Similar to the ARPHMM, we see that the error decays $O(1/S)$ and that moderate buffering (e.g., $B = 10$) deceases the error by orders of magnitude in Figure 9 (left). And we see that the error decays geometrically in buffer size $O(L^B)$ in Figure 9 (right).

In Figures 10 and 11, we compare SGLD (no-buffer and buffer), SGRLD (no-buffer and buffer), LD, RLD, and a blocked Gibb sampler. We fit our samplers on one training sequence and evaluate performance on one test sequence. We consider two training sequences of lengths $T = 10^4$ and $T = 10^6$ and evaluate on the same test sequence of length $T = 10^4$. For the SGMCMC methods, we use a subsequence size of $S = 20$ with $B = 0$ (no buffer) and $B = 10$ (buffer). We see that even with a large subsequence size, buffering is crucial for accurate inference, as SGMCMC methods without buffering converge to a different stationary distribution than the posterior.

In Table 2, we evaluate the KSD of the different MCMC methods. We see that SGMCMC with buffering slightly outperforms the full sequence methods for $T = 10^4$ and significantly outperforms the full sequence methods for $T = 10^6$, while SGMCMC without buffering performs poorly due to bias.

6.2.2. Synthetic SLDS. We now consider synthetic data from a model we can view as a switching extension of the LGSSM in section 6.2.1 or as a noisy version of the ARHMM in the supplementary materials. The true model parameters θ^* are

$$\Pi = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}, \quad Q_1 = Q_2 = 0.1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 0.1 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$A_1 = 0.9 \cdot \begin{bmatrix} \cos(-\vartheta) & -\sin(-\vartheta) \\ \sin(-\vartheta) & \cos(-\vartheta) \end{bmatrix}, \quad A_2 = 0.9 \cdot \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) \\ \sin(\vartheta) & \cos(\vartheta) \end{bmatrix},$$

where again $\vartheta = \pi/4$. We generate sequences of lengths $T = 10^4$ and 10^6.

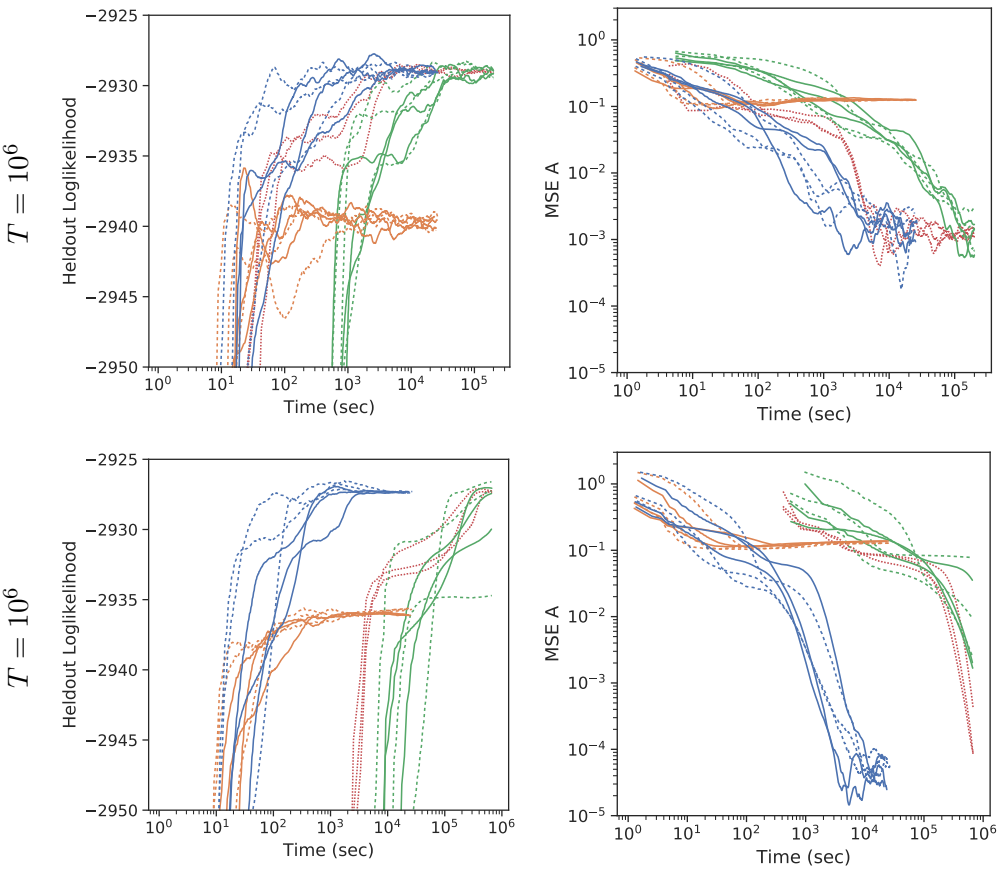

Figure 10. *Metrics vs. runtime on LGSSM with $T = 10^4$ (top), $T = 10^6$ (bottom) for different methods: (Gibbs), (full), (no-buffer), and (buffer) SGMCMC. For SGMCMC methods, solid (—) and dashed (- -) lines indicate SGRLD and SGLD, respectively. The different metrics are (left) heldout loglikelihoood and (right) transition matrix estimation error $MSE(\hat{A}^{(s)}, A^*)$.*

