Supplementary Material of High-Order Stochastic Gradient Thermostats for Bayesian Learning of Deep Models

Chunyuan Li[1], Changyou Chen[1], Kai Fan[2] and Lawrence Carin[1]
[1]Department of Electrical and Computer Engineering, Duke University
[2]Computational Biology and Bioinformatics, Duke University
chunyuan.li@duke.edu, cchangyou@gmail.com, kai.fan@duke.edu, lcarin@duke.edu

A The proof of Lemma 1

Proof In mSGNHT, $\mathbf{X}_t = (\boldsymbol{\theta}_t, \boldsymbol{p}_t, \boldsymbol{\xi}_t)$. The update equations with a Euler integrator are:

$$\begin{cases} \boldsymbol{\theta}_{t+1} & = \boldsymbol{\theta}_t + \boldsymbol{p}_t h \\ \boldsymbol{p}_{t+1} & = \boldsymbol{p}_t - \nabla_{\boldsymbol{\theta}} \tilde{U}_t(\boldsymbol{\theta}_{t+1}) h - \text{diag}(\boldsymbol{\xi}_t) \boldsymbol{p}_t h + \sqrt{2D} \boldsymbol{\zeta}_{t+1} \\ \boldsymbol{\xi}_{t+1} & = \boldsymbol{\xi}_t + (\boldsymbol{p}_{t+1} \odot \boldsymbol{p}_{t+1} - 1)\, h \end{cases}$$

Based on the update equations, it is easily seen that the corresponding Kolmogorov operator $\tilde{P}_h^l$ for mSGNHT is

$$\tilde{P}_h^l = e^{h\mathcal{L}_1} \circ e^{h\mathcal{L}_2} \circ e^{h\mathcal{L}_3} \ , \tag{1}$$

where $\mathcal{L}_1 \triangleq (\boldsymbol{p} \odot \boldsymbol{p} - 1) \cdot \nabla_{\boldsymbol{\xi}}$, $\mathcal{L}_2 \triangleq -\text{diag}(\boldsymbol{\xi}) \boldsymbol{p}_t \cdot \nabla_{\boldsymbol{p}} - \nabla_{\boldsymbol{\theta}} \tilde{U}_l(\boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{p}} + 2DI : \nabla_{\boldsymbol{p}} \nabla_{\boldsymbol{p}}^T$, and $\mathcal{L}_3 \triangleq \boldsymbol{p} \cdot \nabla_{\boldsymbol{\theta}}$. Using the BCH formula, we have

$$\tilde{P}_h^l = e^{h(\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3)} + O(h^2) \ . \tag{2}$$

On the other hand, the local generator of mSGNHT at the t-th iteration can be seen to be:

$$\tilde{\mathcal{L}}_t = \mathcal{L}_1 + \tilde{\mathcal{L}}_2 + \mathcal{L}_3 \ , \tag{3}$$

where $\mathcal{L}_1$ and $\mathcal{L}_3$ are defined previously, and $\tilde{\mathcal{L}}_2 = -\text{diag}(\boldsymbol{\xi}) \boldsymbol{p} \cdot \nabla_{\boldsymbol{p}} - \nabla_{\boldsymbol{\theta}} \tilde{U}_l(\boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{p}} + 2DI : \nabla_{\boldsymbol{p}} \nabla_{\boldsymbol{p}}^T$. According to the Kolmogorov's backward equation, we have

$$\mathbb{E}[f(\mathbf{X}_{t+1})] = e^{h\tilde{\mathcal{L}}_t} f(\mathbf{X}_t) \ . \tag{4}$$

Substitute (4) into (2) and use the fact that $\boldsymbol{p} = \boldsymbol{p}_t + O(h)$ by Taylor expansion, it is easily seen

$$\tilde{P}_h^l = e^{h\tilde{\mathcal{L}}_t + O(h^2)} + O(h^2) = e^{h\tilde{\mathcal{L}}_t} + O(h^2) \ .$$

This completes the proof.

