3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–10

Exact Langevin Dynamics with Stochastic Gradients

Adrià Garriga-Alonso AG919@CAM.AC.UK
University of Cambridge, United Kingdom
Vincent Fortuin FORTUIN@INF.ETHZ.CH
ETH Zürich, Switzerland

Abstract

Stochastic gradient Markov Chain Monte Carlo algorithms are popular samplers for approximate inference, but they are generally biased. We show that many recent versions of these methods (e.g. Chen et al. (2014)) cannot be corrected using Metropolis-Hastings rejection sampling, because their acceptance probability is always zero. We can fix this by employing a sampler with realizable backwards trajectories, such as Gradient-Guided Monte Carlo (Horowitz, 1991), which generalizes stochastic gradient Langevin dynamics (Welling and Teh, 2011) and Hamiltonian Monte Carlo. We show that this sampler can be used with stochastic gradients, yielding nonzero acceptance probabilities which can be computed even across multiple steps.

1. Introduction

Exact posterior inference in many Bayesian models is intractable, which has led to the development of different approximate inference schemes. One popular class of such algorithms are Markov Chain Monte Carlo (MCMC) methods. These simulate a Markov chain, constructed so that its stationary distribution is the true posterior (Geyer, 1992). Among the MCMC methods, Hamiltonian Monte Carlo (HMC) has achieved recognition as being the "gold standard", mostly for its favorable convergence properties (Neal, 2012).

While HMC yields great performance, its adoption in many real applications is hindered by the fact that it requires passes over the whole data set to compute one gradient step, which is prohibitively expensive in many settings (Chen et al., 2014). Especially for large-data Bayesian models such as Bayesian neural networks (BNNs), stochastic gradient MCMC (SG-MCMC) methods are more popular, due to their favorable computational costs and their similarity to stochastic gradient descent (SGD). Arguably, the most popular SG-MCMC method for BNNs in recent years has been stochastic gradient HMC (Chen et al., 2014, SGHMC), although its predecessor stochastic gradient Langevin dynamics (Welling and Teh, 2011, SGLD) also enjoys considerable following.

Since all of these approaches are approximate, in order to ensure that they sample from the true posterior, rejection sampling is often necessary, for instance, following the Metropolis-Hastings (MH) scheme (Metropolis et al., 1953; Hastings, 1970). However, in this work, we draw attention to the unfortunate fact that SGHMC does not allow for MH rejection sampling, since its acceptance probability is always zero. We then revisit gradient-guided Monte Carlo (GGMC) (Horowitz, 1991), which can overcome this problem. We show

© A. Garriga-Alonso & V. Fortuin.

that GGMC provides a unified framework for SGLD and SGHMC and that it yields nonzero acceptance probabilities, which can even be computed across multiple steps.

Our contributions are as follows:

- We show that the SG-MCMC schemes used in recent work (Chen et al., 2014; Wenzel et al., 2020) always have acceptance probability zero, because the backward trajectory is not realizable in the symplectic Euler-Maruyama integrator they employ (Sec. 3).
- We revisit GGMC, a variant of HMC. It generalises HMC and SGLD, always has positive acceptance probability and, with correction, accepts stochastic gradients (Sec. 4). The scheme was independently discovered in the statistics (Neal, 2012; Horowitz, 1991) and molecular dynamics literatures (Bussi and Parrinello, 2007; Leimkuhler and Matthews, 2015). The latter have shown that it simulates Langevin dynamics.
- We examine an augmentation scheme to calculate the acceptance probabilities across multiple stochastic steps (Sec. 5, by Zhang et al. (2020)). This allows us to use stochastic gradients, but sample from the exact posterior, by doing one MH correction using the exact likelihood after many stochastic gradient steps.

Based on our findings, we believe that work in Bayesian deep learning should discard the Euler-Maruyama scheme used by SGHMC and its variants (Ding et al., 2014; Ma et al., 2015). Instead, to have confidence in our sampling, we should use reversible integrators (Matthews and Weare, 2018; Leimkuhler et al., 2019; Zhang et al., 2020, this work) and keep track of their acceptance probability.