We first compare the variance of the three difference Monte Carlo gradient estimators for SLDS: using (x, z) samples (`xz Gradient`) as in (5.11), only using z samples (`z Gradient`) as in (5.12), and only using x samples (`x Gradient`) as in (5.13). Figure 12 presents boxplots of $\tilde{g}(\theta) - g(\theta)$ for the three different estimators at $\theta = \theta^*$. From Figure 12 (left), we see that `z Gradient` (blue) has much lower variance than the other two estimators for the gradient of A. This also holds for the gradients of Q and R (see the supplementary materials). From Figure 12 (right), we see that all three estimators have similar variance for the gradient of Π (with `x Gradient` (green) slightly better than the other two). This agrees with the intuition described in section 5.4.1. Because `z Gradient` has lower variance than the other two estimators, we can use larger stepsizes, leading to faster convergence and mixing.

Figure 13 contains plots of the stochastic gradient error $\mathbb{E}_{\mathcal{S}}\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2$ between the unbiased and buffered estimates (for `z Gradient`) evaluated at the true model parameters $\theta = \theta^*$. For short buffered subsequences (e.g., small S and B), the error decays as expected: $O(L^B/S)$; however, for longer buffered subsequences the error is dominated by the Monte

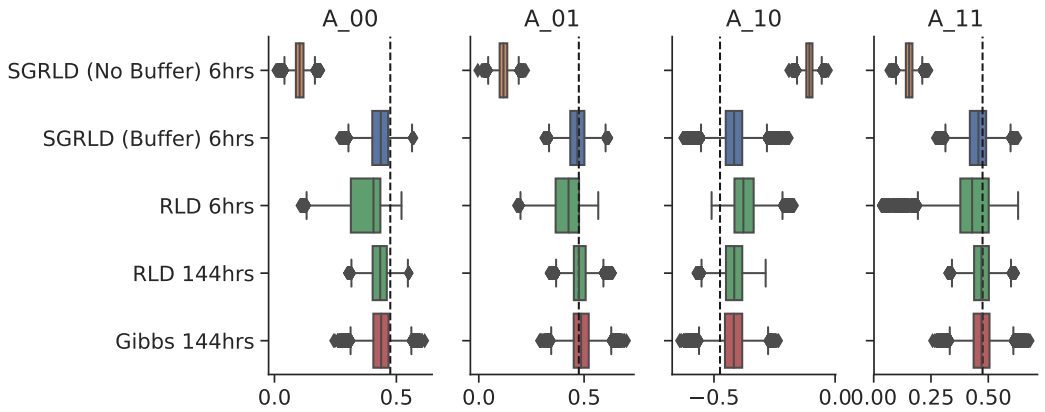

Figure 11. *Boxplot of MCMC samples of transition matrix A for LGSSM data $T = 10^4$. SGRLD with buffering in 6 hours is comparable to RLD or Gibbs in 144 hours. SGRLD without buffering is biased, and RLD in 6 hours has not fully mixed.*

Table 2

$\log_{10}$*(KSD) by variable of LGSSM samplers at 6 hours. Mean and (the standard deviation) over runs in Figure 10.*

	Sampler	A	Q	R
$T = 10^4$	SGLD (No-Buffer)	2.39 (0.01)	1.73 (0.03)	1.48 (0.03)
	SGLD (Buffer)	0.88 (0.11)	0.41 (0.11)	0.86 (0.08)
	LD	0.99 (0.13)	1.12 (0.19)	1.10 (0.17)
	SGRLD (No-Buffer)	2.38 (0.01)	1.70 (0.02)	1.43 (0.02)
	SGRLD (Buffer)	0.85 (0.08)	0.18 (0.12)	0.77 (0.14)
	RLD	0.99 (0.12)	0.90 (0.19)	1.10 (0.17)
	Gibbs	0.74 (0.20)	0.33 (0.18)	1.06 (0.27)
$T = 10^6$	SGLD (No-Buffer)	4.32 (0.01)	3.79 (0.02)	3.50 (0.02)
	SGLD (Buffer)	2.30 (0.19)	1.61 (0.18)	2.84 (0.03)
	LD	4.26 (0.35)	4.00 (0.39)	4.14 (0.19)
	SGRLD (No-Buffer)	4.27 (0.01)	3.77 (0.02)	3.23 (0.03)
	SGRLD (Buffer)	2.17 (0.33)	1.64 (0.21)	3.03 (0.12)
	RLD	4.34 (0.23)	3.76 (0.25)	4.03 (0.23)
	Gibbs	3.46 (0.28)	3.52 (0.14)	3.50 (0.28)

Carlo error in the number of Gibbs steps used in sampling z for calculating $\tilde{g}$ in (5.12).