B The proof of Lemma 2

Proof In symmetric splitting scheme for mSGNHT, according to the splitting in (4) in the main text, the generator $\tilde{\mathcal{L}}_t$ is split into the following sub-generators which can be solved analytically: $\tilde{\mathcal{L}}_l = \mathcal{L}_A + \mathcal{L}_B + \mathcal{L}_{O_l}$, where

$$\begin{aligned} &\mathcal{A} \triangleq \mathcal{L}_A = \boldsymbol{p} \cdot \nabla_{\boldsymbol{\theta}} + (\boldsymbol{p} \odot \boldsymbol{p} - 1) \cdot \nabla_{\boldsymbol{\xi}}, \\ &\mathcal{B} \triangleq \mathcal{L}_B = -\text{diag}(\boldsymbol{\xi}) \boldsymbol{p}_t \cdot \nabla_{\boldsymbol{p}}, \\ &\mathcal{O}_l \triangleq \mathcal{L}_{O_l} = -\nabla_{\boldsymbol{\theta}} \tilde{U}_l(\boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{p}} + 2D : \nabla_{\boldsymbol{p}} \nabla_{\boldsymbol{p}}^T \ . \end{aligned}$$

The corresponding Kolmogorov operator $\tilde{P}_h^l$ for the splitting integrator can be seen to be:

$$\tilde{P}_h^l \triangleq e^{\frac{h}{2}\mathcal{L}_A} \circ e^{\frac{h}{2}\mathcal{L}_B} \circ e^{h\mathcal{L}_{O_l}} \circ e^{\frac{h}{2}\mathcal{L}_B} \circ e^{\frac{h}{2}\mathcal{L}_A},$$

In the following we use the Baker–Campbell–Hausdorff (BCH) formula (Rossmann 2002) to show that $\tilde{P}_h^l$ is a 2nd-order integrator. Specifically,

$$e^{\frac{h}{2}\mathcal{A}} e^{\frac{h}{2}\mathcal{B}} = e^{\frac{h}{2}\mathcal{A} + \frac{h}{2}\mathcal{B} + \frac{h^2}{8}[\mathcal{A},\mathcal{B}] + \frac{1}{96}([\mathcal{A},[\mathcal{A},\mathcal{B}]] + [\mathcal{B},[\mathcal{B},\mathcal{A}]]) + \cdots} \tag{5}$$

$$= e^{\frac{h}{2}\mathcal{A} + \frac{h}{2}\mathcal{B} + \frac{h^2}{8}[\mathcal{A},\mathcal{B}]} + O(h^3) \ , \tag{6}$$

where $[X, Y] \triangleq XY - YX$ is the commutator of X and Y, (5) follows from the BCH formula, and (6) follows by moving high order terms $O(h^3)$ out of the exponential map using Taylor expansion. Similarly, for the other composition, we have

$$\begin{aligned} &e^{h\mathcal{O}_l} e^{\frac{h}{2}\mathcal{A}} e^{\frac{h}{2}\mathcal{B}} = e^{h\mathcal{O}_l} \left(e^{\frac{h}{2}\mathcal{A} + \frac{h}{2}\mathcal{B} + \frac{h^2}{8}[\mathcal{A},\mathcal{B}]} + O(h^3) \right) \\ =& e^{h\mathcal{O}_l + \frac{h}{2}\mathcal{A} + \frac{h}{2}\mathcal{B} + \frac{h^2}{8}[\mathcal{A},\mathcal{B}] + \frac{1}{2}[h\mathcal{O}_l, \frac{h}{2}\mathcal{A} + \frac{h}{2}\mathcal{B} + \frac{h^2}{8}[\mathcal{A},\mathcal{B}]]} + O(h^3) \\ =& e^{h\mathcal{O}_l + \frac{h}{2}\mathcal{A} + \frac{h}{2}\mathcal{B} + \frac{h^2}{8}[\mathcal{A},\mathcal{B}] + \frac{h^2}{4}[\mathcal{O}_l,\mathcal{A}] + \frac{h^2}{4}[\mathcal{O}_l,\mathcal{B}]} + O(h^3) \\ &e^{\frac{h}{2}\mathcal{A}} e^{h\mathcal{O}_l} e^{\frac{h}{2}\mathcal{A}} e^{\frac{h}{2}\mathcal{B}} \\ =& e^{\frac{h}{2}\mathcal{A}} \left(e^{h\mathcal{O}_l + \frac{h}{2}\mathcal{A} + \frac{h}{2}\mathcal{B} + \frac{h^2}{8}[\mathcal{A},\mathcal{B}] + \frac{h^2}{4}[\mathcal{O}_l,\mathcal{A}] + \frac{h^2}{4}[\mathcal{O}_l,\mathcal{B}]} + O(h^3) \right) \\ =& e^{h\mathcal{O}_l + h\mathcal{A} + \frac{h}{2}\mathcal{B} + \frac{h^2}{4}[\mathcal{A},\mathcal{B}] + \frac{h^2}{2}[\mathcal{O}_l,\mathcal{B}]} + O(h^3) \end{aligned}$$