2. Background: Hamiltonian Monte Carlo

We wish to sample from a target distribution $\pi(\boldsymbol{\theta}) \propto \tilde{\pi}(\boldsymbol{\theta})$, which we can only evaluate up to a normalizing constant. HMC first augments the parameter space $\boldsymbol{\theta} \in \mathbb{R}^d$ with a momentum $\boldsymbol{m} \in \mathbb{R}^d$ of the same dimension. It constructs a Markov chain that alternates: re-sampling the momentum, and then following an ordinary differential equation (ODE), specified by the Hamiltonian $H(\boldsymbol{\theta}, \boldsymbol{m})$, for a given amount of time. The stationary distribution of this Markov chain (with exact simulation) is the Boltzmann distribution

$$\bar{P}(\boldsymbol{\theta}, \boldsymbol{m}) \propto \exp\left(-\frac{1}{T}H(\boldsymbol{\theta}, \boldsymbol{m})\right) = \exp\left(-\frac{1}{T}U(\boldsymbol{\theta})\right)\exp\left(-\frac{1}{T}K(\boldsymbol{m})\right). \tag{1}$$

Here $U(\boldsymbol{\theta})$ is the potential energy, which we relate to the unnormalised target distribution by $U(\boldsymbol{\theta}) = -\log\tilde{\pi}(\boldsymbol{\theta})$. The kinetic energy $K(\boldsymbol{m}) = \frac{1}{2}\boldsymbol{m}^\mathsf{T}\boldsymbol{M}^{-1}\boldsymbol{m}$ makes the momenta $\boldsymbol{m}$ Gaussian, and thus easy to sample. $\boldsymbol{M}$ is a positive definite (usually diagonal) preconditioner, the mass matrix. $T \in \mathbb{R}^+$ is a positive temperature. For HMC, $T = 1$, but this changes in the next section. By construction, the tempered target $\tilde{\pi}(\boldsymbol{\theta})^{1/T}$ is a marginal of the Boltzmann distribution. We may write the ODE's full state as $\boldsymbol{s} = (\boldsymbol{\theta}, \boldsymbol{m})$.

Unfortunately, we cannot exactly simulate the ODE for most posteriors, so we must discretise it with an integrator. This unavoidably introduces errors into the simulation, which have to be controlled if we want to guarantee that the sampling is accurate.

The standard way to do this is via the Metropolis-Hastings acceptance probability. For this, we define the transition kernel: $g(\boldsymbol{s}_* \mid \boldsymbol{s}_n) = 1$ if the approximate simulation starting at $\boldsymbol{s}_n = (\boldsymbol{\theta}_n, \boldsymbol{m}_n)$ ends at $\boldsymbol{s}_* = (\boldsymbol{\theta}_*, \boldsymbol{m}_*)$, otherwise 0.

Then, we accept the new sample (i.e. set $\boldsymbol{s}_{n+1} = \boldsymbol{s}_*$) with probability $\min(1, \mathrm{MH}(\boldsymbol{s}_* \mid \boldsymbol{s}_n))$, where

$$\mathrm{MH}(\boldsymbol{s}_* \mid \boldsymbol{s}_n) = \frac{\bar{P}(\boldsymbol{\theta}_*, \boldsymbol{m}_*)}{\bar{P}(\boldsymbol{\theta}_n, \boldsymbol{m}_n)} \frac{g(\boldsymbol{\theta}_n, -\boldsymbol{m}_n \mid \boldsymbol{\theta}_*, -\boldsymbol{m}_*)}{g(\boldsymbol{\theta}_*, \boldsymbol{m}_* \mid \boldsymbol{\theta}_n, \boldsymbol{m}_n)}, \tag{2}$$

Otherwise we reject the sample, that is, set $\boldsymbol{s}_{n+1} = \boldsymbol{s}_n$. We negate the momentum so the integrator applied backwards would retrace the same steps.[1] Afterwards, we are free to negate the momentum again without a rejection step, since $K(-\boldsymbol{m}) = K(\boldsymbol{m})$, so it leaves the target distribution (1) invariant (Neal, 2012, eq. 5.20). For regular HMC, this can be ignored because the momentum gets resampled immediately.

