A Langevinized Ensemble Kalman Filter for Large-Scale Dynamic Learning

Purdue University

Supplementary Material

This material is organized as follows. Section S1 presents more numerical examples including the LSTM model, the comparisons of LEnKF with sequential importance sampling, and the comparisons of LEnKF with some stochastic gradient MCMC algorithms such as SGLD, pSGLD and SGNHT. Section S2 presents the convergence theory of LEnKF, where the proofs for Theorem 1 and Theorem 2 are provided.

S1 More Numerical Examples

S1.1 Bayesian Variable Selection for Large-Scale Linear Regression

Comparison of LEnKF with SGLD, pSGLD and SGNHT as reported in Section 4.1

For comparison, SGLD (Welling and Teh, 2011), preconditioned SGLD (pSGLD, Li et al., 2016), and stochastic gradient Nosé-Hoover thermostat (SGNHT, Ding et al., 2014) were applied to this example. For these algorithms, the learning rates have been tuned to their maximum values such that the simulation converges fast while not exploding. For SGLD, we set $\epsilon_t = 4 \times 10^{-6}/\max\{t_0, t\}^{0.6}$ with $t_0 = 1000$; for pSGLD, we set $\epsilon_t = 5 \times 10^{-6}/\max\{t_0, t\}^{0.6}$ with $t_0 = 1000$; and for SGNHT, we set $\epsilon = 0.0001$. Other than the learning rate, pSGLD contains two more tuning parameters, which control the extremes of the curvatures and the balance of the weights of the historical and current gradients, respectively. They both were set to the default values as suggested by Li et al. (2016). SGNHT also contains an extra parameter, the so-called diffusion parameter, for which different values, including 1, 5, 10, and 20, have been tried. The algorithm performed very similarly with each of the choices. Figure S1 summarizes the results of the algorithm with the diffusion parameter being set to 10.

Further, for fairness of comparison, we ran SGLD, pSGLD and SGNHT for 20,000 iterations, 10,000 iterations, and 15,000 iterations, respectively; and they took about 387 CPU seconds, 410 CPU seconds and 380 CPU seconds, respectively. Each of the three algorithms took slightly longer CPU time than LEnKF.

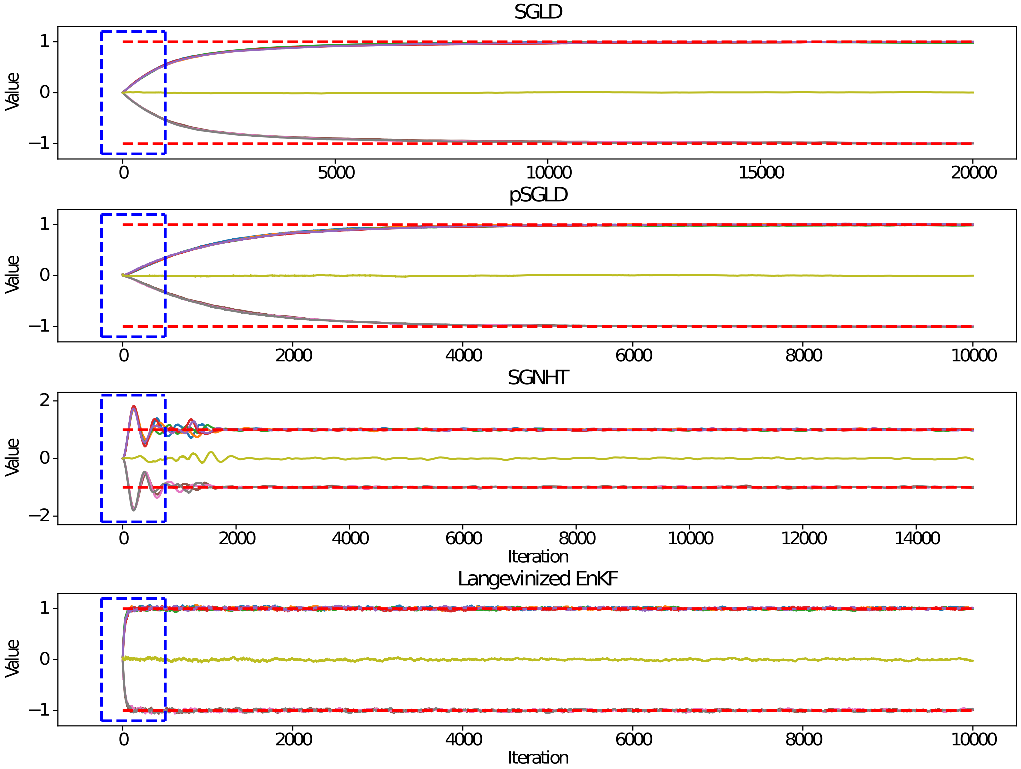

Figure S1: Trajectories of $(\beta_1, \beta_2, \ldots, \beta_9)$ produced by SGLD (upper), pSGLD (upper middle), SGNHT (lower middle), and LEnKF (lower) for a large-scale linear regression example, where the blue rectangle highlights the first 5% iterations of the runs. The highlighted parts are presented in Figure 2 of the main text.

Comparison of LEnKF with parallel SGLD, pSGLD and SGNHT

For a thorough comparison, we also ran SGLD (Welling and Teh, 2011), pSGLD (Li et al., 2016) and SGNHT (Ding et al., 2014) in parallel for the linear regression example considered in Section 4.1. Each of the three algorithms was run for 1,000 iterations with 100 chains and exactly the same parameter setting as used in Section 4.1. The CPU times costed by SGLD, pSGLD, and SGNHT were 38.18, 43.19, 42.39 CPU seconds, respectively.

Recall that LEnKF with an ensemble size of $m = 100$ cost 35.14 CPU seconds for 1,000 iterations on the same computer. Figure S2 shows the trajectories of $(\beta_1, \beta_2, \ldots, \beta_9)$ produced by SGLD, pSGLD, SGNHT and LEnKF in the runs, where each trajectory was obtained by averaging over 100 chains at each iteration. For this example, LEnKF took less than 100 iterations to converge to the true values, SGNHT took about 1000 iterations, while SGLD and pSGLD failed to converge with 1000 iterations.

S1.2 Comparison of LEnKF with sequential importance sampling

To evaluate the performance of LEnKF, we compared it with sequential importance sampling (see e.g. Kantas et al. (2009)). For sequential importance sampling, at each stage $t \in \{2, 3, \ldots, T\}$, we set the trial density functions as $\pi(x_t|x_{t-1})\pi(x_{t-1}|y_{1:t-1})$ for drawing importance particles from $\pi(x_t|y_{1:t})$ based on the particles from $\pi(x_{t-1}|y_{1:t-1})$ and then perform a resampling step to get equally weighted particles. This setting of the trial density function is in parallel to the stage transition procedure of LEnKF, which uses the predictive distribution $\pi(x_t|y_{1:t-1})$ as the prior distribution at each stage. Such a sequential importance sampling algorithm is also called a sequential Monte Carlo (SMC) algorithm, as it generates equally

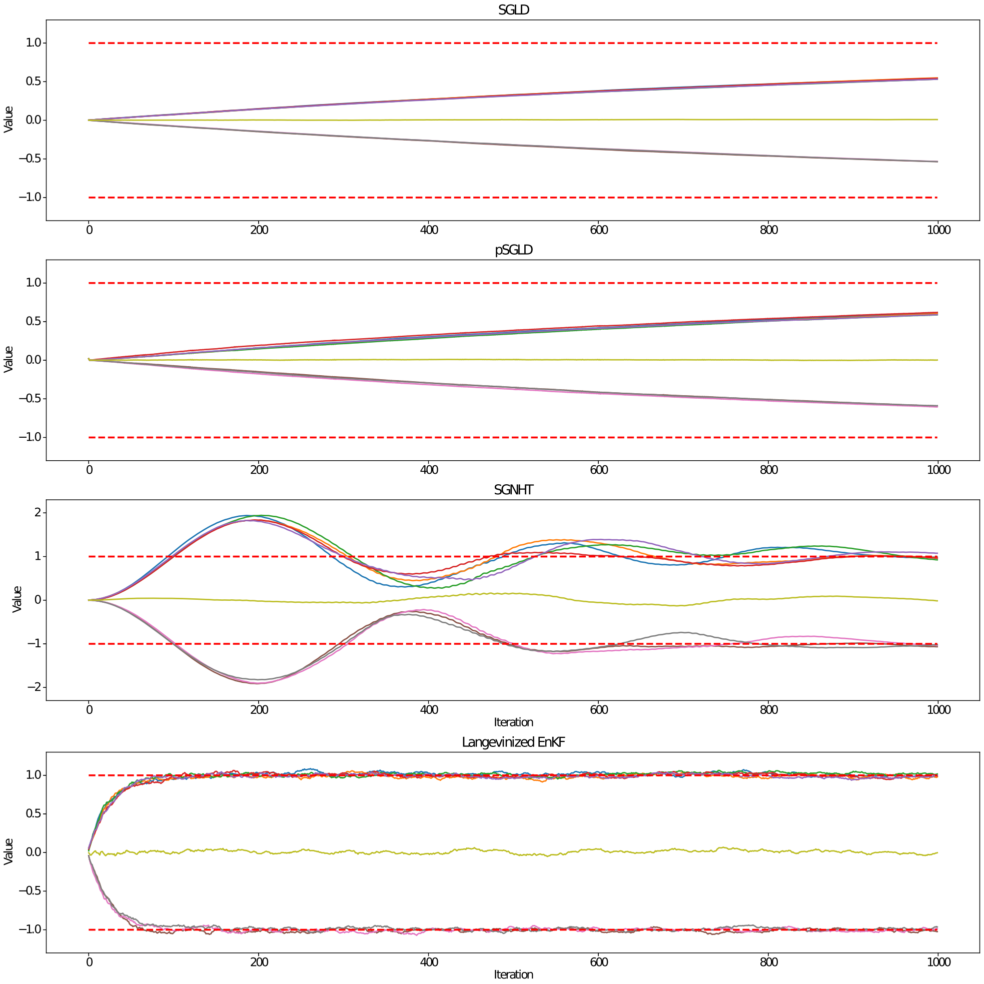

Figure S2: Convergence trajectories of SGLD, pSGLD, SGNHT and LEnKF for a large-scale linear regression example.

weighted particles via resampling at each stage.

The comparison was conducted with the Lorenz-96 model where we set $T = 500$. LEnKF was run with an ensemble size $m = 50$ and the same settings of $\mathcal{K}$ and learning rates as given in Section 4.2. At each stage, the state was estimated by averaging over the ensembles generated in the last iteration, i.e., setting $k_0 = \mathcal{K} - 1$. The run was replicated for 100 times independently, and each run cost 18.4 seconds (with a standard error of 0.23 seconds). SMC was also run for 100 times independently. In each run, we set the population size $m' = 5000$ to maintain its effective sample size at a reasonable size, and estimated the states in the standard way by weighted averaging all particles produced at each stage. Each run cost about 59.7 seconds (with a standard error of 0.61 seconds). With these runs, each algorithm produced 100 estimates for each state $\boldsymbol{X}_t$ for $t = 1, 2, \ldots, T$. Figure S3 shows 100 estimation curves of X_t^{10} produced by each algorithm, which indicate that the estimates produced by SMC have a large variation, while those by LEnKF are more accurate and follow the pattern of the true curve closely, although SMC cost longer CPU time than LEnKF. From the between-runs variation of their estimated curves, we can conclude that LEnKF produced almost the same effective sample size (ESS) at each stage, while the ESS produced by SMC varied with stages. This further implies that LEnKF tends to be immune to sample degeneracy, while SMC doesn't.