In Figure 14, we compare SGRLD (with buffer), using each of the gradient estimators (5.11)–(5.13), and a blocked Gibb sampler. We run our samplers on one training sequence and evaluate performance on another test sequence. For all SGRLD samplers, we used subsequence sizes of $S = 10$ and $B = 10$. As the marginal loglikelihood is not available in closed form for SLDSs, we instead use a Monte Carlo approximation of the EM lower bound $\log \Pr(y \mid \theta) \geq \mathbb{E}_{x,z|y,\theta}[\log \Pr(y, x, z \mid \theta)]$ where the expectation is approximated with samples of x, z drawn using blocked Gibbs for each fixed θ. From Figure 14, we see that SGRLD methods perform similarly to Gibbs for $T = 10^4$ but vastly outperform Gibbs for $T = 10^6$.

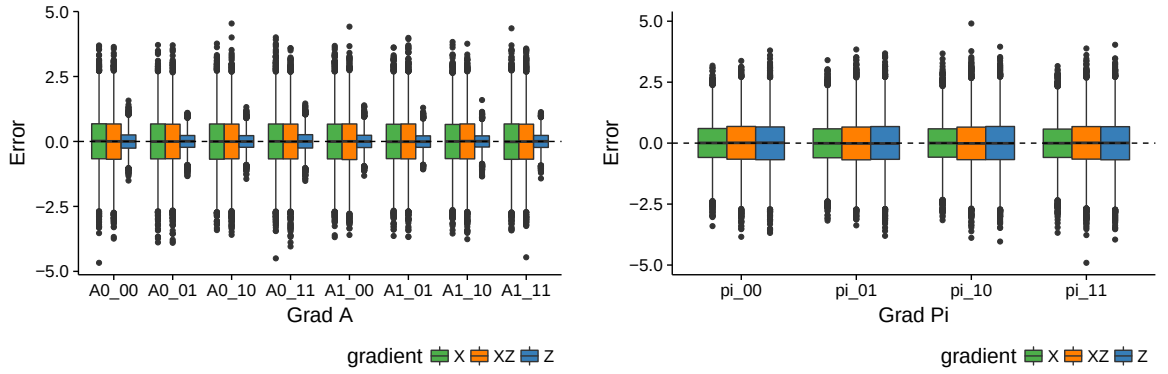

Figure 12. *SLDS gradient error for the different estimators* (5.11)–(5.13). *(Left) Boxplots of* $\tilde{g}(\theta)_A - g(\theta)_A$. *(Right) Boxplots of* $\tilde{g}(\theta)_\Pi - g(\theta)_\Pi$.

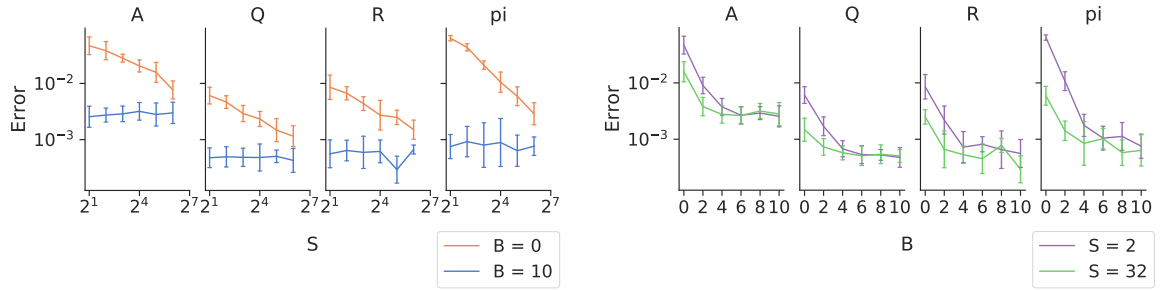

Figure 13. *Stochastic gradient error* $\mathbb{E}_S\|\bar{g}(\theta) - \tilde{g}(\theta)\|_2$ *for* `z Gradient`. *(Left) error varying subsequence length* S *for no-buffer* $B = 0$ *and buffer* $B = 4$. *(Right) error varying buffer size* B *for small* $S = 2$ *and long* $S = 32$ *subsequences. Error bars are the standard deviation over* 100 *datasets.*

6.2.3. Canine seizure iEEG. Recall the data from section 6.1.3. For our SLDS analysis, we set the continuous latent variable dimension to $n = 1$. The number of latent states remains $K = 5$. We again compare SGLD and SGRLD samplers with $S = 100$ and $B = 10$ to Gibbs samplers on both the full data set and a 10% subset of seizures. In Figure 15, we see again that the SGRLD sampler converges much more rapidly than the other methods. In comparison to Figure 8, we also see that the SLDS is a better model for this data than the ARHMM (as measured by heldout likelihood). Qualitatively, the SLDS segmentation of seizures (Figure 15 (right)) is more contiguous than the ARHMM segmentation (Figure 8 (right)).

