# Stochastic Gradient Hamiltonian Monte Carlo

**Tianqi Chen** TQCHEN@CS.WASHINGTON.EDU
**Emily B. Fox** EBFOX@STAT.WASHINGTON.EDU
**Carlos Guestrin** GUESTRIN@CS.WASHINGTON.EDU
MODE Lab, University of Washington, Seattle, WA.

## Abstract

Hamiltonian Monte Carlo (HMC) sampling methods provide a mechanism for defining distant proposals with high acceptance probabilities in a Metropolis-Hastings framework, enabling more efficient exploration of the state space than standard random-walk proposals. The popularity of such methods has grown significantly in recent years. However, a limitation of HMC methods is the required gradient computation for simulation of the Hamiltonian dynamical system—such computation is infeasible in problems involving a large sample size or streaming data. Instead, we must rely on a noisy gradient estimate computed from a subset of the data. In this paper, we explore the properties of such a stochastic gradient HMC approach. Surprisingly, the natural implementation of the stochastic approximation can be arbitrarily bad. To address this problem we introduce a variant that uses second-order Langevin dynamics with a friction term that counteracts the effects of the noisy gradient, maintaining the desired target distribution as the invariant distribution. Results on simulated data validate our theory. We also provide an application of our methods to a classification task using neural networks and to online Bayesian matrix factorization.

## 1. Introduction

Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 2010) sampling methods provide a powerful Markov chain Monte Carlo (MCMC) sampling algorithm. The methods define a Hamiltonian function in terms of the target distribution from which we desire samples—the *potential energy*—and a *kinetic energy* term parameterized by a set of "momentum" auxiliary variables. Based on simple updates to the momentum variables, one simulates from a Hamiltonian dynamical system that enables proposals of distant states. The target distribution is invariant under these dynamics; in practice, a discretization of the continuous-time system is needed necessitating a Metropolis-Hastings (MH) correction, though still with high acceptance probability. Based on the attractive properties of HMC in terms of rapid exploration of the state space, HMC methods have grown in popularity recently (Neal, 2010; Hoffman & Gelman, 2011; Wang et al., 2013).

A limitation of HMC, however, is the necessity to compute the gradient of the potential energy function in order to simulate the Hamiltonian dynamical system. We are increasingly faced with datasets having millions to billions of observations or where data come in as a stream and we need to make inferences online, such as in online advertising or recommender systems. In these ever-more-common scenarios of massive batch or streaming data, such gradient computations are infeasible since they utilize the entire dataset, and thus are not applicable to "big data" problems. Recently, in a variety of machine learning algorithms, we have witnessed the many successes of utilizing a noisy estimate of the gradient based on a *minibatch* of data to scale the algorithms (Robbins & Monro, 1951; Hoffman et al., 2013; Welling & Teh, 2011). A majority of these developments have been in optimization-based algorithms (Robbins & Monro, 1951; Nemirovski et al., 2009), and a question is whether similar efficiencies can be garnered by sampling-based algorithms that maintain many desirable theoretical properties for Bayesian inference. One attempt at applying such methods in a sampling context is the recently proposed stochastic gradient Langevin dynamics (SGLD) (Welling & Teh, 2011; Ahn et al., 2012; Patterson & Teh, 2013). This method builds on first-order Langevin dynamics that do not include the crucial momentum term of HMC.

In this paper, we explore the possibility of marrying the efficiencies in state space exploration of HMC with the big-data computational efficiencies of stochastic gradients. Such an algorithm would enable a large-scale and online

*Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 2014. JMLR: W&CP volume 32. 

Bayesian sampling algorithm with the potential to rapidly explore the posterior. As a first cut, we consider simply applying a stochastic gradient modification to HMC and assess the impact of the noisy gradient. We prove that the noise injected in the system by the stochastic gradient no longer leads to Hamiltonian dynamics with the desired target distribution as the stationary distribution. As such, even before discretizing the dynamical system, we need to correct for this effect. One can correct for the injected gradient noise through an MH step, though this itself requires costly computations on the entire dataset. In practice, one might propose long simulation runs before an MH correction, but this leads to low acceptance rates due to large deviations in the Hamiltonian from the injected noise. The efficiency of this MH step could potentially be improved using the recent results of (Korattikara et al., 2014; Bardenet et al., 2014). In this paper, we instead introduce a stochastic gradient HMC method with *friction* added to the momentum update. We assume the injected noise is Gaussian, appealing to the central limit theorem, and analyze the corresponding dynamics. We show that using such *second-order Langevin dynamics* enables us to maintain the desired target distribution as the stationary distribution. That is, the friction counteracts the effects of the injected noise. For discretized systems, we consider letting the step size tend to zero so that an MH step is not needed, giving us a significant computational advantage. Empirically, we demonstrate that we have good performance even for $\epsilon$ set to a small, fixed value. The theoretical computation versus accuracy tradeoff of this small-$\epsilon$ approach is provided in the Supplementary Material.

A number of simulated experiments validate our theoretical results and demonstrate the differences between (i) exact HMC, (ii) the naïve implementation of stochastic gradient HMC (simply replacing the gradient with a stochastic gradient), and (iii) our proposed method incorporating friction. We also compare to the first-order Langevin dynamics of SGLD. Finally, we apply our proposed methods to a classification task using Bayesian neural networks and to online Bayesian matrix factorization of a standard movie dataset. Our experimental results demonstrate the effectiveness of the proposed algorithm.

## 2. Hamiltonian Monte Carlo

Suppose we want to sample from the posterior distribution of $\theta$ given a set of independent observations $x \in \mathcal{D}$:

$$p(\theta|\mathcal{D}) \propto \exp(-U(\theta)), \tag{1}$$

where the *potential energy* function $U$ is given by

$$U = -\sum_{x\in\mathcal{D}} \log p(x|\theta) - \log p(\theta). \tag{2}$$

Hamiltonian (Hybrid) Monte Carlo (HMC) (Duane et al., 1987; Neal, 2010) provides a method for proposing samples of $\theta$ in a Metropolis-Hastings (MH) framework that efficiently explores the state space as compared to standard random-walk proposals. These proposals are generated from a Hamiltonian system based on introducing a set of auxiliary momentum variables, $r$. That is, to sample from $p(\theta|\mathcal{D})$, HMC considers generating samples from a joint distribution of $(\theta, r)$ defined by

$$\pi(\theta, r) \propto \exp\left(-U(\theta) - \frac{1}{2} r^T M^{-1} r\right). \tag{3}$$

If we simply discard the resulting $r$ samples, the $\theta$ samples have marginal distribution $p(\theta|\mathcal{D})$. Here, $M$ is a mass matrix, and together with $r$, defines a *kinetic energy* term. $M$ is often set to the identity matrix, $I$, but can be used to precondition the sampler when we have more information about the target distribution. The Hamiltonian function is defined by $H(\theta, r) = U(\theta) + \frac{1}{2} r^T M^{-1} r$. Intuitively, $H$ measures the total energy of a physical system with position variables $\theta$ and momentum variables $r$.

To propose samples, HMC simulates the Hamiltonian dynamics

$$\begin{cases} d\theta = & M^{-1} r \, dt \\ dr = & -\nabla U(\theta) \, dt. \end{cases} \tag{4}$$

To make Eq. (4) concrete, a common analogy in 2D is as follows (Neal, 2010). Imagine a hockey puck sliding over a frictionless ice surface of varying height. The potential energy term is based on the height of the surface at the current puck position, $\theta$, while the kinetic energy is based on the momentum of the puck, $r$, and its mass, $M$. If the surface is flat ($\nabla U(\theta) = 0, \forall \theta$), the puck moves at a constant velocity. For positive slopes ($\nabla U(\theta) > 0$), the kinetic energy decreases as the potential energy increases until the kinetic energy is 0 ($r = 0$). The puck then slides back down the hill increasing its kinetic energy and decreasing potential energy. Recall that in HMC, the position variables are those of direct interest whereas the momentum variables are artificial constructs (auxiliary variables).

Over any interval $s$, the Hamiltonian dynamics of Eq. (4) defines a mapping from the state at time $t$ to the state at time $t + s$. Importantly, this mapping is reversible, which is important in showing that the dynamics leave $\pi$ invariant. Likewise, the dynamics preserve the total energy, $H$, so proposals are always accepted. In practice, however, we usually cannot simulate exactly from the continuous system of Eq. (4) and instead consider a discretized system. One common approach is the "leapfrog" method, which is outlined in Alg. 1. Because of inaccuracies introduced through the discretization, an MH step must be implemented (i.e., the acceptance rate is no longer 1). However, acceptance rates still tend to be high even for proposals that can be quite far from their last state.