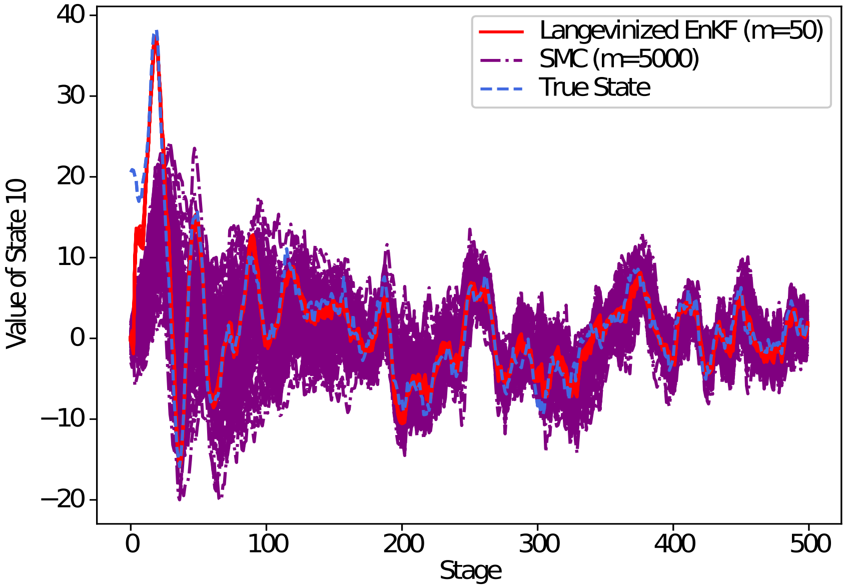

Figure S3: Comparison of LEnKF and SMC: estimated curves of X_t^{10} by LEnKF (red) and SMC (purple) along with stage t in 100 independent runs.

Further, to quantify the relative efficiency of the two algorithms, we calculated the ESS ratio by $R_{ESS,t,10} = m'\sigma^2_{SMC,t,10}/(m\sigma^2_{LEnKF,t,10})$, where $\sigma^2_{SMC,t,10}$ and $\sigma^2_{LEnKF,t,10}$ denote the variance of 100 estimates of X_t^{10} produced by SMC and LEnKF, respectively. Figure S4(a) shows the curve of $R_{ESS,t,10}$ along with stage t. The ESS ratio has very large values around stage 50, which correspond to a high fluctuation of X_t^{10} around the same stage. Figure S4(b) shows the curve of $R_{ESS,t} = \sum_{k=1}^{40} R_{ESS,t,k}/40$, which averages the values of $R_{ESS,t,k}$ over components k. It implies that LEnKF can be much more efficient than SMC especially for the problems with drastically fluctuated state values.

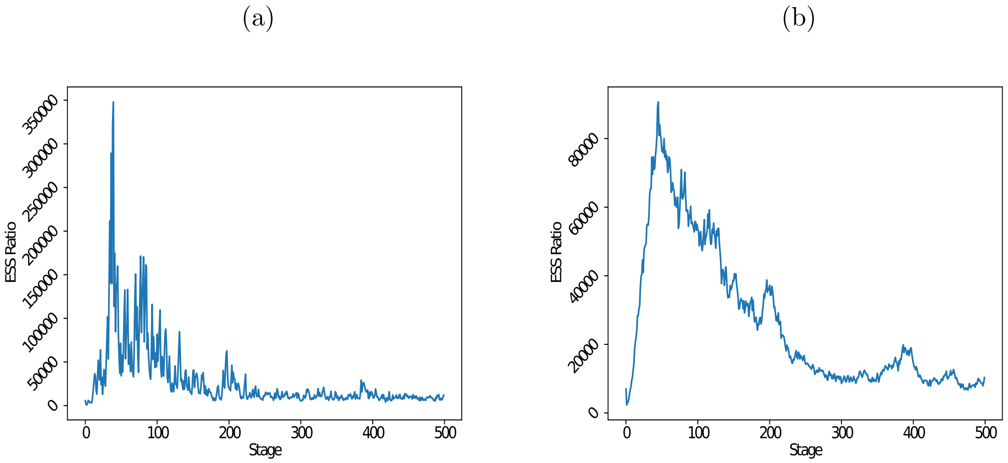

Figure S4: Comparison of LEnKF and SMC: (a) curve of $R_{ESS,t,10}$ along with stage t, which has an average value of 28270.62 over all stages; (b) curve of $R_{ESS,t}$ along with stage t, which has an average value of 26443.01 over all stages.

S1.3 Online Learning with LSTM Neural Networks

Reformulation of LSTM Model The LSTM model is a recurrent neural network model proposed by Hochreiter and Schmidhuber (1997), which has been widely used for machine learning tasks in dealing with time series data. Compared to traditional recurrent neural networks, hidden Markov models and other sequence learning methods, LSTM is less sensitive to gap length of the data sequence. In addition, it is easy to train, less bothered by exploding and vanishing gradient problems. The LSTM model has been successfully used in natural language processing and handwriting recognition.

It won the ICDAR handwriting competition 2009 (Graves et al., 2009) and achieved a record 17.7% phoneme error rate on the classic TIMIT natural speech dataset (Graves et al., 2013). In this section we show that LEnKF is not only able to train LSTM models as the stochastic gradient descent (SGD) method, but also able to quantify uncertainty of the estimates for the quantities of interest.

Consider an autoregressive model of order q, denoted by AR(q). Let $\boldsymbol{z}_t = (z_{t-q+1}, \cdots, z_{t-1}, z_t)$ denote the regression vector at stage t. Let $y_t = z_{t+1} \in \mathbb{R}^d$ denote the target output at stage t. The LSTM model with s hidden neurons is defined by the following set of equations:

$$\begin{aligned}
\eta_t &= h\left(\boldsymbol{W}^{(\eta)}\boldsymbol{z}_t + \boldsymbol{R}^{(\eta)}\boldsymbol{\psi}_{t-1} + \boldsymbol{b}^{(\eta)}\right), \\
\boldsymbol{i}_t &= \sigma\left(\boldsymbol{W}^{(i)}\boldsymbol{z}_t + \boldsymbol{R}^{(i)}\boldsymbol{\psi}_{t-1} + \boldsymbol{b}^{(i)}\right), \\
\boldsymbol{f}_t &= \sigma\left(\boldsymbol{W}^{(f)}\boldsymbol{z}_t + \boldsymbol{R}^{(f)}\boldsymbol{\psi}_{t-1} + \boldsymbol{b}^{(f)}\right), \\
\boldsymbol{c}_t &= \boldsymbol{\Lambda}_t^{(i)}\eta_t + \boldsymbol{\Lambda}_t^{(f)}\boldsymbol{c}_{t-1}, \\
\boldsymbol{o}_t &= \sigma\left(\boldsymbol{W}^{(o)}\boldsymbol{z}_t + \boldsymbol{R}^{(o)}\boldsymbol{\psi}_{t-1} + \boldsymbol{b}^{(o)}\right), \\
\boldsymbol{\psi}_t &= \boldsymbol{\Lambda}_t^{(o)}h\left(\boldsymbol{c}_t\right),
\end{aligned} \tag{S1.1}$$

where $\boldsymbol{\Lambda}_t^{(f)} = \operatorname{diag}\left(\boldsymbol{f}_t\right), \boldsymbol{\Lambda}_t^{(i)} = \operatorname{diag}\left(\boldsymbol{i}_t\right)$, and $\boldsymbol{\Lambda}_t^{(o)} = \operatorname{diag}\left(\boldsymbol{o}_t\right)$. The activation function $h(\cdot)$ applies to vectors pointwisely and is commonly set to $tanh(\cdot)$. The sigmoid function $\sigma(\cdot)$ also applies pointwisely to the vector elements. In terms of LSTM models, $\boldsymbol{z}_t \in \mathbb{R}^{qd}$ is called input vector, $\boldsymbol{c}_t \in \mathbb{R}^s$ is called the state vector, $\boldsymbol{\psi}_t \in \mathbb{R}^s$ is called the output vector, and $\boldsymbol{i}_t$, $\boldsymbol{f}_t$ and $\boldsymbol{o}_t$ are called the input, forget and output gates, respectively. For the coefficient matrices and weight vectors, we have $\boldsymbol{W}^{(\eta)}, \boldsymbol{W}^{(i)}, \boldsymbol{W}^{(f)}, \boldsymbol{W}^{(o)} \in \mathbb{R}^{s\times qd}$, $\boldsymbol{R}^{(\eta)}, \boldsymbol{R}^{(i)}, \boldsymbol{R}^{(f)}, \boldsymbol{R}^{(o)} \in \mathbb{R}^{s\times s}$, and $\boldsymbol{b}^{(\eta)}, \boldsymbol{b}^{(i)}, \boldsymbol{b}^{(f)}, \boldsymbol{b}^{(o)} \in \mathbb{R}^s$. For initialization, we set $\boldsymbol{\psi}_0 = \boldsymbol{0}$, and $\boldsymbol{c}_0 = \boldsymbol{0}$. Given the output vector $\boldsymbol{\psi}_t$, we can model the target output y_t as

$$y_t = \boldsymbol{W}\boldsymbol{\psi}_t + \boldsymbol{b} + \boldsymbol{u}_t, \tag{S1.2}$$

where $\boldsymbol{W} \in \mathbb{R}^{d\times s}, \boldsymbol{b} \in \mathbb{R}^d$, and $\boldsymbol{u}_t \sim N(0, \Gamma_t)$.

For convenience, we group the parameters of the LSTM model as $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{W}^{(\eta)}, \boldsymbol{R}^{(\eta)}, \boldsymbol{b}^{(\eta)}, \boldsymbol{W}^{(i)},\ \boldsymbol{R}^{(i)}, \boldsymbol{b}^{(i)}, \boldsymbol{W}^{(f)}, \boldsymbol{R}^{(f)}, \boldsymbol{b}^{(f)}, \boldsymbol{W}^{(o)}, \boldsymbol{R}^{(o)}, \boldsymbol{b}^{(o)}\} \in \mathbb{R}^{n_\theta}$, where $n_\theta = 4s^2 + 4sqd + 4s + sd + d$. With the state-augmentation approach, we can rewrite the LSTM model as a state-space model with a linear measurement equation as follows:

$$\begin{bmatrix} \boldsymbol{\theta}_t \\ \boldsymbol{c}_t \\ \boldsymbol{\psi}_t \\ \boldsymbol{\gamma}_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_{t-1} \\ \boldsymbol{\Omega}\left(\boldsymbol{c}_{t-1}, \boldsymbol{z}_t, \boldsymbol{\psi}_{t-1}\right) \\ \tau\left(\boldsymbol{c}_t, \boldsymbol{z}_t, \boldsymbol{\psi}_{t-1}\right) \\ \boldsymbol{W}_t\boldsymbol{\psi}_t + \boldsymbol{b} \end{bmatrix} + \begin{bmatrix} \boldsymbol{e}_t \\ \boldsymbol{\zeta}_t \\ \boldsymbol{\xi}_t \\ \boldsymbol{\varepsilon}_t \end{bmatrix}, \tag{S1.3}$$

$$y_t = \boldsymbol{\gamma}_t + \boldsymbol{v}_t,$$

where $\boldsymbol{\varepsilon}_t \sim N(0, \alpha\Gamma_t)$ for some constant $0 < \alpha_t < 1$, $\boldsymbol{v}_t \sim N(0, (1-\alpha)\Gamma_t)$, and Γ_t is as defined in (S1.2). Let $\boldsymbol{x}_t^T = (\boldsymbol{\theta}_t^T, \boldsymbol{c}_t^T, \boldsymbol{\psi}_t^T, \boldsymbol{\gamma}_t^T)$. Then

$$\pi(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{z}_t) = \pi(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \boldsymbol{z}_t)\pi(\boldsymbol{c}_t|\boldsymbol{\theta}_t, \boldsymbol{c}_{t-1}, \boldsymbol{\psi}_{t-1}, \boldsymbol{z}_t)\pi(\boldsymbol{\psi}_t|\boldsymbol{\theta}_t, \boldsymbol{c}_t, \boldsymbol{\psi}_{t-1}, \boldsymbol{z}_t)$$
$$\times \pi(\boldsymbol{\gamma}_t|\boldsymbol{\theta}_t, \boldsymbol{\psi}_t).$$

As in (3.9), we can rewrite the state-space model (S1.3) as a dynamic system:

$$\begin{aligned} \boldsymbol{x}_{t,k} &= \boldsymbol{x}_{t,k-1} + \frac{\epsilon_t}{2}\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{x}_{t,k-1}|\tilde{\boldsymbol{x}}_{t-1,k-1}, \boldsymbol{z}_t) + \boldsymbol{\omega}_{t,k} \\ y_{t,k} &= H_t \boldsymbol{x}_{t,k} + \boldsymbol{v}_{t,k}, \end{aligned} \tag{S1.4}$$

where $\boldsymbol{x}_{t,k}$ denote an estimate of $\boldsymbol{x}_t$ obtained at iteration k for $k = 1, 2, \ldots, \mathcal{K}$, $y_{t,k} = y_t$ for $k = 1, 2, \ldots, \mathcal{K}$, $H_t = (0, I)$ such that $H_t x_t = \boldsymbol{\gamma}_t$, $\boldsymbol{\omega}_{t,k} \sim N(0, \epsilon_t I_p)$, p is the dimension of $\boldsymbol{x}_t$, and $\boldsymbol{v}_{t,k} \sim N(0, (1-\alpha)\Gamma_t)$. With this formulation, Algorithm 3 can be applied to train the LSTM model and quantify uncertainty of the estimates for the quantities of interest.