As a result

$$\begin{aligned} &\tilde{P}_h^l \triangleq e^{\frac{h}{2}\mathcal{B}} e^{\frac{h}{2}\mathcal{A}} e^{h\mathcal{Z}} e^{\frac{h}{2}\mathcal{A}} e^{\frac{h}{2}\mathcal{B}} \\ =& e^{\frac{h}{2}\mathcal{B}} \left(e^{h\mathcal{O}_l + h\mathcal{A} + \frac{h}{2}\mathcal{B} + \frac{h^2}{4}[\mathcal{A},\mathcal{B}] + \frac{h^2}{2}[\mathcal{O}_l,\mathcal{B}]} + O(h^3) \right) \\ =& e^{h\mathcal{O}_l + h\mathcal{A} + h\mathcal{B} + \frac{h^2}{4}[\mathcal{A},\mathcal{B}] + \frac{h^2}{2}[\mathcal{O}_l,\mathcal{B}] + \frac{h^2}{4}[\mathcal{B},\mathcal{A}] + \frac{h^2}{4}[\mathcal{B},\mathcal{O}_l] + \frac{h^2}{8}[\mathcal{B},\mathcal{B}]} + O(h^3) \\ =& e^{h(\mathcal{B} + \mathcal{A} + \mathcal{O}_l)} + O(h^3) \\ =& e^{h(\mathcal{L} + \Delta V_l)} + O(h^3) = e^{h\tilde{\mathcal{L}}_l} + O(h^3) \ . \end{aligned}$$

This completes the proof.

C More details on Lemma 3

Our justification of the symmetric splitting integrator is based on Lemma 3, which is a simplification of the main theorems in (Chen, Ding, and Carin 2015). For completeness, we give details of their main theorems in this section.

To recap notation, for an Itó diffusion with an invariant measure $\rho(\mathbf{X})$, the posterior average is defined as: $\bar{\phi} \triangleq \int_{\mathcal{X}} \phi(\mathbf{X})\rho(\mathbf{X})\mathrm{d}x$ for some test function $\phi(\mathbf{X})$ of interest. Given samples $(x_t)_{t=1}^{T}$ from a SG-MCMC, we use the *sample average* $\hat{\phi} \triangleq \frac{1}{T}\sum_{t=1}^{T}\phi(x_t)$ to approximate $\bar{\phi}$.

In addition, we define an operator for the t-th iteration as:

$$\Delta V_t \triangleq (\nabla_\theta \tilde{U}_t - \nabla_\theta U) \cdot \nabla_{\boldsymbol{p}} \,. \tag{7}$$

Theorem 1 and Theorem 2 summarize the convergence of a SG-MCMC algorithm with a K-th order integrator with respect to the *Bias* and *MSE*, under certain assumptions. Please refer to (Chen, Ding, and Carin 2015) for detailed proofs.

Theorem 1 *Let $\|\cdot\|$ be the operator norm. The bias of an SG-MCMC with a Kth-order integrator at time $\mathcal{T} = hT$ can be bounded as:*

$$\left|\mathbb{E}\hat{\phi} - \bar{\phi}\right| = O\left(\frac{1}{Th} + \frac{\sum_t \|\mathbb{E}\Delta V_t\|}{T} + h^K\right) \,.$$

Theorem 2 *For a smooth test function ϕ, the MSE of an SG-MCMC with a Kth-order integrator at time $\mathcal{T} = hT$ is bounded, for some $C > 0$ independent of (T, h), as*

$$\mathbb{E}\left(\hat{\phi} - \bar{\phi}\right)^2 \leq C\left(\frac{\frac{1}{T}\sum_t \mathbb{E}\left\|\Delta V_t\right\|^2}{T} + \frac{1}{Th} + h^{2K}\right) \,.$$

We simplify Theorem 1 and Theorem 2 to Lemma 3, where the functions $\mathcal{B}_{\text{bias}} \triangleq O\left(\frac{1}{Th} + \frac{\sum_t \|\mathbb{E}\Delta V_t\|}{T}\right)$ and $\mathcal{B}_{\text{mse}} \triangleq C\left(\frac{\frac{1}{T}\sum_t \mathbb{E}\|\Delta V_t\|^2}{T} + \frac{1}{Th}\right)$, independent of the order of an integrator.