6.2.4. Historical cities' weather data. We apply SGMCMC to historical city weather data from Kaggle [7]. The data consists of hourly temperature, pressure, and humidity measurements ($m = 3$) for 20 U.S. cities over 5 years with $T = 44,000$ hourly observations per city. We fit SLDS models with $n = 3$ and $K = 4$ to both the hourly and daily average observations, treating the cities independently. For both sets of observations, we perform an 80-20 train-test split over 20 cities, running inference on the training set (16 cities) and evaluating loglikelihood on the test set (four cities).

Figure 16 (top-left) shows the heldout loglikelihood vs. the runtime for the different sam-

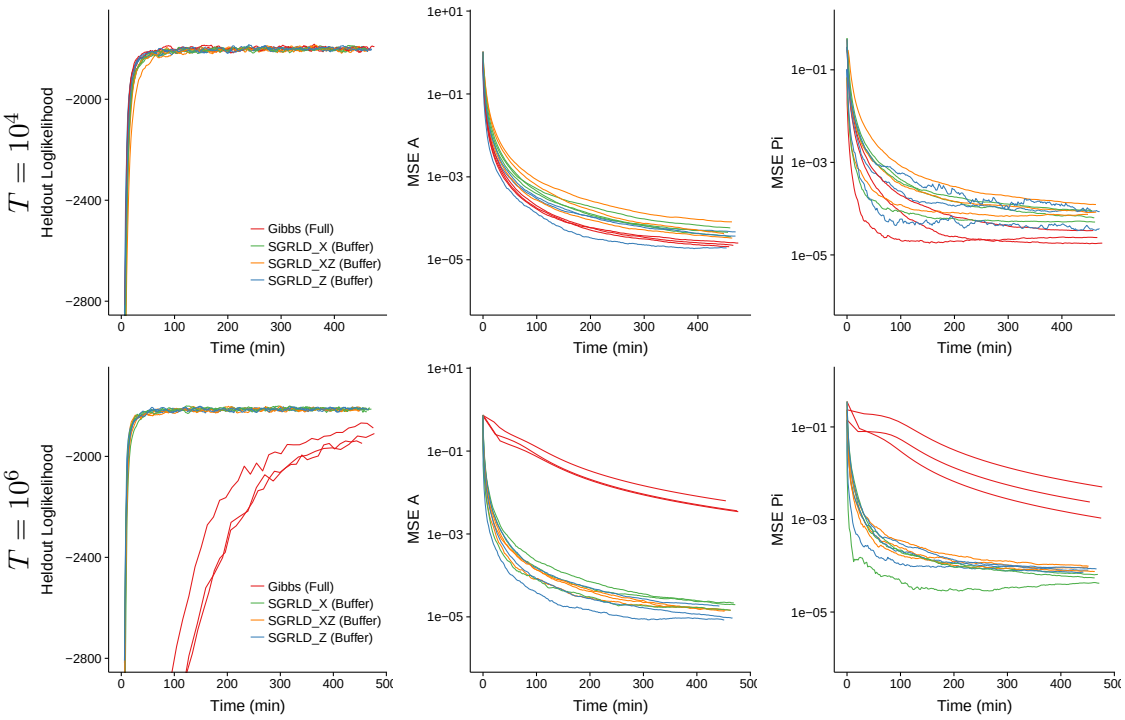

Figure 14. *Metrics vs. runtime on SLDS data for different inference methods: Gibbs, SGRLD X, SGRLD XZ, and SGLRD Z. (Top) $T = 10^4$. (Bottom) $T = 10^6$. The metrics are (left) heldout loglikelihood, (center) estimation error $MSE(\hat{A}^{(s)}, A^*)$, (right) estimation error $MSE(\hat{\Pi}^{(s)}, \Pi^*)$.*

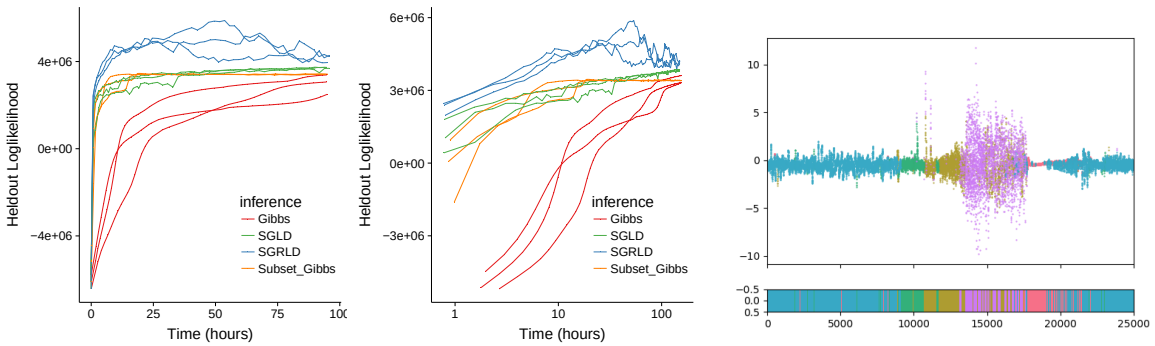

Figure 15. *SLDS canine seizure data: (left) heldout loglikelihood vs. time, (center) heldout loglikelihood vs. time on log-scale, (right) example segmentation by ARHMM fit with SGRLD. The MCMC methods compared are Gibbs, subset Gibbs, SGLD, and SGLRD.*