3. SG-MCMC acceptance probabilities are zero

All the Markov chains we consider in this paper are discretisations of Langevin's stochastic differential equations (SDEs) which, borrowing notation from Wenzel et al. (2020), are

$$\mathrm{d}\boldsymbol{\theta} = \boldsymbol{M}^{-1}\boldsymbol{m}\,\mathrm{d}t \tag{3}$$

$$\mathrm{d}\boldsymbol{m} = -\gamma\boldsymbol{m}\,\mathrm{d}t - \nabla_{\boldsymbol{\theta}}U(\boldsymbol{\theta})\,\mathrm{d}t + \boldsymbol{M}^{1/2}\sqrt{2\gamma T}\,\mathrm{d}\boldsymbol{W}. \tag{4}$$

Here, $\gamma \in \mathbb{R}^+$ is a friction coefficient and $\boldsymbol{W}$ is a standard Wiener process with covariance $\boldsymbol{I}$.

If $\gamma = 0$, these become Hamilton's equations, used for HMC. Setting $\mathrm{d}\boldsymbol{m} = 0$, we obtain the overdamped Langevin equation, which discretises to the Metropolis-adjusted Langevin algorithm (MALA; e.g. Roberts and Rosenthal (1998)).

For time step h, the symplectic Euler-Maruyama scheme in Chen et al. (2014, SGHMC) and Wenzel et al. (2020) simulates these SDEs as follows. Using noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$,

$$\boldsymbol{m}_{n+1} = (1 - h\gamma)\boldsymbol{m}_n - h\nabla_{\boldsymbol{\theta}}U(\boldsymbol{\theta}_n) + \sqrt{h}\boldsymbol{M}^{1/2}\sqrt{2\gamma T}\boldsymbol{\epsilon} \tag{5}$$

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + h\boldsymbol{M}^{-1}\boldsymbol{m}_{n+1}\,. \tag{6}$$

Like in HMC, it is the discretisation, not the SDE, that determines our simulation's Markov transition kernel g. Unfortunately, we will see that the backward transition is unrealizable, so its density $g(\boldsymbol{\theta}_n, -\boldsymbol{m}_n \mid \boldsymbol{\theta}_{n+1}, -\boldsymbol{m}_{n+1})$ is almost always zero. Because this density appears in the numerator of eq. (2), the resulting acceptance probability for the Euler-Maruyama scheme is zero as well.

Theorem 1 *For any starting point $(\boldsymbol{\theta}_n, \boldsymbol{m}_n)$ and step size $h > 0$, we sample $(\boldsymbol{\theta}_{n+1}, \boldsymbol{m}_{n+1})$ using equations (5, 6). Then, the backward transition density $g(\boldsymbol{\theta}_n, -\boldsymbol{m}_n \mid \boldsymbol{\theta}_{n+1}, -\boldsymbol{m}_{n+1})$ is zero, with probability 1 over the randomness of the forward transition.*

1. Hamiltonian dynamics are time-reversible. We also need to use a time-reversible integrator, like the leapfrog or velocity Verlet, so the ratio of g in eq. (2) is 1.

Proof Consider the transition distribution, $g(\boldsymbol{\theta}_*, \boldsymbol{m}_* \,|\, \boldsymbol{\theta}, \boldsymbol{m})$. Because it has been generated from equations (5, 6), any realization from g must satisfy them. In particular, it has to satisfy eq. (6), i.e. $\boldsymbol{\theta}_* = \boldsymbol{\theta} + h\boldsymbol{M}^{-1}\boldsymbol{m}_*$. Thus, for the purposes of the acceptance probability (eq. 2), we can define the density of g as zero outside of its support,

$$g(\boldsymbol{\theta}_*, \boldsymbol{m}_* \,|\, \boldsymbol{\theta}, \boldsymbol{m}) = \begin{cases} \mathcal{N}(\boldsymbol{m}_* \,|\, (1-h\gamma)\boldsymbol{m} - h\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}), h\boldsymbol{M}2\gamma T) & \text{if } \boldsymbol{\theta}_* = \boldsymbol{\theta} + h\boldsymbol{M}^{-1}\boldsymbol{m}_* \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Now, let us check whether this condition can simultaneously hold for the forward and backward transitions in eq. (2). That is, is it possible that