Wind Stress Data We considered the wind stress dataset, which can be downloaded at https:// iridl.ldeo.columbia.edu. The dataset consists of gridded (at a 2×2 degrees resolution and corresponding to $d = 1470$ spatial locations) monthly summaries of meridional wind pseudo-stress collected from Jan 1961 to Feb 2002. For this dataset, we set $q = 6$ and $T = 300$, i.e., modeling the data of the first 300 months using an AR(6) LSTM model. The data was scaled into the range $(-1, 1)$ in preprocessing and then scaled back to the original range in results reporting.

LEnKF was first applied to this example. For the model part, we set $\boldsymbol{e}_t \sim N(0, 0.0001I)$, $\boldsymbol{\zeta}_t \sim N(0, 0.0001I)$, $\boldsymbol{\xi}_t \sim N(0, 0.0001I)$, $\boldsymbol{u}_t \sim N(0, 0.0001I)$. These model parameters are assumed to be known, although this is not necessary as discussed at the end of the paper. For this example, we have tried different settings for the model parameters. In general, a smaller variance setting will lead to a better fitting to the observations. For the algorithmic part, we set the ensemble size $m = 100$, $\mathcal{K} = 10$, $k_0 = 9$, $\alpha = 0.9$, the number of hidden neurons $s = 20$, and the learning rate $\epsilon_{t,k} = 0.0001/\max\{\kappa_b, k\}^{0.95}$ with $\kappa_b = 8$ for $k = 1, \cdots, \mathcal{K}$ and $t = 1, \cdots, T$. At each stage t, the wind stress was estimated by averaging over $\hat{y}_{t,k} = H_t \boldsymbol{x}_{t,k}$ for last $\mathcal{K}/2$ iterations. In addition, the credible interval for each component of $\boldsymbol{x}_t$ was calculated based on the ensemble obtained at stage t. Each run cost about 5334.5 CPU seconds. The results are summarized in Figure S5, where the wind stress estimates at four selected spatial locations and their 95% credible intervals are plotted along with stages.

For comparison, SGD was also applied to this example with the same setting as LEnKF, i.e., they share the same learning rate and the same iteration number $\mathcal{K} = 10$ at each stage. The results are also summarized in Figure S5, where the wind stress estimates at four selected spatial locations are plotted along with stages. Each run of SGD cost about 15.9 CPU seconds. Since LEnKF had an ensemble size $m = 100$, each chain cost only 53.3 CPU seconds. LEnKF cost more CPU time and as return, it produced more samples for uncertainty quantification.

Further, we calculated the mean squared fitting error $||\hat{y}_t - y_t||_2^2$ for stages $t = 1, 2, \ldots, T$ and for both methods. The results are summarized in Figure S6, which indicates that LEnKF produced slightly smaller fitting errors than SGD. Figure S7 shows the heat maps of the wind stress fitted by LEnKF and SGD for six different months, August 1965, October 1969, December 1973, February 1978, April 1982, and June 1986. The comparison with the true heat maps indicates that both SGD and LEnKF can train the LSTM model very well for this example.

In summary, this example shows that LEnKF is not only able to train LSTM model as does SGD, but also able to quantify uncertainty of the estimates for the quantities of interest.

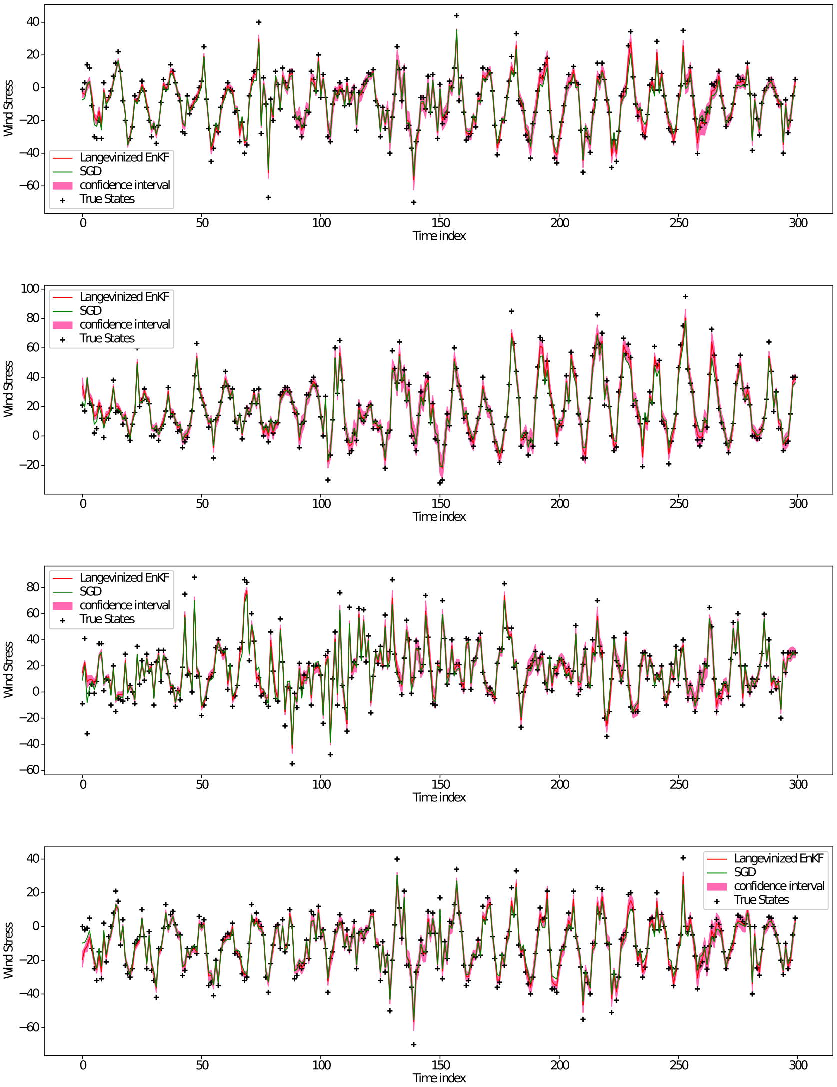

Figure S5: Wind stress estimates at four spatial locations and their 95% credible interval along with stages: the red line is for the LEnKF estimate; the pink shaded band is for credible intervals of LEnKF, the green line is for the SGD estimate; and the blue cross '+' is for the true wind stress value.

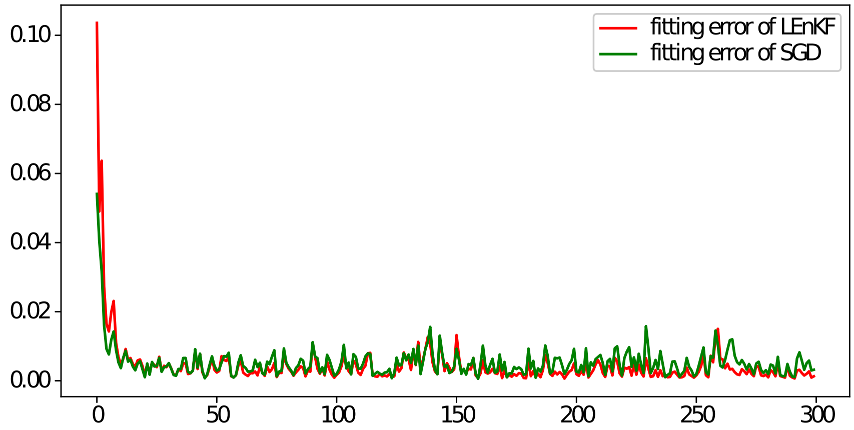

Figure S6: Comparison of the mean squared fitting errors produced by SGD, LEnKF and ensemble averaging LEnKF.

S2 Convergence Theory of LEnKF

We are interested in studying the convergence of LEnKF in 2-Wasserstein distance. The r-Wasserstein distance between two probability measures μ and ν is defined by

$$W_r(\mu,\nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{X}\times\mathbb{X}} d(x,y)^r d\pi(x,y) \right)^{1/r} = \inf_{\pi \in \Pi(\mu,\nu)} \{E_\pi d(X,Y)^r\}^{1/r},$$

where $\Pi(\mu,\nu)$ denotes the collection of all probability measures on $\mathbb{X}\times\mathbb{X}$ with marginals μ and ν respectively.

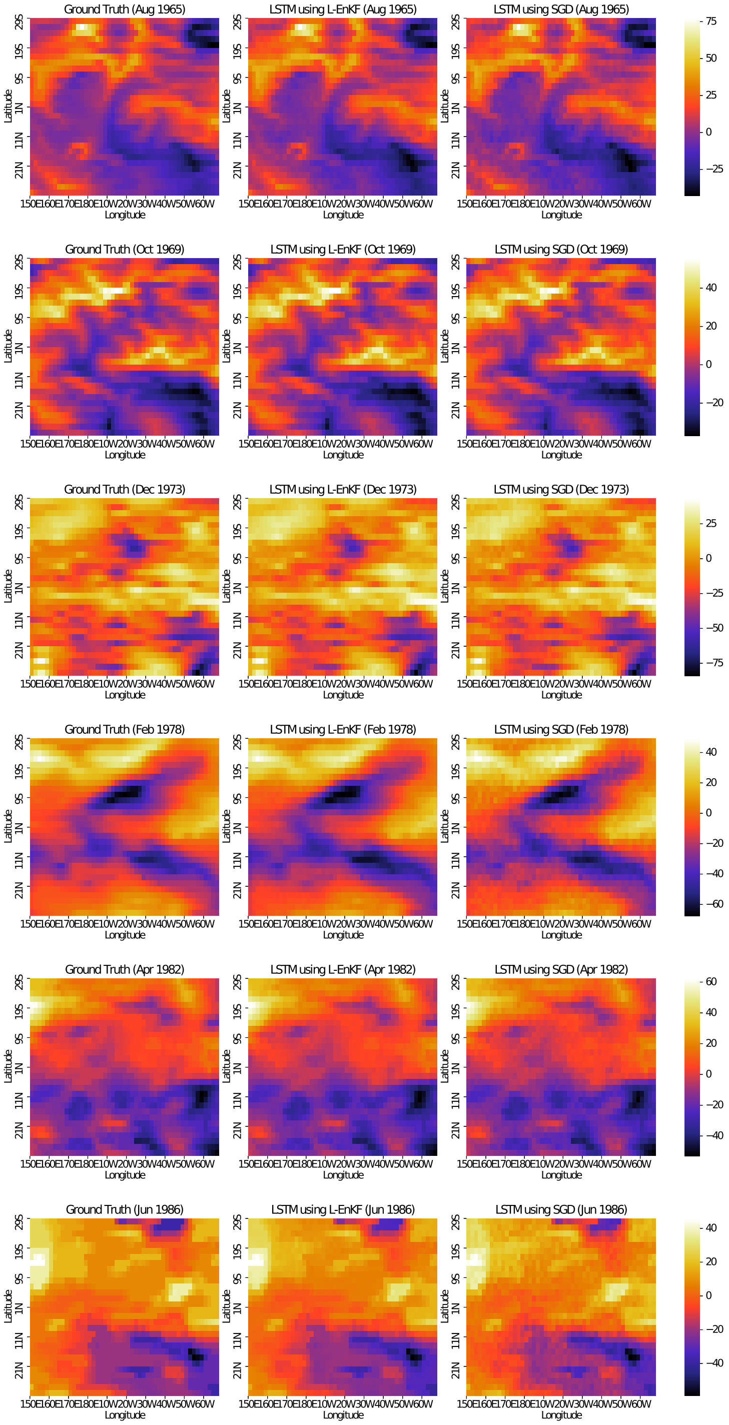

Figure S7: Heat maps of the wind stress fitted by LEnKF and SGD for six different months, August 1965, October 1969, December 1973, February 1978, April 1982, and June 1986: For both left and right panels, the left, middle and right columns show the true heat map, the heat map fitted by LEnKF, and the heat map fitted by SGD, respectively.