plers on the daily data. From this plot, we see that SGRLD clearly outperforms Gibbs. Although Gibbs converges quickly on the daily data, it gets stuck in local optima. In particular, the Gibbs runs converge to a suboptimal parametrization that mixes over three states, while SGRLD converges to a two-state (summer-winter) solution (with the remaining states for sudden shifts or jumps). For example, Figure 16 (top-center and right) contains fits of the daily model to the Houston time series for both Gibbs and SGRLD, respectively. Figure 16

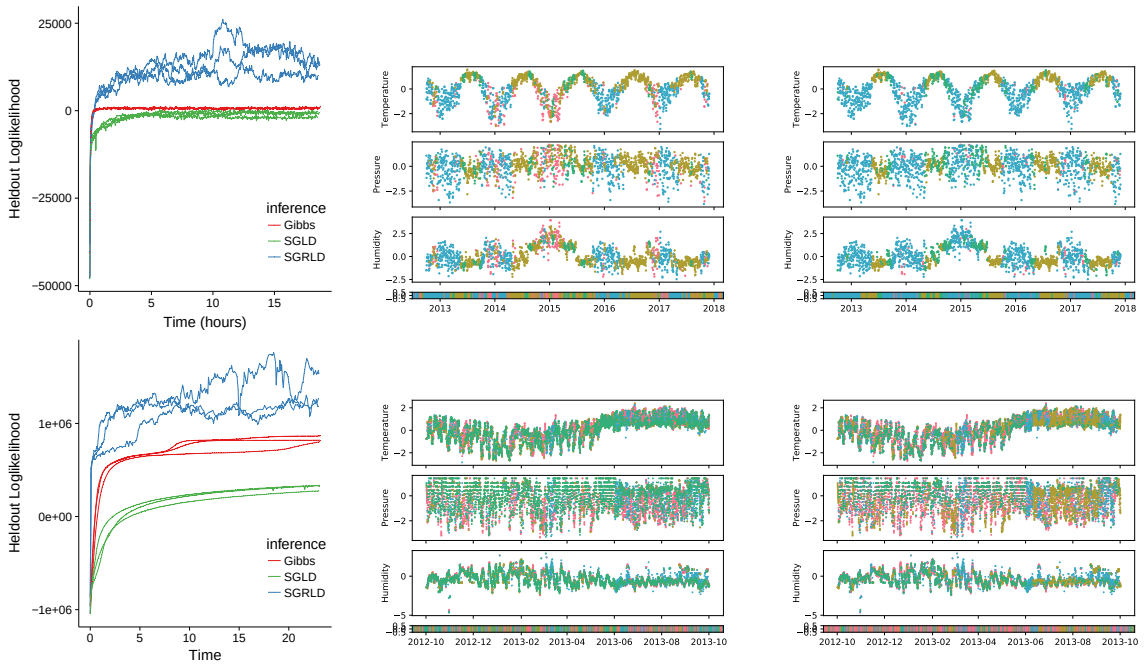

Figure 16. *SLDS weather data. (Top) daily aggregated data, (bottom) hourly data. (Left) heldout loglikelihood vs. runtime, (center) Gibbs Houston fit, (right) SGRLD Houston fit.*

(bottom-left) shows the heldout loglikelihood vs. the runtime of the different samplers for the hourly data. SGRLD again outperforms Gibbs and, for the hourly data, the Gibbs sampler is significantly slower than the SGMCMC samplers.

7. Conclusions.

In this work, we developed stochastic gradient MCMC samplers for SSMs of sequential data. Our key contribution is a *buffered* gradient estimator $\tilde{g}(\theta)$ for general discrete-time SSMs based on Fisher's identity. We developed bounds for the error of this buffered gradient estimator and showed that the error decays geometrically in the buffer size under mild conditions. Using this estimator and bound, we developed SGRLD samplers for discrete (Gaussian HMM, ARHMM), continuous (LGSSM), and mixed-type (SLDS) SSMs. In our experiments, we find that our methods can provide orders of magnitude runtime speedups compared to Gibbs sampling, control bias with modest buffer size, and converge and mix more rapidly using preconditioning. In particular, our SGRLD method only uses subsequences at each iteration and is able to take advantage of geometric structure using the complete-data Fisher information matrix.