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + h\boldsymbol{M}^{-1}\boldsymbol{m}_{n+1} \quad \text{and} \quad \boldsymbol{\theta}_n = \boldsymbol{\theta}_{n+1} + h\boldsymbol{M}^{-1}(-\boldsymbol{m}_n)\ ? \tag{8}$$

Solving the linear system we can see that, since $\boldsymbol{M}$ is positive definite and $h > 0$, eq. (8) is true if and only if $\boldsymbol{m}_n = \boldsymbol{m}_{n+1}$. Since the step size is $h > 0$, this only happens when the Gaussian random draw $\boldsymbol{\epsilon}$ in eq. (5) is equal to $\boldsymbol{v} = (h\boldsymbol{M}2\gamma T)^{-1/2}(h\gamma\boldsymbol{m}_n + h\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}_n))$. Because the covariance of $\boldsymbol{\epsilon}$ is the identity, this is a probability zero event.[2]

Thus, with probability 1, $\boldsymbol{\epsilon} \neq \boldsymbol{v}$, which implies $\boldsymbol{m}_n \neq \boldsymbol{m}_{n+1}$. We know the left-hand-side expression of eq. (8) is true, so that implies $\boldsymbol{\theta}_n \neq \boldsymbol{\theta}_{n+1} + h\boldsymbol{M}^{-1}(-\boldsymbol{m}_n)$, which implies by eq. (7) that $g(\boldsymbol{\theta}_n, -\boldsymbol{m}_n \,|\, \boldsymbol{\theta}_{n+1}, -\boldsymbol{m}_{n+1}) = 0$. ■

The easiest way to ensure $\boldsymbol{m}_n = \boldsymbol{m}_{n+1}$, such that the acceptance probability remains nonzero, is to set the time step $h = 0$. This, however, makes the sampler useless. If we take the limit $h \to 0$ instead, the Euler-Maruyama scheme gets arbitrarily close to the true SDE trajectory. Crucially however, per theorem 1, the acceptance probability remains 0 for any $h > 0$. Thus, it is impossible to use the acceptance probability to monitor the discretisation error. The Euler-Maruyama scheme *cannot* satisfy detailed balance. So far, we have not considered stochastic gradients, but this result includes them: it still holds if we substitute an arbitrary $\boldsymbol{g}_n$ for $\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}_n)$.

4. GGMC solves the problem by using a different integrator

To get a nonzero acceptance probability, we can just use a time-reversible integrator. The one we present here has been discovered several times in statistics (Horowitz, 1991; Neal, 1993) and molecular dynamics (Bussi and Parrinello, 2007; Leimkuhler and Matthews, 2015). We call it Gradient-guided Monte Carlo (GGMC), following Horowitz (1991), because it operates similarly to random walk Metropolis-Hastings, but with a gradient influencing the random walk's direction. Neal (1993, sec. 5.1) calls it the "stochastic dynamics" method, or "Langevin Monte Carlo (LMC) with partial momentum refreshment" (Neal, 2012). Leimkuhler and Matthews (2015) call this integrator "OBABO" because of the order of its steps.

The integrator splits Langevin dynamics (eqs. 4–3) into three parts, which can be solved exactly, and applies them in sequence (Leimkuhler and Matthews, 2015). Let $a = e^{-\gamma h}$, and

2. This is undesirable for another reason: we would like the momentum to change during the sampling process, so the state changes sometimes, i.e. $\boldsymbol{s}_{n+1} \neq \boldsymbol{s}_n$.