**Algorithm 1:** Hamiltonian Monte Carlo

**Input**: Starting position $\theta^{(1)}$ and step size $\epsilon$
**for** $t = 1, 2 \cdots$ **do**
  *Resample momentum* $r$
  $r^{(t)} \sim \mathcal{N}(0, M)$
  $(\theta_0, r_0) = (\theta^{(t)}, r^{(t)})$
  *Simulate discretization of Hamiltonian dynamics in Eq.* (4)*:*
  $r_0 \leftarrow r_0 - \frac{\epsilon}{2}\nabla U(\theta_0)$
  **for** $i = 1$ ***to*** $m$ **do**
    $\theta_i \leftarrow \theta_{i-1} + \epsilon M^{-1} r_{i-1}$
    $r_i \leftarrow r_{i-1} - \epsilon \nabla U(\theta_i)$
  **end**
  $r_m \leftarrow r_m - \frac{\epsilon}{2}\nabla U(\theta_m)$
  $(\hat{\theta}, \hat{r}) = (\theta_m, r_m)$
  *Metropolis-Hastings correction*:
  $u \sim \text{Uniform}[0, 1]$
  $\rho = e^{H(\hat{\theta},\hat{r}) - H(\theta^{(t)}, r^{(t)})}$
  **if** $u < \min(1, \rho)$, **then** $\theta^{(t+1)} = \hat{\theta}$
**end**

There have been many recent developments of HMC to make the algorithm more flexible and applicable in a variety of settings. The "No U-Turn" sampler (Hoffman & Gelman, 2011) and the methods proposed by Wang et al. (2013) allow automatic tuning of the step size, $\epsilon$, and number of simulation steps, $m$. Riemann manifold HMC (Girolami & Calderhead, 2011) makes use of the Riemann geometry to adapt the mass $M$, enabling the algorithm to make use of curvature information to perform more efficient sampling. We attempt to improve HMC in an orthogonal direction focused on computational complexity, but these adaptive HMC techniques could potentially be combined with our proposed methods to see further benefits.

## 3. Stochastic Gradient HMC

In this section, we study the implications of implementing HMC using a stochastic gradient and propose variants on the Hamiltonian dynamics that are more robust to the noise introduced by the stochastic gradient estimates. In all scenarios, instead of directly computing the costly gradient $\nabla U(\theta)$ using Eq. (2), which requires examination of the entire dataset $\mathcal{D}$, we consider a noisy estimate based on a *minibatch* $\tilde{\mathcal{D}}$ sampled uniformly at random from $\mathcal{D}$:

$$\nabla \tilde{U}(\theta) = -\frac{|\mathcal{D}|}{|\tilde{\mathcal{D}}|} \sum_{x \in \tilde{\mathcal{D}}} \nabla \log p(x|\theta) - \nabla \log p(\theta), \ \tilde{\mathcal{D}} \subset \mathcal{D}. \tag{5}$$

We assume that our observations $x$ are *independent* and, appealing to the central limit theorem, approximate this noisy gradient as

$$\nabla \tilde{U}(\theta) \approx \nabla U(\theta) + \mathcal{N}(0, V(\theta)). \tag{6}$$

Here, $V$ is the covariance of the stochastic gradient noise, which can depend on the current model parameters and sample size. Note that we use an abuse of notation in Eq. (6) where the addition of $\mathcal{N}(\mu, \Sigma)$ denotes the introduction of a random variable that is distributed according to this multivariate Gaussian. As the size of $\tilde{\mathcal{D}}$ increases, this Gaussian approximation becomes more accurate. Clearly, we want minibatches to be small to have our sought-after computational gains. Empirically, in a wide range of settings, simply considering a minibatch size on the order of hundreds of data points is sufficient for the central limit theorem approximation to be accurate (Ahn et al., 2012). In our applications of interest, minibatches of this size still represent a significant reduction in the computational cost of the gradient.

### 3.1. Naïve Stochastic Gradient HMC

The most straightforward approach to stochastic gradient HMC is simply to replace $\nabla U(\theta)$ in Alg. 1 by $\nabla \tilde{U}(\theta)$. Referring to Eq. (6), this introduces noise in the momentum update, which becomes $\Delta r = -\epsilon \nabla \tilde{U}(\theta) = -\epsilon \nabla U(\theta) + \mathcal{N}(0, \epsilon^2 V)$. The resulting discrete time system can be viewed as an $\epsilon$-discretization of the following continuous stochastic differential equation:

$$\begin{cases} d\theta = & M^{-1} r \, dt \\ dr = & -\nabla U(\theta) \, dt + \mathcal{N}(0, 2B(\theta) dt). \end{cases} \tag{7}$$

Here, $B(\theta) = \frac{1}{2}\epsilon V(\theta)$ is the diffusion matrix contributed by gradient noise. As with the original HMC formulation, it is useful to return to a continuous time system in order to derive properties of the approach. To gain some intuition about this setting, consider the same hockey puck analogy of Sec. 2. Here, we can imagine the puck on the same ice surface, but with some random wind blowing as well. This wind may blow the puck further away than expected. Formally, as given by Corollary 3.1 of Theorem 3.1, when $B$ is nonzero, $\pi(\theta, r)$ of Eq. (3) is no longer invariant under the dynamics described by Eq. (7).

**Theorem 3.1.** *Let $p_t(\theta, r)$ be the distribution of $(\theta, r)$ at time $t$ with dynamics governed by Eq. (7). Define the entropy of $p_t$ as $h(p_t) = -\int_{\theta,r} f(p_t(\theta, r)) d\theta dr$, where $f(x) = x \ln x$. Assume $p_t$ is a distribution with density and gradient vanishing at infinity. Furthermore, assume the gradient vanishes faster than $\frac{1}{\ln p_t}$. Then, the entropy of $p_t$ increases over time with rate*

$$\partial_t h(p_t(\theta, r)) = \int_{\theta,r} f''(p_t)(\nabla_r p_t(\theta, r))^T B(\theta) \nabla_r p_t(\theta, r) d\theta dr. \tag{8}$$

*Eq.* (8) *implies that* $\partial_t h(p_t(\theta, r)) \geq 0$ *since* $B(\theta)$ *is a positive semi-definite matrix.*

Intuitively, Theorem 3.1 is true because the noise-free Hamiltonian dynamics preserve entropy, while the additional noise term strictly increases entropy if we assume (i) $B(\theta)$ is positive definite (a reasonable assumption due to the normal full rank property of Fisher information) and (ii) $\nabla_r p_t(\theta, r) \neq 0$ for all $t$. Then, jointly, the entropy strictly increases over time. This hints at the fact that the distribution $p_t$ tends toward a uniform distribution, which can be very far from the target distribution $\pi$.

**Corollary 3.1.** *The distribution* $\pi(\theta, r) \propto \exp\left(-H(\theta, r)\right)$ *is no longer invariant under the dynamics in Eq.* (7).

The proofs of Theorem 3.1 and Corollary 3.1 are in the Supplementary Material.

Because $\pi$ is no longer invariant under the dynamics of Eq. (7), we must introduce a correction step even before considering errors introduced by the discretization of the dynamical system. For the correctness of an MH step (based on the entire dataset), we appeal to the same arguments made for the HMC data-splitting technique of Neal (2010). This approach likewise considers minibatches of data and simulating the (continuous) Hamiltonian dynamics on each batch sequentially. Importantly, Neal (2010) alludes to the fact that the resulting $H$ from the split-data scenario may be far from that of the full-data scenario after simulation, which leads to lower acceptance rates and thereby reduces the apparent computational gains in simulation. Empirically, as we demonstrate in Fig. 2, we see that even finite-length simulations from the noisy system can diverge quite substantially from those of the noise-free system. Although the minibatch-based HMC technique considered herein is slightly different from that of Neal (2010), the theory we have developed in Theorem 3.1 surrounding the high-entropy properties of the resulting invariant distribution of Eq. (7) provides some intuition for the observed deviations in $H$ both in our experiments and those of Neal (2010).

The poorly behaved properties of the trajectory of $H$ based on simulations using noisy gradients results in a complex computation versus efficiency tradeoff. On one hand, it is extremely computationally intensive in large datasets to insert an MH step after just short simulation runs (where deviations in $H$ are less pronounced and acceptance rates should be reasonable). Each of these MH steps requires a costly computation using *all* of the data, thus defeating the computational gains of considering noisy gradients. On the other hand, long simulation runs between MH steps can lead to very low acceptance rates. Each rejection corresponds to a wasted (noisy) gradient computation and simulation using the proposed variant of Alg. 1. One possible direction of future research is to consider using the recent results of Korattikara et al. (2014) and Bardenet et al. (2014) that show that it is possible to do MH using a subset of data. However, we instead consider in Sec. 3.2 a straightforward modification to the Hamiltonian dynamics that alleviates the issues of the noise introduced by stochastic gradients. In particular, our modification allows us to again achieve the desired $\pi$ as the invariant distribution of the continuous Hamiltonian dynamical system.