There are many interesting directions for future work. This buffered gradient estimator for sequential data could be applied to other stochastic gradient methods, such as maximum likelihood estimation or variational inference [3, 45]. The approach could also be extended to nonlinear continuous SSMs (e.g., stochastic volatility models), replacing message passing with particle filtering [2, 11, 25, 56]. The buffered gradient estimator could likewise be applied to diffusions with control variates [5, 14] or with augmented dynamics, such as using momentum (SGHMC) [16] or temperature (SGNHT) [23]. In terms of analysis, the standard SGLD

error analysis could be extended to analyze the optimal trade-off between buffer size and subsequence length.

Acknowledgments. We would like to thank Drausin Wulsin, Jack Baker, Chris Nemeth, and other members of the Dynamode lab at UW for their helpful discussions.

REFERENCES

[1] S. Ahn, A. Korattikara, and M. Welling, *Bayesian posterior sampling via stochastic gradient Fisher scoring*, in Proceedings of the International Conference on Machine Learning, 2012, pp. 1771–1778.
[2] C. Andrieu, A. Doucet, and R. Holenstein, *Particle Markov chain Monte Carlo methods*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 72 (2010), pp. 269–342.
[3] E. Archer, I. M. Park, L. Buesing, J. Cunningham, and L. Paninski, *Black Box Variational Inference for State Space Models*, preprint, https://arxiv.org/abs/1511.07367, 2015.
[4] J. Baker, P. Fearnhead, E. Fox, and C. Nemeth, *Large-scale stochastic sampling from the probability simplex*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2018, pp. 6722–6732.
[5] J. Baker, P. Fearnhead, E. B. Fox, and C. Nemeth, *Control variates for stochastic gradient MCMC*, Stat. Comput., 29 (2019), pp. 599–615.
[6] M. J. Beal, *Variational Algorithms for Approximate Bayesian Inference*, Ph.D thesis, University of London, London, 2003.
[7] D. Beniaguev, *Historical Hourly Weather Data* 2012–2017, https://www.kaggle.com/selfishgene/historical-hourly-weather-data.
[8] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
[9] M. Briers, A. Doucet, and S. Maskell, *Smoothing algorithms for state-space models*, Ann. Inst. Statist. Math., 62 (2010), pp. 61–89.
[10] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*, CRC Press, Boca Raton, FL, 2011.
[11] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, Springer, New York, 2005.
[12] C. K. Carter and R. Kohn, *On Gibbs sampling for state space models*, Biometrika, 81 (1994), pp. 541–553.
[13] H. P. Chan, C.-W. Heng, and A. Jasra, *Theory of segmented particle filters*, Adv. in Appl. Probab., 48 (2016), pp. 69–87.
[14] N. S. Chatterji, N. Flammarion, Y.-A. Ma, P. L. Bartlett, and M. I. Jordan, *On the theory of variance reduction for stochastic gradient Monte Carlo*, in Proceedings of the International Conference on Machine Learning, 2018, pp. 764–773.
[15] C. Chen, N. Ding, and L. Carin, *On the convergence of stochastic gradient MCMC algorithms with high-order integrators*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2015, pp. 2278–2286.
[16] T. Chen, E. Fox, and C. Guestrin, *Stochastic gradient Hamiltonian Monte Carlo*, in Proceedings of the International Conference on Machine Learning, 2014, pp. 1683–1691.
[17] B. Cloez and M. Hairer, *Exponential ergodicity for Markov processes with random switching*, Bernoulli, 21 (2015), pp. 505–536.
[18] A. S. Dalalyan and A. G. Karagulyan, *User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient*, Stochastic Process. Appl., to appear.
[19] K. A. Davis, H. Ung, D. Wulsin, J. Wagenaar, E. Fox, N. Patterson, C. Vite, G. Worrell, and B. Litt, *Mining continuous intracranial EEG in focal canine epilepsy: Relating interictal bursts to seizure onsets*, Epilepsia, 57 (2016), pp. 89–98.
[20] P. Del Moral, A. Doucet, and S. Singh, *Forward Smoothing Using Sequential Monte Carlo*, preprint, https://arxiv.org/abs/1012.5390, 2010.
[21] P. Del Moral, A. Jasra, and Y. Zhou, *Biased online parameter inference for state-space models*, Methodol. Comput. Appl. Probab., 19 (2017), pp. 727–749.