$\boldsymbol{\epsilon}, \boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ be Gaussian random variables with identity covariance. Then

$$\boldsymbol{m}_{n+1/4} = \sqrt{a}\boldsymbol{m}_n + \sqrt{(1-a)T}\boldsymbol{M}^{1/2}\boldsymbol{\epsilon}, \tag{O.1}$$

$$\boldsymbol{m}_{n+1/2} = \boldsymbol{m}_{n+1/4} - \frac{h}{2}\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}_n), \tag{B.1}$$

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + h\boldsymbol{M}^{-1}\boldsymbol{m}_{n+1/2}, \tag{A}$$

$$\boldsymbol{m}_{n+3/4} = \boldsymbol{m}_{n+1/2} - \frac{h}{2}\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}_{n+1}), \tag{B.2}$$

$$\boldsymbol{m}_{n+1} = \sqrt{a}\boldsymbol{m}_{n+3/4} + \sqrt{(1-a)T}\boldsymbol{M}^{1/2}\boldsymbol{\epsilon}'. \tag{O.2}$$

Steps O.1 and O.2 are partial momentum refreshment, and steps B.1, A and B.2 form one leapfrog step. Thus, the discretisation error is locally $O(h^3)$ (Neal, 2012). Establishing the trajectory discretisation error is complicated, because it requires solving a partial differential equation (PDE). However, all found PDE solutions indicate that the error is $O(h^2)$ (Leimkuhler and Matthews, 2015; Sivak et al., 2014), the same as HMC.

This scheme generalizes several popular samplers. To obtain HMC, we may repeat BAB many times, and use O as momentum refreshment. Alternatively, setting $\gamma = 0$ implies $a = 1$, so the O steps disappear, and the integrator becomes one leapfrog step. Adding a separate momentum refreshment then also yields HMC. To obtain MALA[3] or SGLD (Welling and Teh, 2011), we take $\gamma \to \infty$. This implies $a \to 0$, which means that step O.1 entirely ignores the incoming momentum $\boldsymbol{m}_n$ and just samples a fresh one. Because of this, we do not need to calculate the next momentum $\boldsymbol{m}_{n+1}$ at the end of a step, or represent the momentum at all. Thus, steps B.2 and O.2 fall out. We may also fold steps O.1 and B.1 into A to obtain the more common single-step expression for SGLD.

Note that there are other modern methods that simulate Langevin dynamics while preserving reversibility. For instance, Matthews and Weare (2018) propose a variant of ABOBA that corrects for the (known or estimated) covariance of the gradient noise, similar to SGHMC, to get the correct $O(h^2)$ statistics. Also, Zhang et al. (2020) develop an integrator that uses time-asymmetric gradients but, nevertheless, is realizable in the reverse direction and thus has a nonzero acceptance probability.

Acceptance probability. The key ingredient is the ratio of backward and forward transition probabilities, which we derive in appendix A. The resulting acceptance probability is

$$\log \mathrm{MH}\left(\boldsymbol{\theta}_{n+1}, \boldsymbol{m}_{n+1} \,|\, \boldsymbol{\theta}_n, \boldsymbol{m}_n\right) = -\frac{1}{T}\big(U(\boldsymbol{\theta}_{n+1}) - U(\boldsymbol{\theta}_n) + K(\boldsymbol{m}_{n+3/4}) - K(\boldsymbol{m}_{n+1/4})\big). \tag{9}$$

This is the same expression as in HMC or Langevin MC, were we to regard only the inner leapfrog step as the transition to accept or reject. Interestingly, the acceptance probability does not at all depend on the permanence a of the momentum.

5. Acceptance probabilities for multiple and stochastic steps

The inventors of HMC, Duane et al. (1987), already observed that the kinetic and potential energies used to specify the transition probability g need not be the same ones used to

3. Leimkuhler and Matthews (2015) recommend the BAOAB ordering for more accurate sampling in MALA.

calculate the acceptance probability. Only the latter will determine what distribution we actually sample from. That is, the sampler (be it HMC, or GGMC) may use stochastic or incorrect gradients: so long as the MH acceptance probability is calculated with respect to the target $\tilde{\pi}(\boldsymbol{\theta})^{1/T} e^{-K(\boldsymbol{m})/T}$, the resulting algorithm will estimate averages with respect to the target distribution.

This would seem to be irrelevant in practice because, naïvely, we need to calculate the exact acceptance probability after every (stochastic) GGMC step. This would bring the overall algorithm's per-step complexity to linear in the size of the data set, which is precisely what we want to avoid when using stochastic gradients.