In addition, from Theorem 1 and Theorem 2, we can get the optimal convergence rate of a SG-MCMC algorithm with respect to the *Bias* and *MSE* by optimizing the bounds. Specifically, the optimal convergence rates of the *Bias* for the Euler integrator is $T^{-1/2}$ with optimal stepsize $\propto T^{-1/2}$, while this is $T^{-2/3}$ for the symmetric splitting operator with optimal stepsize $\propto T^{-1/3}$. For the MSE, the rate for the Euler integrator is $T^{-2/3}$ with optimal stepsize $\propto T^{-1/3}$, compared to $T^{-4/5}$ with optimal stepsize $\propto T^{-1/5}$ for the symmetric splitting integrator.

D Latent Dirichlet Allocation

Following (Ding et al. 2014), this section describes the semi-collapsed posterior of the LDA model, and the *Expanded-Natural* representation of the prabability simplexes used in (Patterson and Teh 2013).

Let $\mathbf{W} = \{w_{jv}\}$ be the observed words, $\mathbf{Z} = \{z_{jv}\}$ be the topic indicator variables, where j indexes the documents and v indexes the words. Let $(\pi)_{kw}$ be the topic-word distribution, n_{jkw} be the number of word w in document j allocated to topic k, $\cdot$ means marginal sum, *i.e.,* $n_{jk\cdot} = \sum_w n_{dkw}$. The semi-collapsed posterior of the LDA model is

$$p(\mathbf{W}, \mathbf{Z}, \pi|\alpha, \tau) = p(\pi|\tau)\prod_{j=1}^{J} p(\boldsymbol{w}_j, \boldsymbol{z}_j|\alpha, \pi), \tag{8}$$

where J is the number of documents, α is the parameter in the Dirichlet prior of the topic distribution for each document, τ is the parameter in the prior of π, and

$$p(\boldsymbol{w}_j, \boldsymbol{z}_j|\alpha, \pi) = \prod_{k=1}^{K}\frac{\Gamma\left(\alpha + n_{jk\cdot}\right)}{\Gamma\left(\alpha\right)}\prod_{w=1}^{W}\pi_{kw}^{n_{jkw}}. \tag{9}$$

The *Expanded-Natural* representation of the simplexes π_k's in (Patterson and Teh 2013) is used, where

$$\pi_{kw} = \frac{e^{\theta_{kw}}}{\sum_{w'} e^{\theta_{kw'}}} \,. \tag{10}$$

Following (Ding et al. 2014), a Gaussian prior on θ_{kw} is adopted,

$$p(\theta_{kw}|\tau = \{\beta, \sigma\}) = \mathcal{N}(\theta_{kw}, \sigma^2) \,.$$

The stochastic gradient of the log-posterior of parameter θ_{kw} with a mini-batch $\mathcal{S}_t$ becomes,

$$\begin{aligned}\frac{\partial\tilde{U}(\boldsymbol{\theta})}{\partial\theta_{kw}} &= \frac{\partial\log\tilde{p}(\boldsymbol{\theta}|\mathbf{W}, \tau, \alpha)}{\partial\theta_{kw}} \\ &= \frac{\beta - \theta_{kw}}{\sigma^2} + \frac{J}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t}\mathbb{E}_{\boldsymbol{z}_j|\boldsymbol{w}_j,\theta,\alpha}\left(n_{jkw} - \pi_{kw}n_{jk\cdot}\right).\end{aligned} \tag{11}$$

for the t-th iteration, where $\mathcal{S}_t \subset \{1, 2, \cdots, J\}$, and $|\cdot|$ is the cardinality of a set.