[22] P. Diaconis and D. Freedman, *Iterated random functions*, SIAM Rev., 41 (1999), pp. 45–76, https://doi.org/10.1137/S0036144598338446.
[23] N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven, *Bayesian sampling using stochastic gradient thermostats*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2014, pp. 3203–3211.
[24] R. Douc, E. Moulines, and Y. Ritov, *Forgetting of the initial condition for the filter in general state-space hidden Markov chain: A coupling approach*, Electron. J. Probab., 14 (2009), pp. 27–49.
[25] A. Doucet and A. M. Johansen, *A tutorial on particle filtering and smoothing: Fifteen years later*, in The Oxford Handbook of Nonlinear Filtering, Oxford University Press, Oxford, UK, 2011.
[26] K. A. Dubey, S. J. Reddi, S. A. Williamson, B. Poczos, A. J. Smola, and E. P. Xing, *Variance reduction in stochastic gradient Langevin dynamics*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2016, pp. 1154–1162.
[27] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*, 2nd ed., Oxford Statist. Sci. Ser. 38, Oxford University Press, Oxford, UK, 2012.
[28] A. Durmus and É. Moulines, *Quantitative bounds of convergence for geometrically ergodic Markov chain in the Wasserstein distance with application to the Metropolis adjusted Langevin algorithm*, Stat. Comput., 25 (2015), pp. 5–19.
[29] S. R. Eddy, *Profile hidden Markov models*, Bioinformatics, 14 (1998), pp. 755–763.
[30] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*, Appl. Math. (N. Y.) 29, Springer, New York, 2008.
[31] N. Foti, J. Xu, D. Laird, and E. Fox, *Stochastic variational inference for hidden Markov models*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2014, pp. 3599–3607.
[32] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, *Bayesian nonparametric inference of switching dynamic linear models*, IEEE Trans. Signal Process., 59 (2011), pp. 1569–1595.
[33] E. B. Fox, *Bayesian Nonparametric Learning of Complex Dynamical Phenomena*, Ph.D. thesis, MIT, Cambridge, MA, 2009.
[34] A. Gelman, J. B. Carlin, D. B. Rubin, A. Vehtari, D. B. Dunson, and H. S. Stern, *Bayesian Data Analysis*, CRC Press, Boca Raton, FL, 2014.
[35] M. Girolami and B. Calderhead, *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 73 (2011), pp. 123–214.
[36] J. Gonzalez, Y. Low, and C. Guestrin, *Residual splash for optimally parallelizing belief propagation*, in Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, 2009, pp. 177–184.
[37] C. A. Goodhart and M. O'Hara, *High frequency data in financial markets: Issues and applications*, J. Empir. Finance, 4 (1997), pp. 73–114.
[38] J. Gorham and L. Mackey, *Measuring sample quality with kernels*, in Proceedings of the International Conference on Machine Learning, 2017, pp. 1292–1301.
[39] J. D. Hamilton, *Time Series Analysis*, Vol. 2, Princeton University Press, Princeton, NJ, 1994.
[40] J. E. Johndrow and J. C. Mattingly, *Error Bounds for Approximations of Markov Chains Used in Bayesian Sampling*, preprint, https://arxiv.org/abs/1711.05382, 2017.
[41] J. E. Johndrow, J. C. Mattingly, S. Mukherjee, and D. B. Dunson, *Optimal Approximating Markov Chains for Bayesian Inference*, preprint, https://arxiv.org/abs/1508.03387, 2017.
[42] M. Johnson and A. Willsky, *Stochastic variational inference for Bayesian time series models*, in Proceedings of the International Conference on Machine Learning, 2014, pp. 1854–1862.
[43] M. J. Johnson and A. S. Willsky, *Bayesian nonparametric hidden semi-Markov models*, J. Mach. Learn. Res., 14 (2013), pp. 673–701.
[44] C.-J. Kim and C. R. Nelson, *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*, Vol. 1, MIT Press, Cambridge, MA, 1999.
[45] R. G. Krishnan, U. Shalit, and D. Sontag, *Structured inference networks for nonlinear state space models*, in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 2101–2109.
[46] F. Le Gland and L. Mevel, *Exponential forgetting and geometric ergodicity in hidden Markov models*, Math. Control Signals Systems, 13 (2000), pp. 63–93.