Instead, perhaps we can calculate the MH correction only after many GGMC steps. Naïvely, this is impractical, since even the two-step transition probability is an intractable integral: $g_2(\boldsymbol{s}_{n+2} \mid \boldsymbol{s}_n) = \int g(\boldsymbol{s}_{n+2} \mid \boldsymbol{s}_{n+1}) g(\boldsymbol{s}_{n+1} \mid \boldsymbol{s}_n) \mathrm{d}\boldsymbol{s}_{n+1}$. However, it turns out we can just multiply the per-step probabilities. Starting at $\boldsymbol{s}_n$, after N steps of MCMC, we can accept the resulting sample $\boldsymbol{s}_{n+N}$ with probability $\min(1, \mathrm{MH}(\boldsymbol{s}_{n+N} \mid \boldsymbol{s}_n))$ (Zhang et al., 2020), where

$$\mathrm{MH}(\boldsymbol{s}_{n+N} \mid \boldsymbol{s}_n) = \prod_{i=1}^{N} \mathrm{MH}(\boldsymbol{s}_{n+i} \mid \boldsymbol{s}_{n+i-1}) = \frac{\bar{P}(\boldsymbol{s}_{n+N})}{\bar{P}(\boldsymbol{s}_n)} \prod_{i=1}^{N} \frac{g(\boldsymbol{s}_{n+i} \mid \boldsymbol{s}_{n+i-1})}{g(\boldsymbol{s}_{n+i-1} \mid \boldsymbol{s}_{n+i})}. \tag{10}$$

Due to telescopic cancellation, we need only evaluate the exact likelihood at the end of many steps. For example, we could follow 10 epochs of stochastic gradients, and then spend 1 epoch evaluating the likelihood exactly for the MH correction step. This can be combined with other algorithms (Neal, 2012; Sohl-Dickstein et al., 2014, sec. 5.4) to reduce the rejection probability.

Zhang et al. (2020, Lemma 2) propose this deferred MH scheme and show its correctness for any sampler with a homogeneous Markov chain. We can extend it to inhomogeneous Markov chains, such as the cosine schedule (Zhang et al., 2019) by considering the following scheme. We augment the state of the Markov chain with N random variables, $\{\zeta_i\}_{i=1}^{N}$, that will determine the randomness of the intermediate steps. Their target distributions have to be time-symmetric, i.e. $p_{\zeta_i}(\cdot) = p_{\zeta_{N-i+1}}(\cdot)$ for all i. Before an MCMC multi-step, we sample all the ζ_i from their target distribution. Then, we use them to provide all the randomness for the intermediate steps, so each step is just the result of a deterministic transformation:

$$\boldsymbol{s}_{n+i} = f_i(\boldsymbol{s}_{n+i-1}; \zeta_1, \ldots, \zeta_i), \text{ for all } i = 1, \ldots, N. \tag{11}$$

The functions f_i have to be time-symmetric as well ($f_i = f_{N-i+1}$). Otherwise, the backward transition is not realizable using the same ζ_i as the forward one, so the acceptance probability is zero. If they are time-symmetric, the resulting acceptance probability is eq. (10).

In this view, the cosine schedule (Zhang et al., 2019) has acceptance probability zero, even if a backward-realizable integrator is used. The cosine schedule is not a time-symmetric transformation: the schedule of learning rates is high at the beginning and low at the end of a cycle.

Wu et al. (2020) employ a similar construction to use MCMC samplers as normalizing flows. Neal (1996, Section 3.5.1) proposed the "partial gradient" construction: a variant of this for HMC and dataset minibatching in particular, motivated by the same operator-splitting view of integrators as the leapfrog.

6. Conclusion

We have shown that two popular SG-MCMC samplers cannot be corrected using rejection sampling, since their acceptance probabilities are zero. We have then revisited GGMC, which solves this problem and generalizes SGLD and HMC. Finally, we have also presented a way to compute the acceptance probabilities of GGMC across several steps, leading to a practical method for drawing exact posterior samples using only stochastic gradients.