When $\sigma = 1$, we obtain the same stochastic gradient as in (Patterson and Teh 2013) using Riemannian manifold. To calculate the expectation term, we use the same Gibbs sampling method as in (Patterson and Teh 2013),

$$p(z_{jv} = k|\boldsymbol{w}_d, \boldsymbol{\theta}, \alpha) = \frac{\left(\alpha + n_{jk\cdot}^{\backslash v}\right)e^{\theta_{kw_{jv}}}}{\sum_{k'}\left(\alpha + n_{jk'\cdot}^{\backslash v}\right)e^{\theta_{k'w_{jv}}}}, \tag{12}$$

where $\backslash v$ denotes the count excluding the v-th topic assignment variable. The expectation is estimated by the samples.

Running time For the results reported in the main text, to achieve reported results of LDA on ICML dataset, mSGNHT-E, mSGNHT-S and Gibbs take 1000 iterations, the running times are 161.99, 180.55 and 516.12 second, respectively.

E Logistic Regression

For each data $\{\boldsymbol{x}, y\}$, $\boldsymbol{x} \in \mathbb{R}^P$ is the input data, $y \in \{0, 1\}$ is the label. Logistic Regression gives the prabability

$$P(y|\boldsymbol{x}) \propto g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + \exp\left(-(\mathbf{W}^\top\boldsymbol{x} + \boldsymbol{c})\right)} \,. \tag{13}$$

A Gaussian prior is placed on model parameters $\boldsymbol{\theta} = \{\mathbf{W}, \boldsymbol{c}\} \propto \mathcal{N}(0, \sigma^2\mathbf{I})$. We set $\sigma^2 = 10$ in our experiment.

We study the effectiveness of mSGNHT-S for different h and D. Learning curves of test errors and training log-likelihoods are shown in Fig. 1. Generally, the performances of the proposed mSGNHT-S are consistently more stable than the mSGNHT-E and SGHMC, across varying h and D.

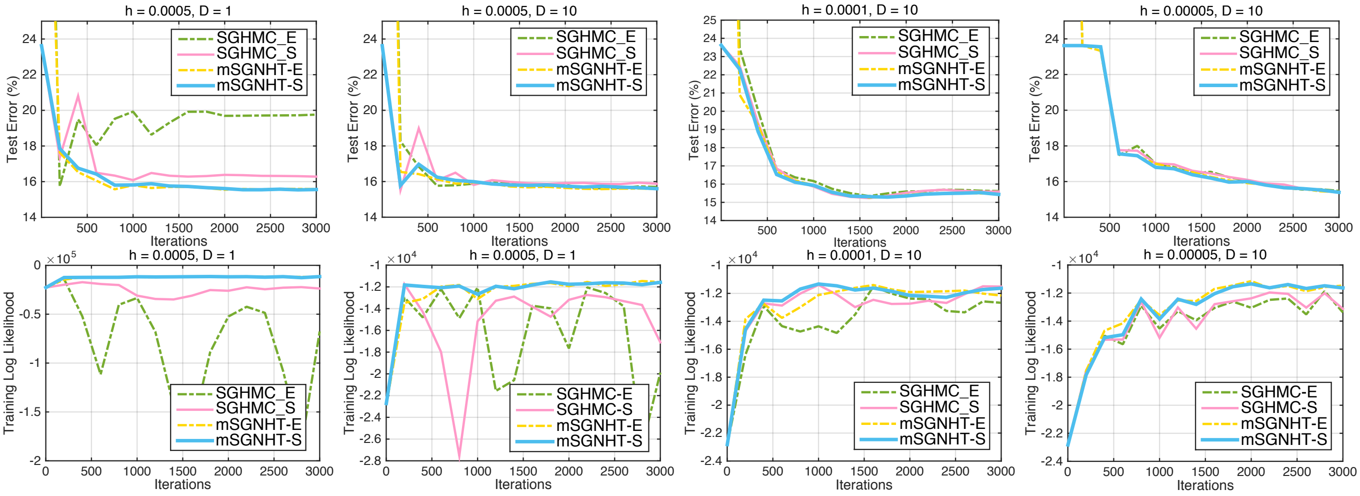

Figure 1: Learning curves for logistic regression on a9a dataset for varying h and D. Column 1-2 share the same h; column 2-4 share the same D. Top row is testing error; bottom row is training log-likelihood.