[47] C. Li, C. Chen, D. E. Carlson, and L. Carin, *Preconditioned stochastic gradient Langevin dynamics for deep neural networks*, in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1788–1794.
[48] S. Linderman, M. Johnson, A. Miller, R. Adams, D. Blei, and L. Paninski, *Bayesian learning and inference in recurrent switching linear dynamical systems*, in Proceedings of the Twentieth International Conference on Artificial Intelligence and Statistics, 2017, pp. 914–922.
[49] Q. Liu, J. Lee, and M. Jordan, *A kernelized Stein discrepancy for goodness-of-fit tests*, in Proceedings of the International Conference on Machine Learning, 2016, pp. 276–284.
[50] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*, Springer-Verlag, Berlin, 2005.
[51] Y.-A. Ma, T. Chen, and E. B. Fox, *A complete recipe for stochastic gradient MCMC*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2015, pp. 2917–2925.
[52] Y.-A. Ma, N. J. Foti, and E. B. Fox, *Stochastic gradient MCMC methods for hidden Markov models*, in Proceedings of the International Conference on Machine Learning, 2017, pp. 2265–2274.
[53] N. Madras and D. Sezer, *Quantitative bounds for Markov chain convergence: Wasserstein and total variation distances*, Bernoulli, 16 (2010), pp. 882–908.
[54] Mayo Clinic and University of Pennsylvania, http://ieeg.org.
[55] T. Nagapetyan, A. B. Duncan, L. Hasenclever, S. J. Vollmer, L. Szpruch, and K. Zygalakis, *The True Cost of Stochastic Gradient Langevin Dynamics*, preprint, https://arxiv.org/abs/1706.02692, 2017.
[56] J. Olsson, O. Cappé, R. Douc, and E. Moulines, *Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models*, Bernoulli, 14 (2008), pp. 155–179.
[57] K. Palla, D. A. Knowles, and Z. Ghahramani, *A reversible infinite HMM using normalised random measures*, in Proceedings of the International Conference on Machine Learning, 2014, pp. 1998–2006.
[58] S. Patterson and Y. W. Teh, *Stochastic gradient Riemannian Langevin dynamics on the probability simplex*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2013, pp. 3102–3110.
[59] L. R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, Proc. IEEE, 77 (1989), pp. 257–286.
[60] M. Raginsky, A. Rakhlin, and M. Telgarsky, *Non-convex learning via stochastic gradient Langevin dynamics: A nonasymptotic analysis*, in Proceedings of the Conference on Learning Theory, 2017, pp. 1674–1703.
[61] G. O. Roberts and J. S. Rosenthal, *Optimal scaling of discrete approximations to Langevin diffusions*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 60 (1998), pp. 255–268.
[62] G. O. Roberts and R. L. Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli, 2 (1996), pp. 341–363.
[63] J. K. Rosenstein, S. Ramakrishnan, J. Roseman, and S. K. L, *Single ion channel recordings with CMOS-anchored lipid membranes*, Nano Lett., 13 (2013), pp. 2682–2686.
[64] D. Rudolf and N. Schweizer, *Perturbation theory for Markov chains via Wasserstein distance*, Bernoulli, 24 (2018), pp. 2610–2639.
[65] S. L. Scott, *Bayesian methods for hidden Markov models: Recursive computing in the* 21*st century*, J. Amer. Statist. Assoc., 97 (2002), pp. 337–351.
[66] U. Simsekli, R. Badeau, T. Cemgil, and G. Richard, *Stochastic quasi-Newton Langevin Monte Carlo*, in Proceedings of the International Conference on Machine Learning, 2016, pp. 642–651.
[67] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, *Nonparametric belief propagation*, Commun. ACM, 53 (2010), pp. 95–103.
[68] Y. W. Teh, A. H. Thiery, and S. J. Vollmer, *Consistency and fluctuations for stochastic gradient Langevin dynamics*, J. Mach. Learn. Res., 17 (2016), pp. 193–225.
[69] X. T. Tong and R. Van Handel, *Ergodicity and stability of the conditional distributions of nondegenerate Markov chains*, Ann. Appl. Probab., 22 (2012), pp. 1495–1540.
[70] N. Tripuraneni, S. Gu, H. Ge, and Z. Ghahramani, *Particle Gibbs for infinite hidden Markov models*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2015, pp. 2386–2394.
[71] R. Van Hande, *The stability of conditional Markov processes and Markov chains in random environments*, Ann. Probab., 37 (2009), pp. 1876–1925.

[72] C. Villani, *Optimal Transport: Old and New*, Grundlehren Math. Wiss. 338, Springer-Verlag, Berlin, 2009.
[73] N. X. Vinh, J. Epps, and J. Bailey, *Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance*, J. Mach. Learn. Res., 11 (2010), pp. 2837–2854.
[74] M. Welling and Y. W. Teh, *Bayesian learning via stochastic gradient Langevin dynamics*, in Proceedings of the International Conference on Machine Learning, 2011, pp. 681–688.
[75] N. Whiteley, *Stability properties of some particle filters*, Ann. Appl. Probab., 23 (2013), pp. 2500–2537.
[76] D. F. Wulsin, *Bayesian Nonparametric Modeling of Epileptic Events*, Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, 2013.
[77] T. Xifara, C. Sherlock, S. Livingstone, S. Byrne, and M. Girolami, *Langevin diffusions and the Metropolis-adjusted Langevin algorithm*, Statist. Probab. Lett., 91 (2014), pp. 14–19.
[78] P. Xu, J. Chen, and Q. Gu, *Global convergence of Langevin dynamics based algorithms for nonconvex optimization*, in Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2018, pp. 3122–3133.
[79] F. X.-F. Ye, Y.-a. Ma, and H. Qian, *Estimate Exponential Memory Decay in Hidden Markov Model and Its Applications*, preprint, https://arxiv.org/abs/1710.06078, 2017.
[80] S.-Z. Yu, *Hidden semi-Markov models*, Artif. Intell., 174 (2010), pp. 215–243.
[81] Y. Zeng and S. Wu, *State-Space Models: Applications in Economics and Finance*, Vol. 1, Springer, New York, 2013.