S2.1 Proof of Theorem 1

Proof. First, we consider the Kalman gain matrix $K_t = Q_t H_t^T (R_t + H_t Q H_t^T)^{-1}$, which, with some algebra, can be shown

$$K_t = (I - K_t H_t) Q_t H_t^T R_t^{-1} = (H_t^T R_t^{-1} H_t + Q_t^{-1})^{-1} H_t^T R_t^{-1}. \qquad \text{(S2.1)}$$

Let $\mu_t = E(x_t^f | x_{t-1}^a) = x_{t-1}^a + \delta_t$, where $\delta_t = \epsilon_t \frac{n}{2N} \nabla \log \pi(x_{t-1}^a)$. Therefore, $x_t^f = \mu_t + w_t$.

Taking conditional expectation on both sides of equation (3.5), we have

$$E(x_t^a | x_{t-1}^a) = \mu_t + K_t(y_t - H_t \mu_t) = x_t^f + K_t(y_t - H_t x_t^f) - (I - K_t H_t) w_t. \quad \text{(S2.2)}$$

With the identity (S2.1), (S2.2) can be further written as

$$
\begin{aligned}
&E(x_t^a|x_{t-1}^a) = x_{t-1}^a + \delta_t + K_t(y_t - H_t x_{t-1}^a - H_t\delta_t) \\
&= x_{t-1}^a + K_t(y_t - H_t x_{t-1}^a) + (I - K_t H_t)\delta_t \\
&= x_{t-1}^a + (I - K_t H_t) Q_t H_t^T R_t^{-1}(y_t - H_t x_{t-1}^a) + (I - K_t H_t)\delta_t \\
&= x_{t-1}^a + (I - K_t H_t) Q_t \left[H_t^T R_t^{-1}(y_t - H_t x_{t-1}^a) + Q_t^{-1}\delta_t\right] \\
&= x_{t-1}^a + \frac{n}{2N}(I - K_t H_t) Q_t \left[\frac{N}{n} H_t^T V^{-1}(y_t - H_t x_{t-1}^a) + \nabla \log \pi(x_{t-1}^a)\right], \\
&= x_{t-1}^a + \frac{\epsilon_t}{2}\Sigma_t \left[\frac{N}{n} H_t^T V^{-1}(y_t - H_t x_{t-1}^a) + \nabla \log \pi(x_{t-1}^a)\right],
\end{aligned}
\tag{S2.3}
$$

by defining $\Sigma_t = \frac{n}{N}(I - K_t H_t)$ and by noting $Q_t = \epsilon_t I_p$ and $R_t = 2V$.

For LEnKF, the difference between equations (3.5) and (S2.2) is

$$
e_t = (I - K_t H_t) w_t - K_t v_t = w_t - K_t(H_t w_t + v_t),
$$

for which the mean $E(e_t) = 0$ and the covariance is given by

$$
\begin{aligned}
\mathrm{Var}(e_t) &= \frac{n}{N} Q_t + K_t(\frac{n}{N} H_t Q_t H_t^T + \frac{n}{N} R_t) K_t^T - 2\frac{n}{N} K_t H_t Q_t \\
&= \frac{n}{N}\left[Q_t + K_t H_t Q_t - 2K_t H_t Q_t\right] \\
&= \frac{n}{N}(I - K_t H_t) Q_t = \epsilon_t \Sigma_t,
\end{aligned}
$$

where the second equality holds due to the symmetry of Q_t and R_t and the identity $K_t(H_tQ_tH_t^T+R_t)K_t^T = K_t(H_tQ_tH_t^T+R_t)(H_tQ_t^TH_t^T+R_t^T)^{-1}H_tQ_t^T = K_tH_tQ_t$. Then, by (S2.3), we have

$$\begin{aligned} x_t^a &= x_t^f + K_t\left[y_n - H_tx_t^f - v_t\right] \\ &= x_{t-1}^a + \frac{\epsilon_t}{2}\Sigma_t\left[\frac{N}{n}H_t^TV_t^{-1}(y_t - H_tx_{t-1}^a) + \nabla\log\pi(x_{t-1}^a)\right] + e_t, \end{aligned} \tag{S2.4}$$

where $\frac{N}{n}H_t^TV^{-1}(y_t - H_tx_{t-1}^a)$ represents an unbiased estimator for the gradient of the log-likelihood function, and $\nabla\log\pi(x_{t-1}^a)$ represents the gradient of the log-prior density function. The proof can then be concluded. □

Remark S1. Let $\tilde{\pi}_t$ denote the empirical distribution of x_t^a, and let $\pi_* = \pi(x|y)$ denotes the target posterior distribution $\pi(x|y)$. By Corollary S1 (with $\eta = 0$), we have $\lim_{t\to\infty} W_2(\tilde{\pi}_t, \pi_*) = 0$. For Algorithm 2, it is easy to see that (S2.8) is satisfied, for which the bias factor $\eta = 0$ as $\frac{N}{n}H_t^TV^{-1}(y_t - H_tx_{t-1}^a) + \nabla\log\pi(x_{t-1}^a)$ forms an unbiased estimator of $\nabla\log\pi(x|y)$ at x_{t-1}^a, and the variance of the estimation error is upper bounded by a quadratic function of $\|x_{t-1}^a\|$.

S2.2 Proof of Theorem 2

Let $\pi_t = \pi(x_t|y_{1:t})$ denote the filtering distribution at stage t, and let $\tilde{\pi}_t$ denote the marginal distribution of $x_{t,\mathcal{K}}^{a,i}$ generated by Algorithm 3 at iteration $\mathcal{K}$ of stage t. To study the convergence of Algorithm 3, we make the following assumptions for the dynamic system (1.1):

Assumption S1. π_t *is* s_t*-strongly log-concave:*

$$f(x_t) - f(x_t') - \nabla f(x_t')^T (x_t - x_t') \geq \frac{s_t}{2} \|x_t - x_t'\|_2^2, \quad \forall x_t, x_t' \in \mathbb{R}_t^p, \quad \text{(S2.5)}$$

where $f(x_t) = -\log \pi(x_t|y_{1:t}) = -\log \pi_t$*, and* s_t *is a positive number satisfying* $s_t \geq c_1 N_t$ *for some constant* $c_1 > 0$*.*

Assumption S2. $\log(\pi_t)$ *is* S_t*-gradient Lipschitz continuous:*

$$\|\nabla f(x_t) - \nabla f(x_t')\|_2 \leq S_t \|x_t - x_t'\|_2, \quad \forall x_t, x_t' \in \mathbb{R}_t^p. \quad \text{(S2.6)}$$

where S_t *is a positive number satisfying* $S_t \leq c_2 N_t$ *for some constant* $c_2 > 0$*. Note that we must have* $s_t \leq S_t$*.*

Assumption S3. *Let* $\Sigma_{t,k} = \frac{n}{N}(I - K_{t,k} H_{t,k})$*, and assume that*

$$\lambda_{t,l} \leq \inf_k \lambda_{\min}(\Sigma_{t,k}) \leq \sup_k \lambda_{\max}(\Sigma_{t,k}) \leq \lambda_{t,u}$$

for some $\lambda_{t,l}$ *and* $\lambda_{t,u}$*, where* $\lambda_{\max}(\cdot)$ *and* $\lambda_{\min}(\cdot)$ *denote the largest and smallest eigenvalues, respectively. In addition, there exist constants* $c_3 > 0$ *and* $c_4 > 0$ *such that* $c_3(n_t/N_t) \leq \lambda_{t,l} \leq \lambda_{t,u} \leq c_4(n_t/N_t)$*.*

Assumption S4. $\lambda_{\max}(H_{t,k}^T V_t^{-1} H_{t,k}) \leq c_5 n_t$ *for some constant* $c_5 > 0$*.*

Assumption S5. *The stochastic error induced by the sub-sampling procedure has a bounded variance, i.e.,* $\forall x \in \mathbb{R}_t^p$*,*

$$E[\|(N_t/n_t)H_{t,k}^T V_t^{-1}(y_{t,k} - H_{t,k}x) - H_t^T \Gamma_t^{-1}(y_t - H_t x)\|^2] \leq \sigma_{t,s}^2(p + \|x\|^2),$$

for some constant $\sigma_{t,s}^2 > 0$*, where the expectation is with respect to random sub-sampling. In addition, we assume that* $s_t\lambda_{t,l} - \sqrt{2}\sigma_{t,s}\lambda_{t,u} > 0$*.*

Assumption S6. *The state propagator* $g(x_t)$ *is* l*-Lipschitz and bounded by* M_g *(i.e.,* $\sup_x \|g(x)\| \leq M_g$*), and* $\lambda'_{t,s} \geq \lambda_{\max}(U_t) \geq \lambda_{\min}(U_t) \geq \lambda_{t,s} > 0$ *for some positive constants* $\lambda'_{t,s}$ *and* $\lambda_{t,s}$*.*

Assumption S7. *There exist some constant* M *such that* $W_2(\nu_{t+1}, \pi_{t+1}) \leq M$ *for all* $t \geq 0$*, where* $\nu_{t+1}(x_{t+1}) = \int \pi(x_{t+1}|x_t)\pi_t(x_t)dx_t$ *is the ideal stage initial distribution of* $x_{t+1,0}^{a,i}$ *for* $t \geq 1$*, and* ν_1 *is the initial distribution used at stage 1. Similarly, we define* $\tilde{\nu}_{t+1}(x_{t+1}) = \int \pi(x_{t+1}|x_t)\tilde{\pi}_t(x_t)dx_t$ *to be the practical stage initial distribution of* $x_{t+1,0}^{a,i}$ *for* $t \geq 1$*.*

Assumption S8. *There exists a constant* c_7 *such that* $\widetilde{V}_t = \int \|x\|^2 d\pi_t \leq c_7 p$*.*

Remark S2. Log-concavity and strong log-concavity are preserved by products and marginalization (Saumard and Wellner, 2014). If the prior density $\pi(x_1)$ is log-concave, the state transition density $\pi(x_t|x_{t-1})$ is log-concave with respect to both x_t and x_{t-1} for each stage t, and the emission density $\pi(y_t|x_t)$ is λ_t-strongly-log-concave for each stage t where λ_t is the smallest eigenvalue of $H_t^T\Gamma_t^{-1}H_t$, then Assumption S1 holds with $s_t = \lambda_t$. Furthermore, by Brascamp-Lieb inequality (Brascamp and Lieb, 2002), we must have that π_t has finite variance, that is

$$p\sigma_{t,v}^2 := E_{\pi_t}\|X - E(X)\|^2 \leq p/s_t. \tag{S2.7}$$

Strongly log-concave conditions are commonly used in the theoretical study of Langevin Monte Carlo, see e.g. Dalalyan and Karagulyan (2019) and Cheng and Bartlett (2018). These conditions potentially can be relaxed following the work of Durmus and Moulines (2017).

Remark S3. With some simple linear algebra, we can show

$$\Sigma_{t,k} = \frac{n_t}{N_t}(I - K_{t,k}H_{t,k}) = \frac{n_t}{N_t}(I - \epsilon_{t,k}H_{t,k}^T(\epsilon_{t,k}H_{t,k}H_{t,k}^T + 2V_t)^{-1}H_{t,k}),$$

which implies that all eigenvalues of $\Sigma_{t,k}$ lie in the range $(0, n_t/N_t)$. Thus, it is reasonable to assume that $\lambda_{t,l}, \lambda_{t,u} \asymp n_t/N_t$. In addition, due to random subsampling used in stochastic gradient evaluation, it is natural to assume that $\sigma_{t,s}^2 \asymp (N_t/n_t)$. Therefore, $s_t\lambda_{t,l} - \sqrt{2}\sigma_{t,s}\lambda_{t,u} > 0$ holds trivially by Assumption S1 and large subsample size n_t.

Remark S4. Assumption S7 says the ideal initialization distribution at each stage is not too bad, which essentially requires that the data are coherent to the state space model (1.1) in the sense that the predictive distribution based on $y_{1:t}$ (i.e., $\pi(x_{t+1}|y_{1:t})$) and the state estimate based on y_{t+1} only (i.e., $\hat{x}_{t+1} = (H_{t+1}^T H_{t+1})^{-1} H_{t+1}^T y_{t+1}$) are close.