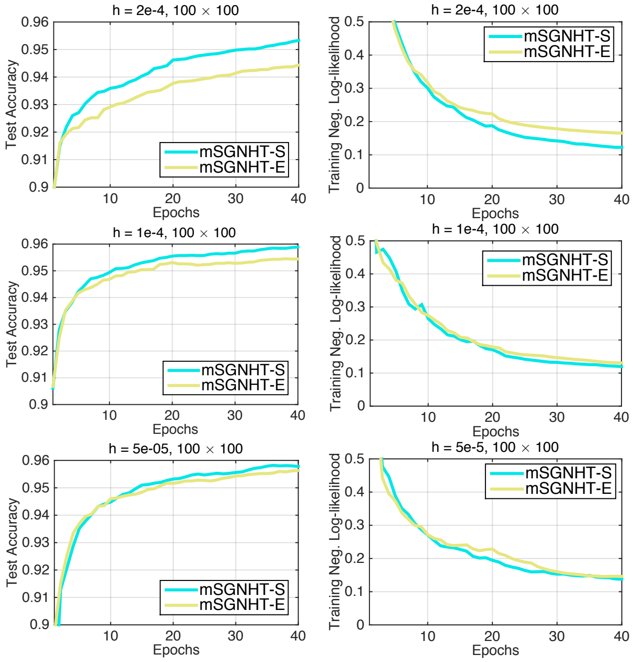

Figure 2: Learning curves for FNN (ReLU link) on MNIST dataset for varying stepsize h. Step size decreases top-down.

Furthermore, mSGNHT-S converges faster than mSGNHT-E, especially at the beginning of learning. Comparing column 1-2 (fixing h, varying D), mSGNHT-S is more robust to the choice of diffusion factor D. Comparing column 2-4 (fixing D, varying h), larger h potentially brings larger gradient-estimation errors and numerical errors. mSGNHT is shown to significantly outperform others when h is large.

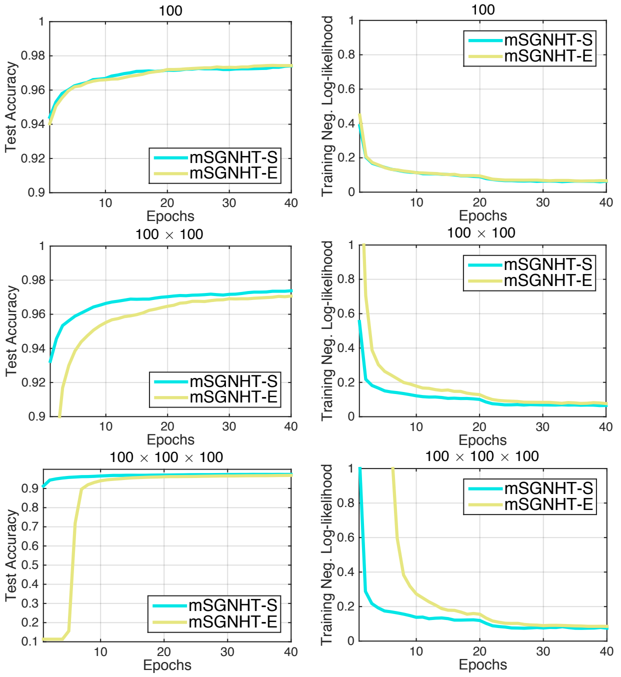

Figure 3: Learning curves for FNN (Sigmoid link) on MNIST dataset for varying network depth. Depth increases top-down.

F Deep Neural Networks

F.1 Sensitivity to Step-size

For feedforward neural nets (FNN) with a 2-layer model of 100-100 ReLU, we study the performance of mSGNHT-S for different h. $D = 5$. We test a wide range of h, and show in Fig. 2 the learning curves of the test accuracy and training negative log-likelihood for $h = 2{\times}10^{-4}, 1{\times}10^{-4}, 5{\times}10^{-5}$. mSGNHT-S is less sensitive to step size; it maintains fast convergence when h is large, while mSGNHT-S significantly slows down.