### 3.2. Stochastic Gradient HMC with Friction

In Sec. 3.1, we showed that HMC with stochastic gradients requires a frequent costly MH correction step, or alternatively, long simulation runs with low acceptance probabilities. Ideally, instead, we would like to minimize the effect of the injected noise on the dynamics themselves to alleviate these problems. To this end, we consider a modification to Eq. (7) that adds a "friction" term to the momentum update:

$$\begin{cases} d\theta = M^{-1} r \, dt \\ dr = -\nabla U(\theta) \, dt - BM^{-1} r dt + \mathcal{N}(0, 2B dt). \end{cases} \tag{9}$$

Here and throughout the remainder of the paper, we omit the dependence of $B$ on $\theta$ for simplicity of notation. Let us again make a hockey analogy. Imagine we are now playing street hockey instead of ice hockey, which introduces friction from the asphalt. There is still a random wind blowing, however the friction of the surface prevents the puck from running far away. That is, the friction term $BM^{-1}r$ helps decrease the energy $H(\theta, r)$, thus reducing the influence of the noise. This type of dynamical system is commonly referred to as *second-order Langevin dynamics* in physics (Wang & Uhlenbeck, 1945). Importantly, we note that the Langevin dynamics used in SGLD (Welling & Teh, 2011) are first-order, which can be viewed as a limiting case of our second-order dynamics when the friction term is large. Further details on this comparison follow at the end of this section.

**Theorem 3.2.** $\pi(\theta, r) \propto \exp(-H(\theta, r))$ *is the unique stationary distribution of the dynamics described by Eq.* (9).

*Proof.* Let $G = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$, $D = \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix}$, where $G$ is an anti-symmetric matrix, and $D$ is the symmetric (diffusion) matrix. Eq. (9) can be written in the following decomposed form (Yin & Ao, 2006; Shi et al., 2012)

$$\begin{aligned} d \begin{bmatrix} \theta \\ r \end{bmatrix} &= - \begin{bmatrix} 0 & -I \\ I & B \end{bmatrix} \begin{bmatrix} \nabla U(\theta) \\ M^{-1} r \end{bmatrix} dt + \mathcal{N}(0, 2D dt) \\ &= - \left[D + G\right] \nabla H(\theta, r) dt + \mathcal{N}(0, 2D dt). \end{aligned}$$

The distribution evolution under this dynamical system is

governed by a Fokker-Planck equation

$$\partial_t p_t(\theta, r) = \nabla^T \{[D+G]\,[p_t(\theta, r)\nabla H(\theta, r) + \nabla p_t(\theta, r)]\}. \tag{10}$$

See the Supplementary Material for details. We can verify that $\pi(\theta, r)$ is invariant under Eq. (10) by calculating $\left[e^{-H(\theta,r)}\nabla H(\theta, r) + \nabla e^{-H(\theta,r)}\right] = 0$. Furthermore, due to the existence of diffusion noise, $\pi$ is the unique stationary distribution of Eq. (10). □

In summary, we have shown that the dynamics given by Eq. (9) have a similar invariance property to that of the original Hamiltonian dynamics of Eq. (4), even with noise present. The key was to introduce a friction term using second-order Langevin dynamics. Our revised momentum update can also be viewed as akin to partial momentum refreshment (Horowitz, 1991; Neal, 1993), which also corresponds to second-order Langevin dynamics. Such partial momentum refreshment was shown to not greatly improve HMC in the case of noise-free gradients (Neal, 2010). However, as we have demonstrated, the idea is crucial in our stochastic gradient scenario in order to counterbalance the effect of the noisy gradients. We refer to the resulting method as *stochastic gradient HMC* (SGHMC).

#### Connection to First-order Langevin Dynamics

As we previously discussed, the dynamics introduced in Eq. (9) relate to the first-order Langevin dynamics used in SGLD (Welling & Teh, 2011). In particular, the dynamics of SGLD can be viewed as second-order Langevin dynamics with a large friction term. To intuitively demonstrate this connection, let $BM^{-1} = \frac{1}{dt}$ in Eq. (9). Because the friction and momentum noise terms are very large, the momentum variable $r$ changes much faster than $\theta$. Thus, relative to the rapidly changing momentum, $\theta$ can be considered as *fixed*. We can study this case as simply:

$$dr = -\nabla U(\theta)dt - BM^{-1}rdt + \mathcal{N}(0, 2Bdt) \tag{11}$$

The fast evolution of $r$ leads to a rapid convergence to the stationary distribution of Eq. (11), which is given by $\mathcal{N}(MB^{-1}\nabla U(\theta), M)$. Let us now consider a change in $\theta$, with $r \sim \mathcal{N}(MB^{-1}\nabla U(\theta), M)$. Recalling $BM^{-1} = \frac{1}{dt}$, we have

$$d\theta = -M^{-1}\nabla U(\theta)dt^2 + \mathcal{N}(0, 2M^{-1}dt^2), \tag{12}$$

which exactly aligns with the dynamics of SGLD where $M^{-1}$ serves as the preconditioning matrix (Welling & Teh, 2011). Intuitively, this means that when the friction is large, the dynamics do not depend on the decaying series of past gradients represented by $dr$, reducing to first-order Langevin dynamics.

**Algorithm 2:** Stochastic Gradient HMC

**for** $t = 1, 2 \cdots$ **do**
  *optionally, resample momentum* $r$ *as* $r^{(t)} \sim \mathcal{N}(0, M)$
  $(\theta_0, r_0) = (\theta^{(t)}, r^{(t)})$
  *simulate dynamics in Eq.*(13)*:*
  **for** $i = 1$ ***to*** $m$ **do**
    $\theta_i \leftarrow \theta_{i-1} + \epsilon_t M^{-1} r_{i-1}$
    $r_i \leftarrow r_{i-1} - \epsilon_t \nabla \tilde{U}(\theta_i) - \epsilon_t C M^{-1} r_{i-1} + \mathcal{N}(0, 2(C - \hat{B})\epsilon_t)$
  **end**
  $(\theta^{(t+1)}, r^{(t+1)}) = (\theta_m, r_m)$, no M-H step
**end**

### 3.3. Stochastic Gradient HMC in Practice

In everything we have considered so far, we have assumed that we know the noise model $B$. Clearly, in practice this is not the case. Imagine instead that we simply have an estimate $\hat{B}$. As will become clear, it is beneficial to instead introduce a user specified friction term $C \succeq \hat{B}$ and consider the following dynamics

$$\begin{cases} d\theta = M^{-1}r\,dt \\ dr = -\nabla U(\theta)\,dt - CM^{-1}rdt \\ \qquad\quad + \mathcal{N}(0, 2(C - \hat{B})dt) + \mathcal{N}(0, 2Bdt) \end{cases} \tag{13}$$

The resulting SGHMC algorithm is shown in Alg. 2. Note that the algorithm is purely in terms of user-specified or computable quantities. To understand our choice of dynamics, we begin with the unrealistic scenario of perfect estimation of $B$.

**Proposition 3.1.** *If $\hat{B} = B$, then the dynamics of Eq.* (13) *yield the stationary distribution $\pi(\theta, r) \propto e^{-H(\theta,r)}$.*

*Proof.* The momentum update simplifies to $r = -\nabla U(\theta)\,dt - CM^{-1}rdt + \mathcal{N}(0, 2Cdt)$, with friction term $CM^{-1}$ and noise term $\mathcal{N}(0, 2Cdt)$. Noting that the proof of Theorem 3.2 only relied on a matching of noise and friction, the result follows directly by using $C$ in place of $B$ in Theorem 3.2. □

Now consider the benefit of introducing the $C$ terms and revised dynamics in the more realistic scenario of inaccurate estimation of $B$. For example, the simplest choice is $\hat{B} = 0$. Though the true stochastic gradient noise $B$ is clearly non-zero, as the step size $\epsilon \to 0$, $B = \frac{1}{2}\epsilon V$ goes to 0 and $C$ dominates. That is, the dynamics are again governed by the controllable injected noise $\mathcal{N}(0, 2Cdt)$ and friction $CM^{-1}$. It is also possible to set $\hat{B} = \frac{1}{2}\epsilon\hat{V}$, where $\hat{V}$ is estimated using empirical Fisher information as in (Ahn et al., 2012) for SGLD.

### COMPUTATIONAL COMPLEXITY

The complexity of Alg. 2 depends on the choice of $M$, $C$ and $\hat{B}$, and the complexity for estimating $\nabla\tilde{U}(\theta)$—denoted as $g(|\mathcal{D}|, d)$—where $d$ is the dimension of the parameter space. Assume we allow $\hat{B}$ to be an arbitrary $d \times d$ positive definite matrix. Using empirical Fisher information estimation of $\hat{B}$, the per-iteration complexity of this estimation step is $O(d^2|\tilde{\mathcal{D}}|)$. Then, the time complexity for the $(\theta, r)$ update is $O(d^3)$, because the update is dominated by generating Gaussian noise with a full covariance matrix. In total, the per-iteration time complexity is $O(d^2|\tilde{\mathcal{D}}| + d^3 + g(|\tilde{\mathcal{D}}|, d))$. In practice, we restrict all of the matrices to be diagonal when $d$ is large, resulting in time complexity $O(d|\tilde{\mathcal{D}}| + d + g(|\tilde{\mathcal{D}}|, d))$. Importantly, we note that our SGHMC time complexity is the *same* as that of SGLD (Welling & Teh, 2011; Ahn et al., 2012) in both parameter settings.