Lemma S1. *Let μ and ν be two distribution laws on $\mathbb{R}^p$, and let f be an L-Lipschitz continuous function, then*

$$\left\| \int f(x) d\mu(x) - \int f(x) d\nu(x) \right\| \leq L W_2(\mu, \nu).$$

Proof. By the definition of 2-Wasserstein distance, there exist random variables X_1 and X_2, whose marginal distributions follow μ and ν respectively, such that $\|X_1 - X_2\|_{L_2} = (E\|X_1 - X_2\|_2^2)^{1/2} = W_2(\mu, \nu)$.

$$\begin{aligned}
&\left\| \int f(x) d\mu(x) - \int f(x) d\nu(x) \right\| = \|Ef(X_1) - Ef(X_2)\| \\
&\leq E\|f(X_1) - f(X_2)\| \leq EL\|X_1 - X_2\|_2 \\
&= LE\sqrt{\|X_1 - X_2\|_2^2} \leq L\sqrt{E\|X_1 - X_2\|_2^2} = LW_2(\mu, \nu).
\end{aligned}$$

□

Lemma S2. *If Assumption S6 holds, then*

$$W_2(\tilde{\nu}_{t+1}, \nu_{t+1}) \leq l W_2(\pi_t, \tilde{\pi}_t).$$

Proof. By the definition of 2-Wasserstein distance, there exist random variables X_1 and X_2, whose marginal distributions are π_t and $\tilde{\pi}_t$ respectively, and $E(\|X_1 - X_2\|_2^2) = W_2^2(\tilde{\pi}_t, \pi_t)$. Define $Y_1 = g(X_1) + u$, and $Y_2 = g(x_2) + u$, where $u \sim N(0, U_{t+1})$ such that the marginal distributions of Y_1 and Y_2 are ν_{t+1} and $\tilde{\nu}_{t+1}$ respectively. Then,

$$\begin{aligned} & W_2^2(\tilde{\nu}_{t+1}, \nu_{t+1}) \leq E\|Y_1 - Y_2\|_2^2 = E\|g(X_1) - g(X_2)\|_2^2 \\ & \leq E l^2 \|X_1 - X_2\|_2^2 = l^2 W_2^2(\pi_t, \tilde{\pi}_t). \end{aligned}$$

□

Lemma S3. *If f is an L-Lipschitz continuous function, then* $E\|f(X) - E(f(X))\|_2^2 \leq L^2 E\|X - E(X)\|_2^2$.

Proof. Let X_1 and X_2 be two independent copies of X. Then

$$\begin{aligned}E\|f(X)-E(f(X))\|_2^2 &= (1/2)E\|f(X_1)-f(X_2)\|_2^2\\ &\le (1/2)E(L\|X_1-X_2\|_2)^2\\ &\le L^2(1/2)E(\|X_1-X_2\|_2)^2\\ &= L^2E\|X-E(X)\|_2^2.\end{aligned}$$

□

Lemma S4. *Let $X\sim\mu$ and $Y\sim\nu$, then*

$$E\|Y-E(Y)\|_2^2\le E\|X-E(X)\|_2^2+W_2^2(\mu,\nu)+2W_2(\mu,\nu)\sqrt{E\|X-E(X)\|_2^2}.$$

Proof. By definition of Wasserstein metric, we can assume that X and Y satisfy $\|X-Y\|_{L_2}=(E\|X-Y\|_2^2)^{1/2}=W_2(\mu,\nu)$. Without loss of generality, we also assume that EX, the mean of measure μ, is 0. Then

$$\begin{aligned}&[E\|Y-E(Y)\|_2^2-E\|X\|_2^2]-W_2^2(\mu,\nu)\\ =&EY^TY-(EY)^T(EY)-EX^TX-EX^TX-EY^TY+2EX^TY\\ =&2EX^TY-2EX^TX-(EY)^T(EY)\le 2EX^T(Y-X)\le 2E\|X\|_2\|Y-X\|_2\\ \le&2\sqrt{E\|X\|_2^2E\|Y-X\|_2^2}=2W_2(\mu,\nu)\sqrt{E\|X\|_2^2}.\end{aligned}$$

□

Lemma S5. *Let $X \sim \mu$ and $Y \sim \nu$, then*

$$E\|Y\|^2 \leq E\|X\|^2 + W_2^2(\mu,\nu) + 2W_2(\mu,\nu)\sqrt{E\|X\|^2}.$$

Proof. By definition of Wasserstein metric, W.O.L.G, we can assume that X and Y satisfy $E\|X-Y\|^2 = W_2^2(\mu,\nu)$. Then

$$\begin{aligned}
&[E\|Y\|^2 - E\|X\|^2] - W_2^2(\mu,\nu)\\
=&EY^TY - EX^TX - EX^TX - EY^TY + 2EX^TY\\
=&2EX^TY - 2EX^TX = 2EX^T(Y-X) \leq 2E\|X\|\|Y-X\|\\
\leq&2\sqrt{E\|X\|^2E\|Y-X\|^2} = 2W_2(\mu,\nu)\sqrt{E\|X\|^2}.
\end{aligned}$$

□

Lemma S6 is a generalization of Theorem 4 of Dalalyan and Karagulyan (2019), and a generalization of Lemma S2 of Song et al. (2020) as well.

Lemma S6. *Let x_k and x_{k+1} be two random vectors in $\mathbb{R}^p$ satisfying*

$$x_{k+1} = x_k - \epsilon\Sigma[\nabla f(x_k) + \zeta_k] + \sqrt{2\epsilon}e_{k+1},$$

where $e_{k+1} \sim N(0,\Sigma)$, and ζ_k denotes the random error of the gradient estimate which can depend on x_k. Let π_k be the distribution of x_k, and let

$\pi_* \propto \exp\{-f\}$ *be the target distribution. Suppose that* ζ_k *satisfies*

$$\|E(\zeta_k|x_k)\|^2 \leq \eta^2 p, \quad E[\|\zeta_k - E(\zeta_k|x_k)\|^2] \leq \sigma_1^2 p + \sigma_2^2 \|x_k\|^2, \tag{S2.8}$$

for some constants η *and* σ*, and* ζ_k*'s are independent of* e_{k+1}*'s. If the function* f *is* s*-strongly convex and* S*-gradient-Lipschitz,* $\lambda_{\min}(\Sigma) = \lambda_l$*,* $\lambda_{\max}(\Sigma) = \lambda_u$*, and the learning rate* $\epsilon \leq 2/(s\lambda_l + S\lambda_u)$*, then*

$$\begin{aligned} W_2^2(\pi_{k+1}, \pi_*) \leq \Big[(1 - \lambda_l s\epsilon + \sqrt{2}\sigma_2\lambda_u\epsilon)W_2(\pi_k, \pi_*) + 1.65S(\lambda_u^3\epsilon^3 p)^{1/2} + \epsilon\eta\lambda_u\sqrt{p}\Big]^2 \\ + \epsilon^2\sigma_1^2\lambda_u^2 p + 2\epsilon^2\sigma_2^2\lambda_u^2 p\widetilde{V}, \end{aligned} \tag{S2.9}$$

where $\widetilde{V} = \int \|x\|^2 \pi_*(x)dx$*.*

Proof. First of all, the updating iteration can be rewritten as:

$$\tilde{x}_{k+1} = \tilde{x}_k - \epsilon[\nabla\tilde{f}(\tilde{x}_k) + \tilde{\zeta}_k] + \sqrt{2\epsilon}\tilde{e}_{k+1}, \tag{S2.10}$$

where $\tilde{f}(x) = f(\Sigma^{1/2}x)$, $\tilde{x}_k = \Sigma^{-1/2}x_k$, $\tilde{\zeta}_k = \Sigma^{1/2}\zeta_k$ and $\tilde{e}_{k+1} \sim N(0, I)$.

Let $\tilde{\pi}_*$ denote the distribution $\tilde{\pi}_* \propto \exp\{-\tilde{f}\}$. It is easy to see that the distribution $\tilde{\pi}_*$ is $s\lambda_l$-strongly log-concave and $S\lambda_u$-gradient-Lipschitz. In addition, $\tilde{\zeta}_k$ satisfies

$$\begin{aligned}\|E(\tilde{\zeta}_k|\tilde{x}_k)\|^2 &= \|\Sigma^{1/2}E(\zeta_k|x_k)\|^2 \le \lambda_u\eta^2 p\\ E[\|\tilde{\zeta}_k - E(\tilde{\zeta}_k|\tilde{x}_k)\|^2] &= E[\|\Sigma^{1/2}\zeta_k - E(\Sigma^{1/2}\zeta_k|x_k)\|^2] \le \lambda_u\sigma_1^2 p + \lambda_u\sigma_2^2\|\Sigma^{1/2}\tilde{x}_k\|^2,\end{aligned} \tag{S2.11}$$

Let L_t be the stochastic process defined by $dL_t = -(\Sigma^{1/2}\nabla f(\Sigma^{1/2}L_t))dt + \sqrt{2}dW_t$ with initialization $L_0 \sim \tilde{\pi}_*$ (hence $L_t \sim \tilde{\pi}_*$). Define $\Delta_2 = L_\epsilon - \tilde{x}_{t+1}$ and $\Delta_1 = L_0 - \tilde{x}_t$. Then, by the same arguments used in the proof of Proposition 2 in Dalalyan and Karagulyan (2019). we have

$$\begin{aligned}\|\Sigma^{1/2}\Delta_2\|_{L_2}^2 \le &\{\|\Sigma^{1/2}\Delta_1 - \epsilon\Sigma^{1/2}U\|_{L_2} + \|\Sigma^{1/2}W\|_{L_2} + \epsilon\|\Sigma^{1/2}E(\tilde{\zeta}_k|\tilde{x}_k)\|_{L_2}\}^2\\ &+ \epsilon^2\|\Sigma^{1/2}(\tilde{\zeta}_k - E(\tilde{\zeta}_k|\tilde{x}_k))\|_{L_2}^2,\end{aligned} \tag{S2.12}$$

where $W = \int_0^\epsilon(\nabla\tilde{f}(L_t) - \nabla\tilde{f}(L_0))dt$ and $U = \nabla\tilde{f}(\tilde{x}_k + \Delta_1) - \nabla\tilde{f}(\tilde{x}_k)$.

By Lemma 4 of Dalalyan and Karagulyan (2019),

$$\|W\|_{L_2} \le 0.5\sqrt{\epsilon^4 S^3\lambda_u^3 p} + (2/3)\sqrt{2\epsilon^3 p}S\lambda_u \le 1.65S\lambda_u(\epsilon^3 p)^{1/2}.$$

By similar arguments of Lemma 2 of Dalalyan and Karagulyan (2019), we can show that $\|\Sigma^{1/2}\Delta_1 - \epsilon\Sigma^{1/2}U\|_2 \le \rho\|\Sigma^{1/2}\Delta_1\|_2$, where $\rho = \max(1 -$

$s\lambda_l\epsilon, S\lambda_u\epsilon - 1) = 1 - s\lambda_l\epsilon$. Combining with (S2.11) and (S2.12), we have

$$\|\Sigma^{1/2}\Delta_2\|_{L_2}^2 \leq \{\rho\|\Sigma^{1/2}\Delta_1\|_{L_2} + 1.65S(\lambda_u^3\epsilon^3 p)^{1/2} + \epsilon\lambda_u\eta\sqrt{p}\}^2 + \epsilon^2\lambda_u^2\sigma_1^2 p$$
$$+ \epsilon^2\lambda_u^2\sigma_2^2 E\|x_k\|^2,$$

which further implies

$$W_2^2(\pi_{k+1}, \pi_*) \leq \{(1 - s\lambda_l\epsilon)W_2^2(\pi_{k+1}, \pi_*) + 1.65S(\lambda_u^3\epsilon^3 p)^{1/2} + \epsilon\lambda_u\eta\sqrt{p}\}^2$$
$$+ \epsilon^2\lambda_u^2\sigma_1^2 p + \epsilon^2\lambda_u^2\sigma_2^2 E\|x_k\|^2.$$