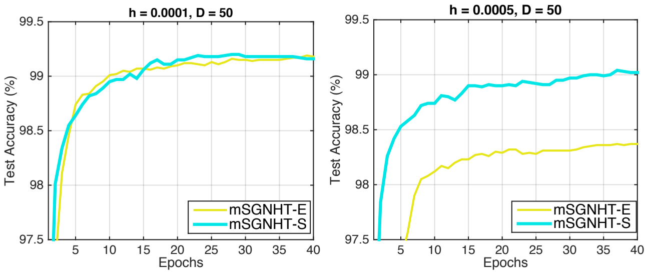

Figure 4: Learning curves of CNN for different step sizes.

F.2 Sigmoid Activation Function

We compare different methods in the case of sigmoid link. Similar to the main paper, we test the FNNs with varying depths, *e.g.*, $\{1, 2, 3\}$, respectively. We set $D = 10$ and $h = 10^{-3}$. Fig. 3 displays learning curves on testing accuracy and training negative log-likelihood. The gaps between mSGNHT-S and mSGNHT-E becomes larger in deeper models. When a 3-layer network is emplyed, mSGNHT-E fails at the first 5 epochs, whilst mSGNHT-S converges pretty well. Moreover, mSGNHT-S converges more accurately, yielding an accuracy 97.36%, while mSGNHT-E gives 96.86%. This manifests the importance of numerical accuracy in SG-MCMC for practical applications, and mSGNHT-S is desirable in this regard.

F.3 Comparison with Other Methods

Based on network size 400-400 with ReLU, we also compared with a recent state-of-the-art inference method for neural nets, Bayes by Backprop (BBB) (Blundell et al. 2015). $h = 2 \times 10^{-4}$ and $D = 60$. The comparison is in Table 1. BBB and SGD are results taken from (Blundell et al. 2015).

Table 1: Classification accuracy on MNIST.

Method	Accuracy (%) ↑
mSGNHT-S	**98.25**
mSGNHT-E	98.20
BBB	98.18
SGD	98.17

F.4 Convolutional Neural Networks

We also performed the comparison with a standard network convolutional neural networks, LeNet (Jarrett et al. 2009), on MNIST dataset. It is 2-layer convolution networks followed by a 2-layer fully-connected FNN, each containing 200 hidden nodes that uses ReLU. Both convolutional layers use 5×5 filter size with 32 and 64 channels, respectively, 2×2 max pooling are used after each convolutional layer. 40 epochs are used, and L is set to 20. In Fig. 4, we tested the stepsizes $h = 10^{-4}$ and $h = 5 \times 10^{-4}$ for mSGNHT-S and mSGNHT-S, and $D = 50$.

Again, under the same network architecture, CNN trained with mSGNHT-S converges fater than mSGNHT-E. In particular, when a large stepsize used, mSGNHT-S has a significant improvement over mSGNHT-E.

G Deep Poisson Factor Analysis

G.1 Model Specification

We first provide model details of Deep Poisson Factor Analysis (DPFA) (Gan et al. 2015). Given a discrete matrix $\mathbf{W} \in \mathbb{Z}_+^{V \times J}$ containing counts from J documents and V words, Poisson factor analysis (PFA) (Zhou et al. 2012) assumes the entries of $\mathbf{W}$ are summations of $K < \infty$ latent counts, each produced by a latent factor (in the case of topic modeling, a hidden topic). For $\mathbf{W}$, the generative process of DPFA with L-layer Sigmoid Belief Networks (SBN), is as follows

$$p(h_{k_L}^{(L)}) = g(b_{k_L}^{(L)}) \tag{14}$$

$$p(h_{k_\ell n}^{(\ell)} = 1 | \boldsymbol{h}_n^{(\ell+1)}) = g\big((\boldsymbol{w}_{k_\ell}^{(\ell)})^\top \boldsymbol{h}_n^{(\ell+1)} + c_{k_\ell}^{(\ell)}\big) \tag{15}$$

$$\mathbf{W} \sim \text{Pois}(\boldsymbol{\Phi}(\boldsymbol{\Psi} \odot \boldsymbol{h}^{(1)})) \tag{16}$$