In practice, we must assume inaccurate estimation of $B$. For a decaying series of step sizes $\epsilon_t$, an MH step is not required (Welling & Teh, 2011; Ahn et al., 2012)[1]. However, as the step size decreases, the efficiency of the sampler likewise decreases since proposals are increasingly close to their initial value. In practice, we may want to tolerate some errors in the sampling accuracy to gain efficiency. As in (Welling & Teh, 2011; Ahn et al., 2012) for SGLD, we consider using a small, non-zero $\epsilon$ leading to some bias. We explore an analysis of the errors introduced by such finite-$\epsilon$ approximations in the Supplementary Material.

### CONNECTION TO SGD WITH MOMENTUM

Adding a momentum term to stochastic gradient descent (SGD) is common practice. In concept, there is a clear relationship between SGD with momentum and SGHMC, and here we formalize this connection. Letting $v = \epsilon M^{-1} r$, we first rewrite the update rule in Alg. 2 as

$$\begin{cases} \Delta\theta = v \\ \Delta v = -\epsilon^2 M^{-1}\nabla\tilde{U}(\theta) - \epsilon M^{-1} C v \\ \qquad\qquad + \mathcal{N}(0, 2\epsilon^3 M^{-1}(C - \hat{B})M^{-1}). \end{cases} \tag{14}$$

Define $\eta = \epsilon^2 M^{-1}$, $\alpha = \epsilon M^{-1} C$, $\hat{\beta} = \epsilon M^{-1}\hat{B}$. The update rule becomes

$$\begin{cases} \Delta\theta = v \\ \Delta v = -\eta\nabla\tilde{U}(x) - \alpha v + \mathcal{N}(0, 2(\alpha - \hat{\beta})\eta). \end{cases} \tag{15}$$

Comparing to an SGD with momentum method, it is clear from Eq. (15) that $\eta$ corresponds to the learning rate and $1-\alpha$ the momentum term. When the noise is removed (via $C = \hat{B} = 0$), SGHMC naturally reduces to a stochastic gradient method with momentum. We can use the equivalent update rule of Eq. (15) to run SGHMC, and borrow experience from parameter settings of SGD with momentum to guide our choices of SGHMC settings. For example, we can set $\alpha$ to a fixed small number (e.g., 0.01 or 0.1), select the learning rate $\eta$, and then fix $\hat{\beta} = \eta\hat{V}/2$. A more sophisticated strategy involves using momentum scheduling (Sutskever et al., 2013). We elaborate upon how to select these parameters in the Supplementary Material.

[1] We note that, just as in SGLD, an MH correction is not even possible because we cannot compute the probability of the reverse dynamics.

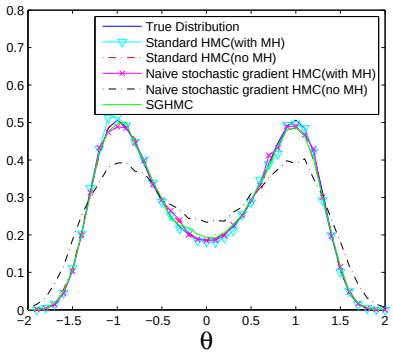


*Figure 1.* Empirical distributions associated with various sampling algorithms relative to the true target distribution with $U(\theta) = -2\theta^2 + \theta^4$. We compare the HMC method of Alg. 1 with and without the MH step to: (i) a naive variant that replaces the gradient with a stochastic gradient, again with and without an MH correction; (ii) the proposed SGHMC method, which does not use an MH correction. We use $\nabla\tilde{U}(\theta) = \nabla U(\theta) + \mathcal{N}(0, 4)$ in the stochastic gradient based samplers and $\epsilon = 0.1$ in all cases. Momentum is resampled every 50 steps in all variants of HMC.

## 4. Experiments

### 4.1. Simulated Scenarios

To empirically explore the behavior of HMC using exact gradients relative to stochastic gradients, we conduct experiments on a simulated setup. As a baseline, we consider the standard HMC implementation of Alg. 1, both with and without the MH correction. We then compare to HMC with stochastic gradients, replacing $\nabla U$ in Alg. 1 with $\nabla\tilde{U}$, and consider this proposal with and without an MH correction. Finally, we compare to our proposed SGHMC, which does not use an MH correction. Fig. 1 shows the empirical distributions generated by the different sampling algorithms. We see that even without an MH correction, both the HMC and SGHMC algorithms provide results close to the true distribution, implying that any errors from considering non-zero $\epsilon$ are negligible. On the other hand, the results of naïve stochastic gradient HMC diverge significantly from the truth unless an MH correction is added. These findings validate our theoretical results; that is, both standard HMC and SGHMC maintain $\pi$ as the invariant distribution as $\epsilon \to 0$ whereas naïve stochastic gradient HMC does not, though this can be corrected for using a (costly) MH step.

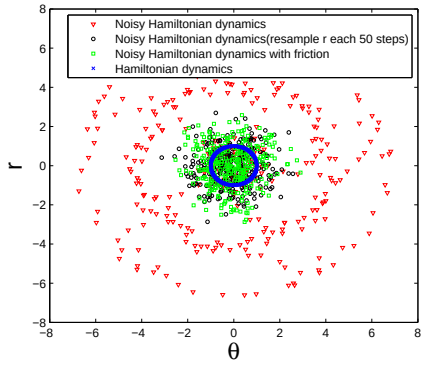


*Figure 2.* Points $(\theta, r)$ simulated from discretizations of various Hamiltonian dynamics over 15000 steps using $U(\theta) = \frac{1}{2}\theta^2$ and $\epsilon = 0.1$. For the noisy scenarios, we replace the gradient by $\nabla\tilde{U}(\theta) = \theta + \mathcal{N}(0, 4)$. We see that noisy Hamiltonian dynamics lead to diverging trajectories when friction is not introduced. Resampling $r$ helps control divergence, but the associated HMC stationary distribution is not correct, as illustrated in Fig. 1.

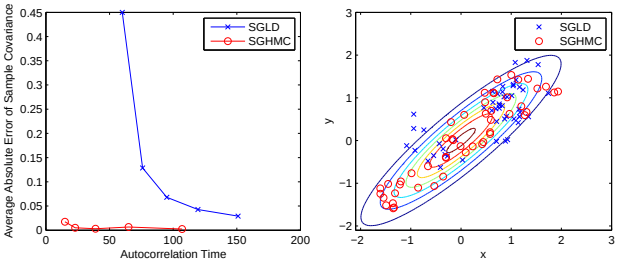


*Figure 3.* Contrasting sampling of a bivariate Gaussian with correlation using SGHMC versus SGLD. Here, $U(\theta) = \frac{1}{2}\theta^T\Sigma^{-1}\theta$, $\nabla\tilde{U}(\theta) = \Sigma^{-1}\theta + \mathcal{N}(0, I)$ with $\Sigma_{11} = \Sigma_{22} = 1$ and correlation $\rho = \Sigma_{12} = 0.9$. *Left:* Mean absolute error of the covariance estimation using ten million samples versus autocorrelation time of the samples as a function of 5 step size settings. *Right:* First 50 samples of SGHMC and SGLD.

We also consider simply simulating from the discretized Hamiltonian dynamical systems associated with the various samplers compared. In Fig. 2, we compare the resulting trajectories and see that the path of $(\theta, r)$ from the noisy system *without* friction diverges significantly. The modification of the dynamical system by adding friction (corresponding to SGHMC) corrects this behavior. We can also correct for this divergence through periodic resampling of the momentum, though as we saw in Fig. 1, the corresponding MCMC algorithm ("Naive stochastic gradient HMC (no MH)") does not yield the correct target distribution. These results confirm the importance of the friction term in maintaining a well-behaved Hamiltonian *and* leading to the correct stationary distribution.

It is known that a benefit of HMC over many other MCMC algorithms is the efficiency in sampling from correlated distributions (Neal, 2010)—this is where the introduction of the momentum variable shines. SGHMC inherits this property. Fig. 3 compares SGHMC and SGLD (Welling & Teh, 2011) when sampling from a bivariate Gaussian with positive correlation. For each method, we examine five different settings of the initial step size on a linearly decreasing scale and generate ten million samples. For each of these sets of samples (one set per step-size setting), we calculate the autocorrelation time[2] of the samples and the average absolute error of the resulting sample covariance. Fig. 3(a) shows the autocorrelation versus estimation error for the five settings. As we decrease the stepsize, SGLD has reasonably low estimation error but high autocorrelation time indicating an inefficient sampler. In contrast, SGHMC achieves even lower estimation error at very low autocorrelation times, from which we conclude that the sampler is indeed efficiently exploring the distribution. Fig. 3(b) shows the first 50 samples generated by the two samplers. We see that SGLD's random-walk behavior makes it challenging to explore the tails of the distribution. The momentum variable associated with SGHMC instead drives the sampler to move along the distribution contours.