By Lemma S5, $E\|x_k\|^2 \leq (W_2(\pi_k, \pi_*) + \sqrt{\widetilde{V}})^2$, we can derive that

$$\begin{aligned}
W_2^2(\pi_{k+1}, \pi_*) &\leq \left[(1 - s\lambda_l\epsilon)W_2(\pi_k, \pi_*) + 1.65S(\lambda_u^3\epsilon^3 p)^{1/2} + \epsilon\eta\lambda_u\sqrt{p}\right]^2 \\
&\quad + \epsilon^2\sigma_1^2\lambda_u^2 p + \epsilon^2\sigma_2^2\lambda_u^2(W_2(\pi_k, \pi_*) + \sqrt{\widetilde{V}})^2 \\
&\leq \left[(1 - s\lambda_l\epsilon)W_2(\pi_k, \pi_*) + 1.65S(\lambda_u^3\epsilon^3 p)^{1/2} + \epsilon\eta\lambda_u\sqrt{p}\right]^2 \\
&\quad + \epsilon^2\sigma_1^2\lambda_u^2 p + 2\epsilon^2\sigma_2^2\lambda_u^2\widetilde{V} + 2\epsilon^2\sigma_2^2\lambda_u^2 W_2^2(\pi_k, \pi_*) \\
&\leq \left[(1 - s\lambda_l\epsilon + \sqrt{2}\sigma_2\epsilon\lambda_u)W_2(\pi_k, \pi_*) + 1.65S(\lambda_u^3\epsilon^3 p)^{1/2} + \epsilon\eta\lambda_u\sqrt{p}\right]^2 \\
&\quad + \epsilon^2\sigma_1^2\lambda_u^2 p + 2\epsilon^2\sigma_2^2\lambda_u^2\widetilde{V},
\end{aligned}$$

which concludes the proof. □

Remark S5. Consider a Langevin Monte Carlo algorithm with inaccurate gradients, varying conditioning matrices and a constant learning rate ϵ, i.e.,

$$x_{k+1} = x_k - \epsilon\Sigma_k[\nabla f(x_k) + \zeta_k] + \sqrt{2\epsilon}\xi_{k+1}; \quad \xi_{k+1} \sim N(0, \Sigma_k).$$

If Σ_k is positive definite, $\lambda_l \leq \inf_k \lambda_{\min}(\Sigma_k) \leq \sup_k \lambda_{\max}(\Sigma_k) \leq \lambda_u$ and $\epsilon \leq 2/(s\lambda_l + S\lambda_u)$, then (S2.9) holds for all iterations. Conditioned on $x_{k+1} \in \Theta$ for some measurable set Θ, it is easy to justify that

$$W_2^2(\hat{\pi}_{k+1}, \pi_*) \leq W_2^2(\pi_{k+1}, \pi_*)/Pr(x_{k+1} \in \Theta), \tag{S2.13}$$

where $\hat{\pi}_{k+1}$ is the marginal distribution of x_{k+1} conditional on $x_{k+1} \in \Theta$. Combining it with (S2.9) and Lemma 1 of Dalalyan and Karagulyan (2019), one can obtain that, if $Pr(x_{k+1} \in \Theta | x_k \in \Theta) \geq 1 - \delta$ for some sufficiently small constant δ and all k, then conditional on the event $\{x_1, \ldots, x_k \in \Theta\}$,

$$\begin{aligned} W_2(\pi_k, \pi_*) \leq & \left(\frac{1 - s\lambda_l\epsilon + \sqrt{2}\sigma_2\epsilon\lambda_u}{\sqrt{1-\delta}} \right)^k W_2(\pi_0, \pi_*) \\ & + \frac{\eta\lambda_u\sqrt{p}}{s\lambda_l - \sqrt{2}\sigma_2\lambda_u - \delta} + \frac{1.65S(\lambda_u^3\epsilon p)^{1/2}}{s\lambda_l - \sqrt{2}\sigma_2\lambda_u - \delta} + \frac{\sqrt{\epsilon}\lambda_u(\sigma_1^2 p + 2\sigma_2^2\widetilde{V})}{1.65S(\lambda_u p)^{1/2}\sqrt{1-\delta}}. \end{aligned} \tag{S2.14}$$

The next corollary provides a decaying learning-rate version of the convergence result (S2.14).

Corollary S1. *Consider a Langevin Monte Carlo algorithm*

$$x_{k+1} = x_k - \epsilon_{k+1}\Sigma_k[\nabla f(x_k) + \zeta_k] + \sqrt{2\epsilon_{k+1}}\xi_k; \quad \xi_k \sim N(0, \Sigma_k),$$

where ζ_k satisfies (S2.8) when $x_k \in \Theta$, Σ_k is positive definite,

$$\lambda_l \leq \inf_k \lambda_{\min}(\Sigma_k) \leq \sup_k \lambda_{\max}(\Sigma_k) \leq \lambda_u,$$

and the learning rate $\epsilon_k = \epsilon_0/k^{\varpi}$ for some $\epsilon_0 \leq 2/[(s\lambda_l + S\lambda_u)$ and $\varpi \in (0,1)$. Let $\mathcal{K}$ denote the total number of iterations of the algorithm. If $s\lambda_l > \sqrt{2}\sigma_2\lambda_u$ and $Pr(x_1 \ldots, x_{\mathcal{K}} \in \Theta) \geq 1 - \delta$ holds with $\delta = o(1/\mathcal{K})$, then we have that conditioned on the event $\{x_1, \ldots, x_{\mathcal{K}} \in \Theta\}$,

$$\limsup_{\mathcal{K}\to\infty} W_2(\pi_{\mathcal{K}}, \pi_*) \leq \frac{\varphi}{1-\varphi}\frac{\eta\lambda_u\sqrt{p}}{s\lambda_l - \sqrt{2}\sigma_2\lambda_u}, \text{ for some constant } \varphi \in (0,1). \tag{S2.15}$$

Proof. The proof of this corollary closely follows the proof of Theorem 2(i) in Song et al. (2020). Let $K_0 = 0$, and K_i $(i > 0)$ be the smallest integer such that $K_i^{-\varpi} \leq (1+i)^{-\chi}$, where $\chi = \varpi/(1-\varpi)$. Thus, asymptotically, we have $K_{i+1} - K_i \approx (\chi/\varpi)K_{i+1}^{\varpi}$.

Note that $Pr(x_1 \ldots, x_k \in \Theta) \geq 1 - \delta$ implies that $Pr(x_i \in \Theta | x_{i-1}\Theta) \geq 1 - \delta$ for all $i \leq k$, and $\delta \leq s\lambda_l - \sqrt{2}\sigma_2\lambda_u$ given a large k since $\delta = o(1/k)$. In the spirit of (S2.14), we have that conditional on the event $\{x_1, \dots, x_{K_{i+1}} \in \Theta\}$,

$$W_2(\pi_{K_{i+1}}, \pi_*) \le (1 - (s\lambda_l - \sqrt{2}\sigma_2\lambda_u)\epsilon_{K_{i+1}})^{K_{i+1}-K_i}(1-\delta)^{-(K_{i+1}-K_i)/2} W_2(\pi_{K_i}, \pi_*) + \frac{\eta\lambda_u\sqrt{p}}{s\lambda_l - \sqrt{2}\sigma_2\lambda_u - \delta} + \left[\frac{1.65S(\lambda_u^3 p)^{1/2}}{s\lambda_l - \sqrt{2}\sigma_2\lambda_u - \delta} + \frac{\lambda_u(\sigma_1^2 p + 2\sigma_2^2\widetilde{V})}{1.65S(\lambda_u p)^{1/2}\sqrt{1-\delta}}\right]\sqrt{\epsilon_{K_i}}.$$

Note that due to the fact that $K_{i+1} - K_i \approx (\chi/\varpi)\epsilon_{K_{i+1}}^{-1}$, we have

$$\lim_{i\to\infty}[1 - (s\lambda_l - \sqrt{2}\sigma_2\lambda_u)\epsilon_{K_{i+1}}]^{K_{i+1}-K_i} = \exp\left\{-\frac{\epsilon_0(s\lambda_l - \sqrt{2}\sigma_2\lambda_u)\chi}{\varpi}\right\} < 1$$

$$\lim_{K_{i+1}\le k, k\to\infty}[1-\delta]^{-(K_{i+1}-K_i)/2} = 1.$$

Therefore, for any positive constant $\varphi > \exp(-\frac{\epsilon_0(s\lambda_l - \sqrt{2}\sigma_2\lambda_u)\chi}{\varpi})$, there exists a constant k_m such that when $k \ge k_m$ and $K_{i+1} \le k$,

$$(1 - (s\lambda_l - \sqrt{2}\sigma_2\lambda_u)\epsilon_{K_{i+1}})^{K_{i+1}-K_i}(1-\delta)^{-(K_{i+1}-K_i)/2} \le \varphi,$$

that is,

$$\begin{aligned} W_2(\pi_{K_{i+1}}, \pi_*) \le \varphi W_2(\pi_{K_i}, \pi_*) + \frac{\eta\lambda_u\sqrt{p}}{s\lambda_l - \sqrt{2}\sigma_2\lambda_u - \delta} \\ + \left[\frac{1.65S(\lambda_u^3 p)^{1/2}}{s\lambda_l - \sqrt{2}\sigma_2\lambda_u - \delta} + \frac{\lambda_u(\sigma_1^2 p + 2\sigma_2^2\widetilde{V})}{1.65S(\lambda_u p)^{1/2}\sqrt{1-\delta}}\right]\sqrt{\epsilon_{K_i}}. \end{aligned}$$

The above recursive inequality implies that for $K_I \le k$ and $k \ge k_m$

$$
\begin{aligned}
W_2(\pi_{K_I}, \pi_*) \le \varphi^I W_2(\pi_{K_0} = \pi_0, \pi_*) + (\sum_{t=1}^{I} \varphi^{t-1}) \frac{\eta \lambda_u \sqrt{p}}{s\lambda_l - \sqrt{2}\sigma_2 \lambda_u - \delta} \\
+ (\sum_{t=1}^{I} \varphi^{t-1} K_{I-t}^{-\varpi/2}) \sqrt{\epsilon_0} \left[\frac{1.65 S(\lambda_u^3 p)^{1/2}}{s\lambda_l - \sqrt{2}\sigma_2 \lambda_u - \delta} + \frac{\lambda_u(\sigma_1^2 p + 2\sigma_2^2 \widetilde{V})}{1.65 S(\lambda_u p)^{1/2} \sqrt{1-\delta}} \right].
\end{aligned}
\tag{S2.16}
$$

As $k \to \infty$ and $I \to \infty$, $\sum_{t=1}^{I} \varphi^{t-1} K_{I-t}^{-\varpi/2} \to 0$, hence we have that $W_2(\pi_{K_{I+1}}, \pi_*) \to \frac{\varphi}{1-\varphi} \frac{\eta \lambda_u \sqrt{p}}{s\lambda_l - \sqrt{2}\sigma_2 \lambda_u}$. □

Remark S6. For technical simplicity, we require $\varpi < 1$ for the decay of the learning rate ($\epsilon_t \propto t^{-\varpi}$) in the above corollary. We conjecture that the corollary still holds under the choice $\epsilon_t \propto t^{-1}$, i.e., $\varpi = 1$. However, more subtle technical tools are necessary to rigorously characterize the convergence rate under the setting $\epsilon_t \propto t^{-1}$, see Teh et al. (2016).

Assumption S9. *The stochastic gradient can be expressed as $\tilde{\nabla} f(x, \zeta) + \zeta'$, where ζ is random variable independent of the past SGLD path and the SGLD noise e_t, and ζ' is deterministic, such that $\tilde{\nabla} f(x, \zeta)$ is an unbiased estimator of $\nabla f(x)$ and ζ' is a bounded bias. That is,*

$$
\begin{aligned}
& E\tilde{\nabla} f(x, \zeta) = \nabla f(x), \quad \|\tilde{\nabla} f(x_1, \zeta) - \tilde{\nabla} f(x_2, \zeta)\| \le L_0 \|x_1 - x_2\|, \\
& \|\tilde{\nabla} f(x_*, \zeta)\| \le M, \quad \|\zeta'\| \le B,
\end{aligned}
\tag{S2.17}
$$

where x_* is the optimum of density f.

Remark S7. In Assumption S9, ζ usually stands for a simple random subsample used in stochastic gradient evaluation. Therefore, Assumption S9 requires that for any fixed subsample, the stochastic gradient function is L_0-Lipschitz.

Lemma S7. *Assume that conditions S9 for the stochastic gradient and s-strongly convexity for the target distribution. For any constant $\kappa > 1$, $\delta > 0$, let*

$$R \geq \max\{\sqrt{400p\log(2\mathcal{K}/\delta)/s\log\kappa}, 2\sqrt{(4M^2+2B^2+2p)/s}, 8B/s\},$$

where $\mathcal{K}$ is the number of SGLD iterations, and let $\epsilon_k = \epsilon_0/k^{\varpi}$ with $\varpi \in [0,1)$. If $\|x_0 - x_*\| \leq R$ with at least $(1-\delta/2)$ probability and the learning rate $\epsilon_0 \leq \min((2-\sqrt{2})^2 p(L_0\kappa R+B+M)^{-2}, 1, s/(8L_0^2))$, then $\max_{i\leq T}\|x_i - x_*\| \leq \kappa R$ holds with at least $(1-\delta)$ probability.