where $g(\cdot)$ is the Sigmoid link. Equation (14) and (15) define Deep Sigmoid Belief Networks (DSBN). $\boldsymbol{\Phi}$ is the factor loading matrix. Each column of $\boldsymbol{\Phi}$, $\boldsymbol{\phi}_k \in \Delta_V$, encodes the relative importance of each word in topic k, with Δ_V representing the V-dimensional simplex. $\boldsymbol{\Theta} \in \mathbb{R}_+^{K \times J}$ is the factor score matrix. Each column $\boldsymbol{\psi}_j$, contains relative topic intensities specific to document j. $\boldsymbol{h}^{(\ell)} \in \{0, 1\}^{K \times 1}$ is a latent binary feature vector, which defines whether certain topics are associated with documents. The factor scores for document j at bottom layer are the element-wise multiplication $\boldsymbol{\psi}_j \odot \boldsymbol{h}^{(1)}$. DPFA is constructed by placing Dirichlet priors on $\boldsymbol{\Phi}_k$ and gamma priors on $\boldsymbol{\psi}_j$. This is summarized as,

$$x_{vj} = \sum_{k=1}^{K} x_{vjk}, \qquad x_{vjk} \sim \text{Pois}(\phi_{vk}\psi_{kj}h_k) \tag{17}$$

with priors specified as $\boldsymbol{\phi}_k \sim \text{Dir}(a_\phi, \ldots, a_\phi)$, $\psi_{kn} \sim \text{Gamma}(r_k, p_k/(1 - p_k))$, $r_k \sim \text{Gamma}(\gamma_0, 1/c_0)$, and $\gamma_0 \sim \text{Gamma}(e_0, 1/f_0)$.

G.2 Model Inference

Following (Gan et al. 2015), we use stochastic gradient Riemannian Langevin dynamics (SGRLD) (Patterson and Teh 2013) to sample the topic-word distributions $\{\boldsymbol{\phi}_k\}$. mSGNHT is used to sample the parameters in DSBN, *i.e.*, $\boldsymbol{\theta} = \big(\mathbf{W}^{(\ell)}, \boldsymbol{c}^{(\ell)}, \boldsymbol{b}^{(L)}\big)$, where $\ell = 1, \cdots, L - 1$. Specifically, the stochastic gradients of $\mathbf{W}^{(\ell)}$ and $\boldsymbol{c}^{(\ell)}$ evaluated on a mini-batch of data (denote $\mathcal{S}$ as the index set of a mini-batch) are calculated,

$$\frac{\partial \tilde{U}}{\partial \boldsymbol{w}_{k_\ell}^{(\ell)}} = \frac{J}{|\mathcal{S}|} \sum_{i \in \mathcal{D}} \mathbb{E}_{\boldsymbol{h}_i^{(\ell)}, \boldsymbol{h}_i^{(\ell+1)}} \left[\left(\tilde{\sigma}_{k_\ell i}^{(\ell)} - h_{k_\ell i}^{(\ell)} \right) \boldsymbol{h}_i^{(\ell+1)} \right], \tag{18}$$

$$\frac{\partial \tilde{U}}{\partial c_{k_\ell}^{(\ell)}} = \frac{J}{|\mathcal{S}|} \sum_{i \in \mathcal{D}} \mathbb{E}_{\boldsymbol{h}_i^{(\ell)}, \boldsymbol{h}_i^{(\ell+1)}} \left[\tilde{\sigma}_{k_\ell i}^{(\ell)} - h_{k_\ell i}^{(\ell)} \right], \tag{19}$$

where $\tilde{\sigma}_{k_\ell i}^{(\ell)} = \sigma((\boldsymbol{w}_{k_\ell}^{(\ell)})^\top \boldsymbol{h}_i^{(\ell+1)} + c_{k_\ell}^{(\ell)})$, and the expectation is taken over posteriors. Monte Carlo integration is used to approximate this quantity.

References

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural networks. In *ICML*.

Chen, C.; Ding, N.; and Carin, L. 2015. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *NIPS*.

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

Gan, Z.; Chen, C.; Henao, R.; Carlson, D.; and Carin, L. 2015. Scalable deep Poisson factor analysis for topic modeling. In *ICML*.

Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.; and LeCun, Y. 2009. What is the best multi-stage architecture for object recognition? In *ICCV*.

Patterson, S., and Teh, Y. W. 2013. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *NIPS*.

Rossmann, W. 2002. *Lie Groups–An Introduction Through Linear Groups*. Oxford Graduate Texts in Mathematics, Oxford Science Publications.

Zhou, M.; Hannah, L.; Dunson, D. B.; and Carin, L. 2012. Beta-negative Binomial process and Poisson factor analysis. In *AISTATS*.