### 4.2. Bayesian Neural Networks for Classification

We also test our method on a handwritten digits classification task using the MNIST dataset[3]. The dataset consists of 60,000 training instances and 10,000 test instances. We randomly split a validation set containing 10,000 instances from the training data in order to select training parameters, and use the remaining 50,000 instances for training. For classification, we consider a two layer Bayesian neural network with 100 hidden variables using a sigmoid unit and an output layer using softmax. We tested four methods: SGD, SGD with momentum, SGLD and SGHMC. For the optimization-based methods, we use the validation set to select the optimal regularizer $\lambda$ of network weights[4]. For the sampling-based methods, we take a fully Bayesian approach and place a weakly informative gamma prior on each layer's weight regularizer $\lambda$. The sampling procedure is carried out by running SGHMC and SGLD using mini-batches of 500 training instances, then resampling hyperparameters after an entire pass over the training set. We run the samplers for 800 iterations (each over the entire training dataset) and discard the initial 50 samples as burn-in.

The test error as a function of MCMC or optimization iteration (after burn-in) is reported for each of these methods in Fig. 4. From the results, we see that SGD with momentum converges faster than SGD. SGHMC also has an advantage over SGLD, converging to a low test error much more rapidly. In terms of runtime, in this case the gra-

[2] Autocorrelation time is defined as $1 + \sum_{s=1}^{\infty} \rho_s$, where $\rho_s$ is the autocorrelation at lag $s$.

[3] http://yann.lecun.com/exdb/mnist/

[4] We also tried MAP inference for selecting $\lambda$ in the optimization-based method, but found similar performance.

So, we get 750 weight samples? Do we know how well these approx the posterior? If we'd like to approx the post

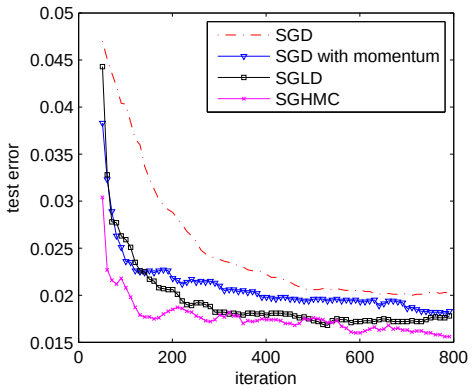


*Figure 4.* Convergence of test error on the MNIST dataset using SGD, SGD with momentum, SGLD, and SGHMC to infer model parameters of a Bayesian neural net.

dient computation used in backpropagation dominates so both have the same computational cost. The final results of the sampling based methods are better than optimization-based methods, showing an advantage to Bayesian inference in this setting, thus validating the need for scalable and efficient Bayesian inference algorithms such as SGHMC.

### 4.3. Online Bayesian Probabilistic Matrix Factorization for Movie Recommendations

Collaborative filtering is an important problem in web applications. The task is to predict a user's preference over a set of items (e.g., movies, music) and produce recommendations. Probabilistic matrix factorization (PMF) (Salakhutdinov & Mnih, 2008b) has proven effective for this task. Due to the sparsity in the ratings matrix (users versus items) in recommender systems, over-fitting is a severe issue with Bayesian approaches providing a natural solution (Salakhutdinov & Mnih, 2008a).

We conduct an experiment in *online* Bayesian PMF on the Movielens dataset ml-1M[5]. The dataset contains about 1 million ratings of 3,952 movies by 6,040 users. The number of latent dimensions is set to 20. In comparing our stochastic-gradient-based approaches, we use minibatches of 4,000 ratings to update the user and item latent matrices. We choose a significantly larger minibatch size in this application than that of the neural net because of the dramatically larger parameter space associated with the PMF model. For the optimization-based approaches, the hyperparameters are set using cross validation (again, we did not see a performance difference from considering MAP estimation). For the sampling-based approaches, the hyperparameters are updated using a Gibbs step after every $2,000$ steps of sampling model parameters. We run the sampler to generate 2,000,000 samples, with the first 100,000 samples discarded as burn-in. We use five-fold cross validation to evaluate the performance of the different methods.

[5]http://grouplens.org/datasets/movielens/

*Table 1.* Predictive RMSE estimated using 5-fold cross validation on the Movielens dataset for various approaches of inferring parameters of a Bayesian probabilistic matrix factorization model.

| METHOD | RMSE |
|---|---|
| SGD | $0.8538 \pm 0.0009$ |
| SGD WITH MOMENTUM | $0.8539 \pm 0.0009$ |
| SGLD | $0.8412 \pm 0.0009$ |
| SGHMC | $0.8411 \pm 0.0011$ |

The results are shown in Table 1. Both SGHMC and SGLD give better prediction results than optimization-based methods. In this experiment, the results for SGLD and SGHMC are very similar. We also observed that the per-iteration running time of both methods are comparable. As such, the experiment suggests that SGHMC is an effective candidate for online Bayesian PMF.

## 5. Conclusion

Moving between modes of a distribution is one of the key challenges for MCMC-based inference algorithms. To address this problem in the large-scale or online setting, we proposed SGHMC, an efficient method for generating high-quality, "distant" steps in such sampling methods. Our approach builds on the fundamental framework of HMC, but using stochastic estimates of the gradient to avoid the costly full gradient computation. Surprisingly, we discovered that the natural way to incorporate stochastic gradient estimates into HMC can lead to divergence and poor behavior both in theory and in practice. To address this challenge, we introduced second-order Langevin dynamics with a friction term that counteracts the effects of the noisy gradient, maintaining the desired target distribution as the invariant distribution of the continuous system. Our empirical results, both in a simulated experiment and on real data, validate our theory and demonstrate the practical value of introducing this simple modification. A natural next step is to explore combining adaptive HMC techniques with SGHMC. More broadly, we believe that the unification of efficient optimization and sampling techniques, such as those described herein, will enable a significant scaling of Bayesian methods.

## Acknowledgements

This work was supported in part by the TerraSwarm Research Center sponsored by MARCO and DARPA, ONR Grant N00014-10-1-0746, DARPA Grant FA9550-12-1-0406 negotiated by AFOSR, NSF IIS-1258741 and Intel ISTC Big Data. We also appreciate the discussions with Mark Girolami, Nick Foti, Ping Ao and Hong Qian.

## References


Ahn, S., Korattikara, A., and Welling, M. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*, pp. 1591–1598, July 2012.

Bardenet, R., Doucet, A., and Holmes, C. Towards scaling up Markov chain Monte Carlo: An adaptive subsampling approach. In *Proceedings of the 30th International Conference on Machine Learning (ICML'14)*, volume 32, pp. 405–413, February 2014.

Duane, S., Kennedy, A.D., Pendleton, B.J., and Roweth, D. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216 – 222, 1987.

Girolami, M. and Calderhead, B. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society Series B*, 73(2):123–214, 03 2011.

Hoffman, M.D. and Gelman, A. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *arXiv*, 1111.4246, 2011.

Hoffman, M.D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Maching Learning Research*, 14(1):1303–1347, May 2013.

Horowitz, A.M. A generalized guided Monte Carlo algorithm. *Physics Letters B*, 268(2):247 – 252, 1991.

Korattikara, A., Chen, Y., and Welling, M. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proceedings of the 30th International Conference on Machine Learning (ICML'14)*, volume 32, pp. 181–189, February 2014.

Levin, D.A., Peres, Y., and Wilmer, E.L. *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

Neal, R.M. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems 5 (NIPS'93)*, pp. 475–482, 1993.

Neal, R.M. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4): 1574–1609, January 2009.

Patterson, S. and Teh, Y.W. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems 26 (NIPS'13)*, pp. 3102–3110. 2013.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3): 400–407, 09 1951.

Salakhutdinov, R. and Mnih, A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*, pp. 880–887, 2008a.

Salakhutdinov, R. and Mnih, A. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20 (NIPS'08)*, pp. 1257–1264, 2008b.

Shi, J., Chen, T., Yuan, R., Yuan, B., and Ao, P. Relation of a new interpretation of stochastic differential equations to Ito process. *Journal of Statistical Physics*, 148(3): 579–590, 2012.

Sutskever, I., Martens, J., Dahl, G. E., and Hinton, G. E. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*, volume 28, pp. 1139–1147, May 2013.

Wang, M.C. and Uhlenbeck, G.E. On the Theory of the Brownian Motion II. *Reviews of Modern Physics*, 17(2-3):323, 1945.

Wang, Z., Mohamed, S., and Nando, D. Adaptive Hamiltonian and Riemann manifold Monte Carlo. In *Proceedings of the 30th International Conference on Machine Learning (ICML'13)*, volume 28, pp. 1462–1470, May 2013.

Welling, M. and Teh, Y.W. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*, pp. 681–688, June 2011.

Yin, L. and Ao, P. Existence and construction of dynamical potential in nonequilibrium processes without detailed balance. *Journal of Physics A: Mathematical and General*, 39(27):8593, 2006.