Proof. The proof strictly follows Lemma 6.1 of Zou et al. (2021). First, denote by e_k an independent normal variable with the identity covariance matrix, we then have

$$\begin{aligned}
E[\|x_{k+1}-x_*\|_2^2|x_k] &= E[\|x_k - x_* - \epsilon_t(\tilde{\nabla} f(x_t,\zeta)+\zeta') + \sqrt{2\epsilon_t}e_{k+1}\|_2^2|x_k] \\
=&\|x_k - x_*\|^2 - 2\epsilon_t E[\langle x_k - x_*, \tilde{\nabla} f(x_t,\zeta)\rangle|x_k] - 2\epsilon_t E[\langle x_k - x_*, \zeta'\rangle|x_k] \\
&+ \epsilon_t^2 E[\|\tilde{\nabla} f(x_t,\zeta)+\zeta'\|_2^2|x_t] + 2p\epsilon_t \\
\leq&\|x_k - x_*\|^2 - 2\epsilon_t s\|x_k - x_*\|^2 + 2\epsilon_t B\|x_k - x_*\| + 4\epsilon_t^2 L_0^2\|x_t - x_*\|_2^2 \\
&+ 4\epsilon_t^2 M^2 + 2\epsilon_t^2 B^2 + 2p\epsilon_t \\
=&(1-2s\epsilon_t + 4L_0^2\epsilon_t^2)\|x_k - x_*\|^2 + 2\epsilon_t B\|x_k - x_*\| + 4\epsilon_t^2 M^2 + 2\epsilon_t^2 B^2 + 2p\epsilon_t.
\end{aligned} \tag{S2.18}$$

The above result implies that $E[\|x_{k+1}-x_*\|_2^2|x_k] \leq (1-s\epsilon_t)\|x_k-x_*\|_2^2$, since $\epsilon_t \leq \min(1, s/(8L_0^2))$ and $\|x_t - x_*\|^2 \geq \max((16M^2+8B^2+8p)/s, 64B^2/s^2)$.

The concavity of the log-function implies that for any $\|x_k - x_*\|_2 \geq R$,

$$E[\log(\|x_{k+1}-x_*\|_2^2)|x_k] \leq \log(E[\|x_{k+1}-x_*\|_2^2|x_k]) \leq -s\epsilon_t + \log(\|x_k - x_*\|_2^2). \tag{S2.19}$$

On the other hand, $\|x_{k+1}-x_*\|_2 - \|x_k - x_*\|_2 \leq \epsilon_t\|\tilde{\nabla} f(x,\zeta)\| + \epsilon_t\|\zeta'\| + \sqrt{2\epsilon_t}\|e_{k+1}\|_2$ and $\|e_{k+1}\|$ has a sub-Gaussian distribution satisfying that $P(\|e_{k+1}\|_2 \geq \sqrt{p}+\sqrt{2}z) \leq e^{-z^2}$ for $z \geq 0$. If $\|x_k\|_2 \leq \kappa R$, then $\|\tilde{\nabla} f(x,\zeta)\| \leq L_0\kappa R + M$. If we further assume that $\epsilon_t \leq (2-\sqrt{2})^2 p(L_0\kappa R + B + M)^{-2}$, then for any $z > 0$,

$$P(\|x_{k+1} - x_*\|_2 - \|x_k - x_*\|_2 \geq 2\sqrt{\epsilon_t p} + 2\sqrt{\epsilon_t} z) \leq e^{-z^2}.$$

If further $R \leq \|x_k\|_2$ holds, then

$$\log(\|x_{k+1} - x_*\|_2^2) - \log(\|x_k - x_*\|_2^2) \leq \frac{2\|x_{k+1} - x_*\|_2 - 2\|x_k - x_*\|_2}{R}.$$

Therefore, if $R \leq \|x_k\|_2 \leq \kappa R$,

$$P(\log(\|x_{k+1} - x_*\|_2^2) - \log(\|x_k - x_*\|_2^2) \geq 4\sqrt{\epsilon_0 p} R^{-1} + 4t\sqrt{\epsilon_0} R^{-1}) \leq e^{-t^2}. \tag{S2.20}$$

With slight modifications to the proof of Fact 1 in Lemma 6.1 of Zou et al. (2021) and Theorem 2 of Shamir (2011) (such that they apply to varying step size SGLD), we have that $[\log(\|x_k\|^2) + s\sum_{i=1}^{k} \epsilon_i]$'s have subgaussian martingale difference. Further, given $800p\log(2\mathcal{K}/\delta)/(sR^2) \leq 2\log\kappa$ for some C'_{ϖ}, then conditioned on the event $\|x_0 - x_*\| \leq R$, $\|x_i - x_*\| \leq \kappa R$ for all $1 \leq i \leq \mathcal{K}$ with probability at least $1 - \delta/2$. This concludes the proof. □

Remark S8. The result of Lemma S7 can be easily generalized to preconditioned SGLD, i.e., (S2.4), via the transformation (S2.10). If the eigenvalues of the preconditioning matrix Σ_k are bounded from above and from below by λ_u and λ_l, respectively, then Lemma S7 holds with $\epsilon_0 \leq \min((2-\sqrt{2})^2 p(L_0\kappa R + B + M)^{-2}\lambda_u^{-2}, 1, \lambda_l s/(8\lambda_u^2 L_0^2))$ and

$$R \geq \sqrt{\lambda_u} \max\{\sqrt{400p\log(2\mathcal{K}/\delta)/(\lambda_l s \log\kappa)}, 2\sqrt{(4\lambda_u^2 M^2 + 2\lambda_u^2 B^2 + 2p)/(\lambda_l s)},$$
$$8\lambda_u B/\lambda_l s\}.$$

Proof of Theorem 2

Proof. Define K_i as in the proof of Corollary S1, and we let $\mathcal{K} = K_{\varkappa}(\asymp \varkappa^{\chi/\varpi})$ for some $\varkappa$ where $\chi = \varpi/(1-\varpi)$.

At stage $t = 1$, Algorithm 4 performs exactly as Algorithm 2; that is, it is a Langevin Monte Carlo algorithm with a varying conditioning matrix as discussed in Section 3. By Corollary S1 with no bias $\eta = 0$, $\Theta = \mathbb{R}^p$ and $\delta = 0$, we obtain that there exists some $\varphi_1 > \exp(-\epsilon_{1,0}(s\lambda_l - \sqrt{2}\sigma_2\lambda_u)\chi/\varpi)$, such that

$$\begin{aligned} W_2(\tilde{\pi}_1, \pi_1) \leq{}& \varphi_1^{\varkappa} W_2(\nu_1, \pi_1) \\ &+ (\sum_{j=1}^{\varkappa} \varphi_1^{j-1} K_{\varkappa-j}^{-\frac{\varpi}{2}})\sqrt{\epsilon_{1,0}} \left[\frac{1.65 S_1 \lambda_{1,u}\sqrt{p\lambda_{1,u}}}{s_1\lambda_{1,l} - \sqrt{2}\sigma_{1,s}\lambda_{1,u}} + \frac{\sigma_{1,s}^2 \lambda_{1,u}(p + 2\widetilde{V}_1)}{1.65 S_1 \sqrt{\lambda_{1,u} p}}\right], \end{aligned} \tag{S2.21}$$

where the first term is of order $O(\varphi_1^{\varkappa}) \asymp \varphi_1^{\mathcal{K}^{\varpi/\chi}}$ and the second term is of order $O(\varkappa^{-\chi/2}) \asymp \mathcal{K}^{-\varpi/2}$ with respect to (w.r.t.) the iteration number $\mathcal{K}$. Note that $\log(\varphi_1^{\varkappa}) = \mathcal{K}^{\varpi/\chi} \log \varphi_1 \asymp -\mathcal{K}^{\varpi/\chi}/\log \mathcal{K}$, therefore, w.r.t. $\mathcal{K}$, $W_2(\tilde{\pi}_1, \pi_1)$ decreases polynomially.

Now, we study $W_2(\tilde{\pi}_{t+1}, \pi_{t+1})$ for $t \geq 1$. As discussed in Section 3.1, at stage $t+1$, the algorithm can be rewritten as

$$\begin{aligned} x_{t+1,k+1}^{a,i} &= x_{t+1,k}^{a,i} + \epsilon_{t+1,k+1}\Sigma_{t+1,k+1}\big[\frac{N}{n} H_{t+1,k}^T V_{t+1}^{-1}(y_{t+1,k} - H_{t+1,k} x_{t+1,k}^{a,i}) \\ &\quad + \nabla \log \pi(x_{t+1,k}^{a,i} | \tilde{x}_{t,k}^i)\big] + e_{t+1} \\ &\triangleq x_{t+1,k}^{a,i} + \epsilon_{t+1,k+1}\Sigma_{t+1,k+1}\left[(I) + (II)\right] + e_{t+1}, \end{aligned} \tag{S2.22}$$

where $e_{t+1} \sim N(0, 2\epsilon_{t+1,k+1}\Sigma_{t+1,k+1})$, and $\tilde{x}_{t,k}^i$ denotes a sample drawn from the set $\mathcal{X}_t$ according to an importance weight proportional to $\pi(x_{t+1,k}^{a,i}|\tilde{x}_{t,k}^i)$.

We first study the bias of the gradient estimate used in (S2.22). Note that the term (I) is unbiased due to the property of simple random sampling. To study the bias of term (II), we define $\pi(z|x_{t+1,k}^{a,i}, y_{1:t}) \propto \pi(x_{t+1,k}^{a,i}|z)\pi_t(z|y_{1:t})$; that is, $\pi(z|x_{t+1}^{a,i}, y_{1:t})$ can be viewed as a posterior density obtained with the prior density $\pi_t(z|y_{1:t})$ and the likelihood $\pi(x_{t+1}^{a,i}|z)$. Similarly, we define $\tilde{\pi}(z|x_{t+1,k}^{a,i}, y_{1:t}) \propto \pi(x_{t+1,k}^{a,i}|z)\tilde{\pi}_t(z|y_{1:t})$. Then, by equation (3.7) of the main text, the bias of term (II) can be bounded by

$$\begin{aligned}
&\left\|\int \nabla \log \pi(x_{t+1,k}^{a,i}|z)[d\tilde{\pi}(z|x_{t+1,k}^{a,i}, y_{1:t}) - d\pi(z|x_{t+1,k}^{a,i}, y_{1:t})]\right\| \\
=&\left\|\int U_t^{-1}[x_{t+1,k}^{a,i} - g(z)][d\tilde{\pi}(z|x_{t+1,k}^{a,i}, y_{1:t}) - d\pi(z|x_{t+1,k}^{a,i}, y_{1:t})]\right\| \\
=&\left\|-\int U_t^{-1}g(z)[d\tilde{\pi}(z|x_{t+1,k}^{a,i}, y_{1:t}) - d\pi(z|x_{t+1,k}^{a,i}, y_{1:t})]\right\| \\
\leq& 2M_g/\lambda_{t,s},
\end{aligned} \tag{S2.23}$$

which holds for any $\tilde{\pi}(\cdot|\cdot)$. In other words, the bias of term (II) is uniformly bounded, i.e., the bound bias requirement of Assumption S9 holds. For the rest of Assumption S9, similar to the discussion in Remark S7, ζ represents a minibatch of the data and an importance particle from the pool of particles collected at the proceeding stage. Thanks to Assumptions S4 and S6, Assumption S9 holds with $L_0 \propto N_t$, and thus Lemma S7 applies.

Without loss of generality, we assume that the global optimum of π_t, $t = 1, 2, \ldots, T$, denoted by x_t^*, is bounded by M_f, i.e., $\|x_t^*\| \leq M_f$. Let $\delta = \mathcal{K}^{-2}$. Without loss of generality, we assume that the initialization of each stage (for $t > 1$) is bounded by $M_f + c_g M_g$ for some constant $c_g > 0$. For simplicity of notation, we simply set $c_g = 1$ in the proof. In simulations, this can be ensured by setting $x_{t,0}^{a,i} = g(x_{t-1,\mathcal{K}}^{a,i}) + u_t^{a,i}$ and restricting $u_t^{a,i}$ to an appropriate range.