# Supplementary Material

## A. Background on Fokker-Planck Equation

The Fokker-Planck equation (FPE) associated with a given stochastic differential equation (SDE) describes the time evolution of the distribution on the random variables under the specified stochastic dynamics. For example, consider the SDE:

$$dz = g(z)dt + \mathcal{N}(0, 2D(z)dt), \tag{16}$$

where $z \in \mathbb{R}^n, g(z) \in \mathbb{R}^n, D(z) \in \mathbb{R}^{n\times n}$. The distribution of $z$ governed by Eq. (16) (denoted by $p_t(z)$), evolves under the following equation

$$\partial_t p_t(z) = -\sum_{i=1}^{n} \partial_{z_i}[g_i(z)p_t(z)] + \sum_{i=1}^{n}\sum_{j=1}^{n} \partial_{z_i}\partial_{z_j}[D_{ij}(z)p_t(z)].$$

Here $g_i(z)$ is the $i$-th entry of vector $g(z)$ and $D_{ij}(z)$ is the $(i, j)$ entry of the matrix $D$. In the dynamics considered in this paper, $z = (\theta, r)$ and

$$D = \begin{bmatrix} 0 & 0 \\ 0 & B(\theta) \end{bmatrix}. \tag{17}$$

That is, the random variables are momentum $r$ and position $\theta$, with noise only added to $r$ (though dependent upon $\theta$). The FPE can be written in the following compact form:

$$\partial_t p_t(z) = -\nabla^T[g(z)p_t(z)] + \nabla^T[D(z)\nabla p_t(z)], \tag{18}$$

where $\nabla^T[g(z)p_t(z)] = \sum_{i=1}^{n} \partial_{z_i}[g_i(z)p_t(z)]$ , and

$$\begin{aligned}
&\nabla^T[D\nabla p_t(\theta, r)] \\
&= \sum_{ij} \partial_{z_i}[D_{ij}(z)\partial_{z_j} p_t(z)] \\
&= \sum_{ij} \partial_{z_i}[D_{ij}(z)\partial_{z_j} p_t(z)] + \sum_{ij} \partial_{z_i}[(\partial_{z_j} D_{ij}(z))p_t(z)] \\
&= \sum_{ij} \partial_{z_i}\partial_{z_j}[D_{ij}(z)p_t(z)].
\end{aligned}$$

Note that $\partial_{z_j} D_{ij}(z) = 0$ for all $i, j$, since $\partial_{r_j} B_{ij}(\theta) = 0$ (the noise is only added to $r$ and only depends on parameter $\theta$).

## B. Proof of Theorem 3.1

Let $G = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$ and $D = \begin{bmatrix} 0 & 0 \\ 0 & B(\theta) \end{bmatrix}$. The noisy Hamiltonian dynamics of Eq. (7) can be written as

$$\begin{aligned}
d\begin{bmatrix} \theta \\ r \end{bmatrix} &= -\begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}\begin{bmatrix} \nabla U(\theta) \\ M^{-1}r \end{bmatrix} dt + \mathcal{N}(0, 2Ddt) \\
&= -G\nabla H(\theta, r)dt + \mathcal{N}(0, 2Ddt).
\end{aligned}$$

Applying Eq. (18), defining $g(z) = -G\nabla H$), the corresponding FPE is given by

$$\partial_t p_t(\theta, r) = \nabla^T[G\nabla H(\theta, r)p_t(\theta, r)] + \nabla^T[D\nabla p_t(\theta, r)]. \tag{19}$$

We use $z = (\theta, r)$ to denote the joint variable of position and momentum. The entropy is defined by $h(p_t(\theta, r)) = -\int_{\theta,r} f(p_t(\theta, r))d\theta dr$. Here $f(x) = x \ln x$ is a strictly convex function defined on $(0, +\infty)$. The evolution of the entropy is governed by

$$\begin{aligned}
\partial_t h(p_t(z)) =& \partial_t \int_z f(p_t(z))dz \\
=& -\int_z f'(p_t(z))\partial_t p_t(z)dz \\
=& -\int_z f'(p_t(z))\nabla^T[G\nabla H(z)p_t(z)]dz \\
&-\int_z f'(p)\nabla^T[D(z)\nabla p_t(z)]dz.
\end{aligned}$$

The entropy evolution can be described as the sum of two parts: the noise-free Hamiltonian dynamics and the stochastic gradient noise term. The Hamiltonian dynamics part does not change the entropy, since

$$\begin{aligned}
&-\int_z f'(p_t(z))\nabla^T[G\nabla H(z)p_t]dz \\
&= -\int_z f'(p_t(z))\nabla^T[G\nabla H(z)]p_t dz \\
&\quad -\int_z f'(p_t(z))(\nabla p_t(z))^T[G\nabla H(z)]dz \\
&= -\int_z (\nabla f(p_t(z)))^T[G\nabla H(z)]dz \\
&= \int_z f(p_t(z))\nabla^T[G\nabla H(z)]dz = 0.
\end{aligned}$$

In the second equality, we use the fact that $\nabla^T[G\nabla H(z)] = -\partial_\theta\partial_r H + \partial_r\partial_\theta H = 0$. The last equality is given by integration by parts, using the assumption that the probability density vanishes at infinity and $f(x) \to 0$ as $x \to 0$ such that $f(p_t(z))[G\nabla H(z)] \to 0$ as $z \to \infty$.

The contribution due to the stochastic gradient noise can be calculated as

$$\begin{aligned}
&-\int_z f'(p_t(z))\nabla^T[D(z)\nabla p_t(z)]dz \\
&= \int_z (f''(p_t(z))\nabla p_t(z))^T D(z)\nabla p_t(z)dz \\
&= \int_{\theta,r} f''(p_t(z))(\nabla_r p_t(\theta, r))^T B(\theta)\nabla_r p_t(\theta, r)d\theta dr.
\end{aligned}$$

The first equality is again given by integration by parts, assuming that the gradient of $p_t$ vanishes at infinity faster than $\frac{1}{\ln p_t(z)}$. That is, $f'(p_t(z))\nabla p_t(z) = (1 + \ln p_t(z))\nabla p_t(z) \to 0$ such that $f'(p_t(z))[D(z)\nabla p_t(z)] \to 0$ as $z \to \infty$. The statement of Theorem 3.1 immediately follows.

## C. Proof of Corollary 3.1

Assume $\pi(\theta, r) = \exp(-H(\theta, r))/Z$ is invariant under Eq. (7) and is a well-behaved distribution such that

$H(\theta, r) \to \infty$ as $\|\theta\|, \|r\| \to \infty$. Then it is straightforward to verify that $\pi(\theta, r)$ and $\ln \pi(\theta, r) \nabla \pi(\theta, r) = \frac{1}{Z} \exp\left(-H(\theta, r)\right) \nabla H^2(\theta, r)$ vanish at infinity, such that $\pi$ satisfies the conditions of Theorem 3.1. We also have $\nabla_r \pi(\theta, r) = \frac{1}{Z} \exp\left(-H(\theta, r)\right) M^{-1} r$. Using the assumption that the Fisher information matrix $B(\theta)$ has full rank, and noting that $f''(p) > 0$ for $p > 0$, from Eq. (8) of Theorem 3.1 we conclude that entropy increases over time: $\partial_t h(p_t(\theta, r))|_{p_t=\pi} > 0$. This contradicts that $\pi$ is the invariant distribution.

## D. FPE for Second-Order Langevin Dynamics

Second-order Langevin dynamics can be described by the following equation

$$\begin{aligned} d\begin{bmatrix} \theta \\ r \end{bmatrix} &= -\begin{bmatrix} 0 & -I \\ I & B \end{bmatrix} \begin{bmatrix} \nabla U(\theta) \\ M^{-1} r \end{bmatrix} dt + \mathcal{N}(0, 2\tau D dt) \\ &= -\left[D + G\right] \nabla H(\theta, r) dt + \mathcal{N}(0, 2\tau D dt), \end{aligned} \tag{20}$$

where $\tau$ is a temperature (usually set to 1). In this paper, we use the following compact form of the FPE to calculate the distribution evolution under Eq (20):

$$\partial_t p_t(\theta, r) = \nabla^T \{[D+G]\left[p_t(\theta, r) \nabla H(\theta, r) + \tau \nabla p_t(\theta, r)\right]\}. \tag{21}$$

To derive this FPE, we apply Eq. (18) to Eq (20), defining $g(z) = -(D + G)\nabla H$, which yields

$$\partial_t p_t(\theta, r) = \nabla^T \{[D+G]\left[\nabla H(\theta, r) p_t(\theta, r)\right]\} + \nabla^T \left[\tau D \nabla p_t(\theta, r)\right].$$

Using the fact that $\nabla^T \left[G \nabla p_t(\theta, r)\right] = -\partial_\theta \partial_r p_t(\theta, r) + \partial_r \partial_\theta p_t(\theta, r) = 0$, we get Eq. (21). This form of the FPE allows easy verification that the stationary distribution is given by $\pi(\theta, r) \propto e^{-\frac{1}{\tau} H(\theta, r)}$. In particular, if we substitute the target distribution into Eq. (21), we note that

$$\left[e^{-\frac{1}{\tau} H(\theta, r)} \nabla H(\theta, r) + \tau \nabla e^{-\frac{1}{\tau} H(\theta, r)} H(\theta, r)\right] = 0$$

such that $\partial_t \pi(\theta, r) = 0$, implying that $\pi$ is indeed the stationary distribution.

The compact form of Eq. (21) can also be used to construct other stochastic processes with the desired invariant distribution. A generalization of the FPE in Eq. (21) is given by Yin & Ao (2006). The system we have discussed in this paper considers cases where $G = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$ and $D$ only depends on $\theta$. In practice, however, it might be helpful to make $G$ depend on $\theta$ as well. For example, to make use of the Riemann geometry of the problem, as in Girolami & Calderhead (2011) and Patterson & Teh (2013), by adapting $G$ according to the local curvature. For us to consider these more general cases, a correction term needs to be added during simulation (Shi et al., 2012). With that correction term, we still maintain the desired target distribution as the stationary distribution.

## E. Reversibility of SGHMC Dynamics

The dynamics of SGHMC are not reversible in the conventional definition of reversibility. However, the dynamics satisfy the following property:

**Theorem E.1.** *Assume $P(\theta_t, r_t|\theta_0, r_0)$ is the distribution governed by dynamics in Eq. (20), i.e. $P(\theta_t, r_t|\theta_0, r_0)$ follows Eq. (21), then for $\pi(\theta, r) \propto \exp(-H(\theta, r))$,*

$$\pi(\theta_0, r_0) P(\theta_t, r_t|\theta_0, r_0) = \pi(\theta_t, -r_t) P(\theta_0, -r_0|\theta_t, -r_t). \tag{22}$$

*Proof.* Assuming $\pi$ is the stationary distribution and $P^*$ the reverse-time Markov process associated with $P$: $\pi(\theta_0, r_0) P(\theta_t, r_t|\theta_0, r_0) = \pi(\theta_t, r_t) P^*(\theta_0, r_0|\theta_t, r_t)$. Let $\mathcal{L}(p) = \nabla^T \{[D + G]\left[p \nabla H(\theta, r) + \tau \nabla p\right]\}$ be the generator of Markov process described by Eq. (21). The generator of the reverse process is given by $\mathcal{L}^*$, which is the adjoint operator of $\mathcal{L}$ in the inner-product space $l^2(\pi)$, with inner-product defined by $\langle p, q \rangle_\pi = E_{x \sim \pi(x)}[p(x) q(x)]$. We can verify that $\mathcal{L}^*(p) = \nabla^T \{[D - G]\left[p \nabla H(\theta, r) + \tau \nabla p\right]\}$. The corresponding SDE of the reverse process is given by

$$d\begin{bmatrix} \theta \\ r \end{bmatrix} = \left[D - G\right] \nabla H(\theta, r) + \mathcal{N}(0, 2\tau D dt),$$

which is equivalent to

$$d\begin{bmatrix} \theta \\ -r \end{bmatrix} = \left[D + G\right] \nabla H(\theta, -r) + \mathcal{N}(0, 2\tau D dt).$$

This means $P^*(\theta_0, r_0|\theta_t, r_t) = P(\theta_0, -r_0|\theta_t, -r_t)$. Recalling that we assume Gaussian momentum, $r$, centered about 0, we also have $\pi(\theta, r) = \pi(\theta, -r)$. Together, we then have

$$\begin{aligned} \pi(\theta_0, r_0) P(\theta_t, r_t|\theta_0, r_0) &= \pi(\theta_t, r_t) P^*(\theta_0, r_0|\theta_t, r_t) \\ &= \pi(\theta_t, -r_t) P(\theta_0, -r_0|\theta_t, -r_t). \end{aligned}$$

□

Theorem E.1 is not strictly detailed balance by the conventional definition since $\mathcal{L}^* \neq \mathcal{L}$ and $P^* \neq P$. However, it can be viewed as a kind of time reversibility. When we reverse time, the sign of speed needs to be reversed to allow backward travel. This property is shared by the noise-free HMC dynamics of (Neal, 2010). Detailed balance can be enforced by the symmetry of $r$ during the re-sampling step. However, we note that we do not rely on detailed balance to have $\pi$ be the stationary distribution of our noisy Hamiltonian with friction (see Eq. (9)).

## F. Convergence Analysis

In the paper, we have discussed that the efficiency of SGHMC decreases as the step size $\epsilon$ decreases. In practice, we usually want to trade a small amount of error for

efficiency. In the case of SGHMC, we are interested in a small, nonzero $\epsilon$ and fast approximation of $B$ given by $\hat{B}$. In this case, even under the continuous dynamics, the sampling procedure contains error that relates to $\epsilon$ due to inaccurate estimation of $B$ with $\hat{B}$. In this section, we investigate how the choice of $\epsilon$ can be related to the error in the final stationary distribution. The sampling procedure with inaccurate estimation of $B$ can be described with the following dynamics

$$\begin{cases} d\theta = M^{-1}r\ dt \\ dr = -\nabla U(\theta)\ dt - CM^{-1}r dt + \mathcal{N}(0, 2(C+\delta S)dt). \end{cases}$$

Here, $\delta S = B - \hat{B}$ is the error term that is not considered by the sampling algorithm. Assume the setting where $\hat{B} = 0$, then we can let $\delta = \epsilon$ and $S = \frac{1}{2}V$. Let $\tilde{\pi}$ be the stationary distribution of the dynamics. In the special case when $V = C$, we can calculate $\tilde{\pi}$ exactly by

$$\tilde{\pi}(\theta, r) \propto \exp\left(-\frac{1}{1+\delta}H(\theta, r)\right). \tag{23}$$

This indicates that for small $\epsilon$, our stationary distribution is indeed close to the true stationary distribution. In general case, we consider the FPE of the distribution of this SDE, given by

$$\partial_t \tilde{p}_t(\theta, r) = [\mathcal{L} + \delta\mathcal{S}]\tilde{p}_t(\theta, r). \tag{24}$$

Here, $\mathcal{L}(p) = \nabla^T\{[D+G]\,[p\nabla H(\theta, r) + \nabla p]\}$ is the operator corresponds to correct sampling process. Let the operator $\mathcal{S}(p) = \nabla_r[S\nabla_r p]$ correspond to the error term introduced by inaccurate $\hat{B}$. Let us consider the $\chi^2$-divergence defined by

$$\chi^2(p, \pi) = E_{x\sim\pi}\left[\frac{(p(x) - \pi(x))^2}{\pi^2(x)}\right] = E_{x\sim\pi}\left[\frac{p^2(x)}{\pi^2(x)}\right] - 1,$$

which provides a measure of distance between the distribution $p$ and the true distribution $\pi$. Theorem F.1 shows that the $\chi^2$-divergence decreases as $\delta$ becomes smaller.

**Theorem F.1.** *Assume $p_t$ evolves according to $\partial_t p_t = \mathcal{L}p_t$, and satisfies the following mixing rate $\lambda$ with respect to $\chi^2$ divergence at $\tilde{\pi}$: $\partial_t\chi^2(p_t, \pi)|_{p_t=\tilde{\pi}} \leq -\lambda\chi^2(\tilde{\pi}, \pi)$. Further assume the process governed by $\mathcal{S}$ ($\partial_t q_t = \mathcal{S}q_t$) has bounded divergence change $|\partial_t\chi^2(q_t, \pi)| < c$. Then $\tilde{\pi}$ satisfies*

$$\chi^2(\tilde{\pi}, \pi) < \frac{\delta c}{\lambda}. \tag{25}$$

*Proof.* Consider the divergence change of $\tilde{p}$ governed by Eq.(24). It can be decomposed into two components, the change of divergence due to $\mathcal{L}$, and the change of divergence due to $\delta\mathcal{S}$

$$\begin{aligned} \partial_t\chi^2(\tilde{p}_t, \pi) =& E_{x\sim\pi}\left[\frac{\tilde{p}(x)}{\pi^2(x)}[\mathcal{L} + \delta\mathcal{S}]\tilde{p}_t(x)\right] \\ =& E_{x\sim\pi}\left[\frac{\tilde{p}_t(x)}{\pi^2(x)}\mathcal{L}\tilde{p}_t(x)\right] + \delta E_{x\sim\pi}\left[\frac{\tilde{p}(x)}{\pi^2(x)}\mathcal{S}\tilde{p}_t(x)\right] \\ =& \partial_t\chi^2(p_t, \pi)|_{p_t=\tilde{p}_t} + \delta\partial_t\chi^2(q_t, \pi)|_{q_t=\tilde{p}_t}. \end{aligned}$$

We then evaluate the above equation at the stationary distribution of the inaccurate dynamics $\tilde{\pi}$. Since $\partial_t\chi^2(\tilde{p}_t, \pi)|_{\tilde{p}=\tilde{\pi}} = 0$, we have

$$\lambda\chi^2(\tilde{\pi}, \pi) = \delta\left|(\partial_t\chi^2(q_t, \pi)|_{q_t=\tilde{\pi}})\right| < \delta c.$$

□

This theorem can also be used to measure the error in SGLD, and justifies the use of small finite step sizes in SGLD. We should note that the mixing rate bound $\lambda$ at $\tilde{\pi}$ exists for SGLD and can be obtained using spectral analysis (Levin et al., 2008), but the corresponding bounds for SGHMC are unclear due to the irreversibility of the process. We leave this for future work.

Our proof relies on a contraction bound relating the error in the transition distribution to the error in the final stationary distribution. Although our argument is based on a continuous-time Markov process, we should note that a similar guarantee can also be proven in terms of a discrete-time Markov transition kernel. We refer the reader to (Korattikara et al., 2014) and (Bardenet et al., 2014) for further details.