By Lemma S7 and remark S8, we define

$$R_t = C_0 \kappa \sqrt{p \log(2\mathcal{K}/\delta)/s_t \log \kappa},$$

for some constant C_0. Let $\epsilon_{t+1,0} \asymp (n_{t+1}^2 \log \mathcal{K})^{-1}$ which satisfies the stepsize assumption in Remark S8. Then we have that all $\mathcal{K}$ iterations of the SGLD path in the $(t+1)$-th stage are bounded within a compact set $\Theta_{t+1} = \{x : \|x - x_{t+1}^*\| \leq \kappa R_{t+1}\}$ with probability $1-\delta$ for a sufficiently large $\mathcal{K}$.

On the other hand, when $\|x_{t+1,k}^{a,i} - x_{t+1}^*\| \leq \kappa R_{t+1}$ for all k, we can further refine the bound for the bias of term II. Define $c_t := \int \pi(x_{t+1,k}^{a,i}|z)\pi_t(z|y_{1:t})dz$, $\tilde{c}_t := \int \pi(x_{t+1,k}^{a,i}|z)\tilde{\pi}_t(z|y_{1:t})dz$, $a_t := \int \nabla_x \pi(x_{t+1,k}^{a,i}|z)\pi_t(z|y_{1:t})dz$, $\tilde{a}_t := \int \nabla_x \pi(x_{t+1,k}^{a,i}|z)\tilde{\pi}_t(z|y_{1:t})dz$, $\delta_{c_t} := \tilde{c}_t - c_t$, and $\delta_{a_t} := \tilde{a}_t - a_t$. Therefore, the bias of the term (II) can be re-expressed as

$$\left|\frac{a_t}{c_t} - \frac{\tilde{a}_t}{\tilde{c}_t}\right| \leq \left|\frac{a_t \delta_{c_t}}{c_t \tilde{c}_t}\right| + \left|\frac{\delta_{a_t}}{c_t}\right|.$$

Note that $\|a_t\|$ is uniformly bounded by $(2\pi)^{-p/2}(\lambda_{t+1,s})^{-(p+1)/2}C$ for some constant C, due to Assumption S6 and the fact

$$\nabla_x \pi(x_{t+1,k}^{a,i}|z) = (2\pi)^{-p/2}(det(U_{t+1}))^{-1/2} U_{t+1}^{-1}(x - g(z)) \\ \times \exp\{-(x-g(z))^T U_{t+1}^{-1}(x-g(z))/2\},$$

by Lemma S1, we have that $|\delta_{c_t}| \leq [(2\pi)^{-p/2}(\lambda_{t+1,s})^{-(p+1)/2}Cl]W_2(\pi_t, \tilde{\pi}_t)$ and $\|\delta_{a_t}\| \leq [(2\pi)^{-p/2}(\lambda_{t+1,s})^{-(p+2)/2}C'l]W_2(\pi_t, \tilde{\pi}_t)$.

With probability $1-\delta$, $x \in \{x : \|x - x_t^*\| \leq \kappa R_0\}$ which implies that $\|x\| \leq \kappa R_0 + M_f$, and

$$\begin{aligned} c_t \geq & \min_{\|x\| \leq R_0 + M_f, z} \frac{1}{(2\pi)^{p/2}(det(U_{t+1}))^{1/2}} \exp\{-(x-g(z))^T U_{t+1}^{-1}(x-g(z)/2) \\ \geq & (2\pi)^{-p/2}(\lambda'_{t+1,s})^{-p/2} \min_{\|x\| \leq \kappa R_0 + M_f, z} \exp\{-\|g(z)\|^2/\lambda_{t+1,s}\} \exp\{-\|x\|^2/\lambda_{t+1,s}\} \\ = & (2\pi)^{-p/2}(\lambda'_{t+1,s})^{-p/2} \exp\{-M_g^2/\lambda_{t+1,s}\} \exp\{-(\kappa R_0 + M_f)^2/\lambda_{t+1,s}\}. \end{aligned} \tag{S2.24}$$

The same bound applies to $\tilde{c}_t$ as well.

Combining all the above derivations together, we obtain the bound when $x_{t+1,k}^{a,i} \in \Theta_{t+1}$,

$$Bias = Bias_{II} \leq C_1^p C_2' l (T/\delta)^{C_3 p/s_{t+1}} W_2(\pi_t, \tilde{\pi}_t), \tag{S2.25}$$

for some constants C_1, C_2 and C_3 that depend on constants $\kappa, \lambda_{t,s}, \lambda'_{t,s}, M_f$ and $c_g M_g$.

By Condition (A.4), the variance of term (I) is bounded by $\sigma_{t+1,s}^2(p +$

$\|x\|^2)$. The variance of term (II) is upper bounded by

$$\begin{aligned}
&E\left\|\nabla \log \pi(x^{a,i}_{t+1,k}|\tilde{x}^i_{t,k}) - E\left(\nabla \log \pi(x^{a,i}_{t+1,k}|\tilde{x}^i_{t,k})\right)\right\|^2 \\
&\leq (l/\lambda_{t,s})^2 E\|\tilde{x}^i_{t,k} - E(\tilde{x}^i_{t,k})\|^2 \quad \text{(by Lemma S3)} \\
&\leq (l/\lambda_{t,s})^2 \left[W_2(\pi_t, \tilde{\pi}_t)^2 + p\sigma^2_{t,v} + 2W_2(\pi_t, \tilde{\pi}_t)\sqrt{p\sigma^2_{t,v}}\right],
\end{aligned} \tag{S2.26}$$

by Lemma S4 and (S2.7). Combining the above results together, the variance of the estimated gradient is upper bounded by

$$\begin{aligned}
&2\sigma^2_{t+1,s}p + 2(l/\lambda_{t,s})^2(W_2(\pi_t, \tilde{\pi}_t) + \sqrt{p\sigma^2_{t,v}})^2 + 2\sigma^2_{t+1,s}\|x^{a,i}_{t+1,k}\|^2 \\
&:= \sigma^2_p + 2\sigma^2_{t+1,s}\|x^{a,i}_{t+1,k}\|^2.
\end{aligned} \tag{S2.27}$$

As implied by (S2.25) and (S2.27), a smaller value of $W_2(\pi_t, \tilde{\pi}_t)$ will help to reduce the variance and bias of the stochastic gradient at stage $t+1$.

Recall that $\tilde{\nu}_{t+1}$ denotes the practical state initial distribution of $x^{a,i}_{t+1,0}$. Applying equation (S2.16) in Corollary S1, we obtain that conditioned on the event $\{x_{t+1,0}, \ldots, x_{t+1,\mathcal{K}} \in \Theta_{t+1}\}$ whose probability is $1 - \mathcal{K}^{-2}$, there exists some $\varphi_{t+1} \in (0, 1)$ such that

$$\begin{aligned}
&W_2(\tilde{\pi}_{t+1}, \pi_{t+1}) \\
\leq &\varphi_{t+1}^{\varkappa} W_2(\tilde{\nu}_{t+1}, \pi_{t+1}) + \frac{\varphi_{t+1}}{1-\varphi_{t+1}} \frac{\lambda_{t+1,u} C_1^p C_2' l(\mathcal{K}/\delta)^{C_3 p/s_{t+1}}}{s_{t+1}\lambda_{t+1,l} - 2\sigma_{t+1,s}\lambda_{t+1,u} - \delta} W_2(\pi_t, \tilde{\pi}_t) \\
+ &(\sum_{j=1}^{\varkappa} \varphi_{t+1}^{j-1} K_{\varkappa-j}^{-\varpi/2}) \sqrt{\epsilon_{t,0}} \left[\frac{1.65 S_{t+1} \sqrt{p\lambda_{t+1,u}^3}}{s_{t+1}\lambda_{t+1,l} - 2\sigma_{t+1,s}\lambda_{t+1,u} - \delta} + \frac{\sqrt{\lambda_{t+1,u}}(\sigma_p^2 + 4\sigma_{t+1,s}^2 \widetilde{V}_t)}{1.65 S_{t+1}\sqrt{p(1-\delta)}} \right] \\
\leq &\varphi_{t+1}^{\varkappa} l W_2(\tilde{\pi}_t, \pi_t) + \varphi_{t+1}^{\varkappa} M + \frac{\varphi_{t+1}}{1-\varphi_{t+1}} \frac{\lambda_{t+1,u} C_1^p C_2' l(\mathcal{K}/\delta)^{C_3 p/s_{t+1}}}{s_{t+1}\lambda_{t+1,l} - 2\sigma_{t+1,s}\lambda_{t+1,u} - \delta} W_2(\pi_t, \tilde{\pi}_t) \\
+ &(\sum_{j=1}^{\varkappa} \varphi_{t+1}^{j-1} K_{\varkappa-j}^{-\varpi/2}) \sqrt{\epsilon_{t,0}} \left[\frac{1.65 S_{t+1} \sqrt{p\lambda_{t+1,u}^3}}{s_{t+1}\lambda_{t+1,l} - 2\sigma_{t+1,s}\lambda_{t+1,u} - \delta} + \frac{\sqrt{\lambda_{t+1,u}}(\sigma_p^2 + 4\sigma_{t+1,s}^2 \widetilde{V}_t)}{1.65 S_{t+1}\sqrt{p(1-\delta)}} \right],
\end{aligned} \tag{S2.28}$$

where we use the fact that $W_2(\tilde{\nu}_{t+1}, \pi_{t+1}) \leq M + lW_2(\tilde{\pi}_t, \pi_t)$ (due to Assumption (A.6) and Lemma S2).

As shown by (S2.21) and the discussion below (S2.21), $W_2(\tilde{\pi}_1, \pi_1)$ decreases polynomially w.r.t. $\mathcal{K}$. Recursively, when p/s_{t+1} is small enough, all terms in the RHS of (S2.28) decreases polynomially w.r.t. $\mathcal{K}$. Therefore, trivially by mathematical induction, we have that conditioned on the event $\{x_{t,0}, \ldots, x_{t,\mathcal{K}} \in \Theta_t \forall t \leq T\}$, $W_2(\tilde{\pi}_T, \pi_T) = o(1)$ as $\mathcal{K}$ goes to infinity. Since this event holds with probability $1 - T/\mathcal{K}^2$, we interpret the convergence result as, with dominating probability, $x_{T,\mathcal{K}}^{a,i}$ follows a probability law π_T' and $\lim_{\mathcal{K}\to\infty} W_2(\pi_T', \pi_T) = 0$. □

Bibliography

Brascamp, H. J. and E. H. Lieb (2002). On extensions of the Brunn-Minkowski and Prékopa-Leindler theorems, including inequalities for log concave functions, and with an application to the diffusion equation. In *Inequalities*, pp. 441–464. Springer.

Cheng, X. and P. L. Bartlett (2018). Convergence of Langevin MCMC in KL-divergence. *Proceedings of Machine Learning Research 83*, 186–211.

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

Ding, N., Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven (2014). Bayesian sampling using stochastic gradient thermostats. In *NIPS*.

Durmus, A. and E. Moulines (2017). Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability 27*(3), 1551–1587.

Graves, A., M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber (2009, May). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*(5), 855–868.

Graves, A., A. rahman Mohamed, and G. E. Hinton (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649.

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural computation 9*, 1735–80.

Kantas, N., A. Doucet, S. Singh, and J. Maciejowski (2009). An overview of sequential monte carlo methods for parameter estimation in general state-space models. *IFAC Proceedings Volumes 42*(10), 774–785. 15th IFAC Symposium on System Identification.

Li, C., C. Chen, D. Carlson, and L. Carin (2016). Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 1788–1794. AAAI Press.

Saumard, A. and J. Wellner (2014). Log-concavity and strong log-concavity: a review. *Statistics Surveys 8*, 45.

Shamir, O. (2011). A variant of Azuma's inequality for martingales with subgaussian tails. *arXiv:1110.2392*.

Song, Q., Y. Sun, M. Ye, and F. Liang (2020). Extended stochastic gradient MCMC algorithms for large-scale Bayesian variable selection. *Biometrika 107*, 997–1004.

Teh, W., A. Thiery, and S. Vollmer (2016). Consistency and fluctuations for stochastic gradient Langevin dynamics. *Journal of Machine Learning Research 17*, 1–33.

Welling, M. and Y. W. Teh (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *ICML*.

Zou, D., P. Xu, and Q. Gu (2021). Faster convergence of stochastic gradient Langevin dynamics for non-log-concave sampling. In *Uncertainty in Artificial Intelligence*, pp. 1152–1162. PMLR.