## G. Setting SGHMC Parameters

As we discussed in Sec. 3.3, we can connect SGHMC with SGD with momentum by rewriting the dynamics as (see Eq.(15))

$$\begin{cases} \Delta\theta = v \\ \Delta v = -\eta\nabla\tilde{U}(x) - \alpha v + \mathcal{N}(0, 2(\alpha - \hat{\beta})\eta). \end{cases}$$

In analogy to SGD with momentum, we call $\eta$ the learning rate and $1 - \alpha$ the momentum term. This equivalent update rule is cleaner and we recommend parameterizing SGHMC in this form.

The $\hat{\beta}$ term corresponds to the estimation of noise that comes from the gradient. One simple choice is to ignore the gradient noise by setting $\hat{\beta} = 0$ and relying on small $\epsilon$. We can also set $\hat{\beta} = \eta\hat{V}/2$, where $\hat{V}$ is estimated using empirical Fisher information as in (Ahn et al., 2012).

There are then three parameters: the learning rate $\eta$, momentum decay $\alpha$, and minibatch size $|\tilde{\mathcal{D}}|$. Define $\beta = \epsilon M^{-1}B = \frac{1}{2}\eta V(\theta)$ to be the exact term induced by introduction of the stochastic gradient. Then, we have

$$\beta = O\left(\eta\frac{|\mathcal{D}|}{|\tilde{\mathcal{D}}|}\mathcal{I}\right), \tag{26}$$

where $\mathcal{I}$ is fisher information matrix of the gradient, $|\mathcal{D}|$ is size of training data, $|\tilde{\mathcal{D}}|$ is size of minibatch, and $\eta$ is our learning rate. We want to keep $\beta$ small so that the resulting dynamics are governed by the user-controlled term and the

sampling algorithm has a stationary distribution close to the target distribution. From Eq. (26), we see that there is no free lunch here: as the training size gets bigger, we can either set a small learning rate $\eta = O(\frac{1}{|\mathcal{D}|})$ or use a bigger minibatch size $|\tilde{\mathcal{D}}|$. In practice, choosing $\eta = O(\frac{1}{|\mathcal{D}|})$ gives better numerical stability, since we also need to multiply $\eta$ by $\nabla\tilde{U}$, the mean of the stochastic gradient. Large $\eta$ can cause divergence, especially when we are not close to the mode of distribution. We note that the same discussion holds for SGLD (Welling & Teh, 2011).

In practice, we find that using a minibatch size of hundreds (e.g $|\tilde{\mathcal{D}}| = 500$) and fixing $\alpha$ to a small number (e.g. $0.01$ or $0.1$) works well. The learning rate can be set as $\eta = \gamma/|\mathcal{D}|$, where $\gamma$ is the "per-batch learning rate", usually set to $0.1$ or $0.01$. This method of setting parameters is also commonly used for SGD with momentum (Sutskever et al., 2013).

## H. Experimental Setup

### H.1. Bayesian Neural Network

The Bayesian neural network model used in Sec. 4.2 can be described by the following equation:

$$P(y = i|x) \propto \exp\left(A_i^T \sigma(B^T x + b) + a_i\right). \tag{27}$$

Here, $y \in \{1, 2, \cdots, 10\}$ is the output label of a digit. $A \in \mathbb{R}^{10\times 100}$ contains the weight for output layers and we use $A_i$ to indicate $i$-th column of $A$. $B \in \mathbb{R}^{d\times 100}$ contains the weight for the first layer. We also introduce $a \in \mathbb{R}^{10}$ and $b \in \mathbb{R}^{100}$ as bias terms in the model. In the MNIST dataset, the input dimension $d = 784$. We place a Gaussian prior on the model parameters

$$P(A) \propto \exp(-\lambda_A \|A\|^2), P(B) \propto \exp(-\lambda_B \|B\|^2)$$
$$P(a) \propto \exp(-\lambda_a \|a\|^2), P(b) \propto \exp(-\lambda_b \|a\|^2).$$

We further place gamma priors on each of the precision terms $\lambda$:

$$\lambda_A, \lambda_B, \lambda_a, \lambda_b \overset{i.i.d.}{\sim} \Gamma(\alpha, \beta).$$

We simply set $\alpha$ and $\beta$ to 1 since the results are usually insensitive to these parameters. We generate samples from the posterior distribution

$$P(\Theta|\mathcal{D}) \propto \prod_{y,x\in\mathcal{D}} P(y|x, \Theta)P(\Theta), \tag{28}$$

where parameter set $\Theta = \{A, B, a, b, \lambda_A, \lambda_B, \lambda_a, \lambda_b\}$. The sampling procedure is carried out by alternating the following steps:

- Sample weights from $P(A, B, a, b|\lambda_A, \lambda_B, \lambda_a, \lambda_b, \mathcal{D})$ using SGHMC or SGLD with minibatch of 500 instances. Sample for 100 steps before updating hyper-parameters.
- Sample $\lambda$ from $P(\lambda_A, \lambda_B, \lambda_a, \lambda_b|A, B, a, b)$ using a Gibbs step. Note that the posterior for $\lambda$ is a gamma distribution by conditional conjugacy.

We used the validation set to select parameters for the various methods we compare. Specifically, for SGD and SGLD, we tried step-sizes $\epsilon \in \{0.1, 0.2, 0.4, 0.8\} \times 10^{-4}$, and the best settings were found to be $\epsilon = 0.1 \times 10^{-4}$ for SGD and $\epsilon = 0.2 \times 10^{-4}$ for SGLD. We then further tested $\epsilon = 0.16 \times 10^{-4}$ and $\epsilon = 0.06 \times 10^{-4}$ for SGD, and found $\epsilon = 0.16 \times 10^{-4}$ gave the best result, thus we used this setting for SGD. For SGD with momentum and SGHMC, we fixed $\alpha = 0.01$ and $\hat{\beta} = 0$, and tried $\eta \in \{0.1, 0.2, 0.4, 0.8\} \times 10^{-5}$. The best settings were $\eta = 0.4 \times 10^{-5}$ for SGD with momentum, and $\eta = 0.2 \times 10^{-5}$ for SGHMC. For the optimization-based methods, we use tried regularizer $\lambda \in \{0, 0.1, 1, 10, 100\}$, and $\lambda = 1$ was found to give the best performance.

### H.2. Online Bayesian Probabilistic Matrix Factorization

The Bayesian probabilistic matrix factorization (BPMF) model used in Sec. 4.3 can be described as:

$$\begin{aligned} \lambda_U, \lambda_V, \lambda_a, \lambda_b &\overset{i.i.d.}{\sim} \text{Gamma}(1, 1) \\ U_{ki} \sim&\mathcal{N}(0, \lambda_U^{-1}), V_{kj} \sim \mathcal{N}(0, \lambda_V^{-1}), \\ a_i \sim&\mathcal{N}(0, \lambda_a^{-1}), b_i \sim \mathcal{N}(0, \lambda_b^{-1}) \\ Y_{ij}|U, V \sim&\mathcal{N}(U_i^T V_j + a_i + b_j, \tau^{-1}). \end{aligned} \tag{29}$$

The $U_i \in \mathbb{R}^d$ and $V_j \in \mathbb{R}^d$ are latent vectors for user $i$ and movie $j$, while $a_i$ and $b_j$ are bias terms. We use a slightly simplified model than the BPMF model considered in (Salakhutdinov & Mnih, 2008a), where we only place priors on precision variables $\lambda = \{\lambda_U, \lambda_V, \lambda_a, \lambda_b\}$. However, the model still benefits from Bayesian inference by integrating over the uncertainty in the crucial regularization parameter $\lambda$. We generate samples from the posterior distribution

$$P(\Theta|Y) \propto P(Y|\Theta)P(\Theta), \tag{30}$$

with the parameter set $\Theta = \{U, V, a, b, \lambda_U, \lambda_V, \lambda_a, \lambda_b\}$. The sampling procedure is carried out by alternating the followings

- Sample weights from $P(U, V, a, b|\lambda_U, \lambda_V, \lambda_a, \lambda_b, Y)$ using SGHMC or SGLD with a minibatch size of 4,000 ratings. Sample for $2,000$ steps before updating the hyper-parameters.
- Sample $\lambda$ from $P(\lambda_U, \lambda_V, \lambda_a, \lambda_b|U, V, a, b)$ using a Gibbs step.

The training parameters for this experiment were directly selected using cross-validation. Specifically, for SGD and

SGLD, we tried step-sizes $\epsilon \in \{0.1, 0.2, 0.4, 0.8, 1.6\} \times 10^{-5}$, and the best settings were found to be $\epsilon = 0.4 \times 10^{-5}$ for SGD and $\epsilon = 0.8 \times 10^{-5}$ for SGLD. For SGD with momentum and SGHMC, we fixed $\alpha = 0.05$ and $\hat{\beta} = 0$, and tried $\eta \in \{0.1, 0.2, 0.4, 0.8\} \times 10^{-6}$. The best settings were $\eta = 0.4 \times 10^{-6}$ for SGD with momentum, and $\eta = 0.4 \times 10^{-6}$ for SGHMC.