It is time for the Bayesian deep learning community to let go of the Euler-Maruyama scheme. Instead, we should explore the breadth of integrators for Langevin dynamics which we have inherited from molecular dynamics and statistics. We have shown that correctness is achievable. Thus, we should demand correctness from our samplers.

References

Giovanni Bussi and Michele Parrinello. Accurate sampling using Langevin dynamics. *Physical Review E*, 75, 2007. doi: 10.1103/PhysRevE.75.056707.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, volume 32 of *Proceedings of Machine Learning Research*, pages 1683–1691, 2014. URL http://proceedings.mlr.press/v32/cheni14.html.

Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems*, volume 27, pages 3203–3211, 2014. URL https://proceedings.neurips.cc/paper/2014/file/21fe5b8ba755eeaece7a450849876228-Paper.pdf.

Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987. doi: 10.1016/0370-2693(87)91197-X.

Charles J Geyer. Practical Markov chain Monte Carlo. *Statistical science*, pages 473–483, 1992.

W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Alan M. Horowitz. A generalized guided Monte Carlo algorithm. *Physics Letters B*, 268(2): 247 – 252, 1991. ISSN 0370-2693. doi: https://doi.org/10.1016/0370-2693(91)90812-5. URL http://www.sciencedirect.com/science/article/pii/0370269391908125.

Ben Leimkuhler and Charles Matthews. *Numerical Methods for Stochastic Molecular Dynamics*, chapter 7, pages 261–328. Springer International Publishing, Cham, 2015. ISBN 978-3-319-16375-8. doi: 10.1007/978-3-319-16375-8_7. URL https://doi.org/10.1007/978-3-319-16375-8_7.

Benedict Leimkuhler, Charles Matthews, and Tiffany Vlaar. Partitioned integrators for thermodynamic parameterization of neural networks. 2019. URL http://arxiv.org/abs/1908.11843v2.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems*, volume 28, pages 2917–2925, 2015. URL https://proceedings.neurips.cc/paper/2015/file/9a4400501febb2a95e79248486a5f6d3-Paper.pdf.

Charles Matthews and Jonathan Weare. Langevin Markov chain Monte Carlo with stochastic gradients. 2018. URL http://arxiv.org/abs/1805.08863v2.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Radford M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, 1993. URL https://www.cs.toronto.edu/~radford/review.abstract.html.

Radford M. Neal. *Bayesian learning for neural networks*, volume 118. Springer, 1996.

Radford M. Neal. MCMC using Hamiltonian dynamics. 2012. URL http://arxiv.org/abs/1206.1901v1.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998. doi: https://doi.org/10.1111/1467-9868.00123. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00123.

David A. Sivak, John D. Chodera, and Gavin E. Crooks. Time step rescaling recovers continuous-time dynamical properties for discrete-time Langevin integration of nonequilibrium systems. *The Journal of Physical Chemistry B*, 118, 2014. doi: 10.1021/jp411770f.

Jascha Sohl-Dickstein, Mayur Mudigonda, and Michael DeWeese. Hamiltonian monte carlo without detailed balance. volume 32 of *Proceedings of Machine Learning Research*, pages 719–726, 2014. URL http://proceedings.mlr.press/v32/sohl-dickstein14.html.

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

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the Bayes posterior in deep neural networks really?, 2020. URL http://arxiv.org/abs/2002.02405v1.

Hao Wu, Jonas Köhler, and Frank Noé. Stochastic normalizing flows. 2020. URL http://arxiv.org/abs/2002.06707v3.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. 2019. URL http://arxiv.org/abs/1902.03932v2.

Ruqi Zhang, A. Feder Cooper, and Christopher De Sa. AMAGOLD: Amortized Metropolis adjustment for efficient stochastic gradient MCMC. 2020. URL http://arxiv.org/abs/2003.00193v1.

Appendix A. Acceptance probability calculation for GGMC

The procedure used by Bussi and Parrinello (2007) is to start with the joint distribution of the noises $\boldsymbol{\epsilon}, \boldsymbol{\epsilon}'$ and transform it to the distribution over $\boldsymbol{\theta}_{n+1}, \boldsymbol{m}_{n+1} \,|\, \boldsymbol{\theta}_n, \boldsymbol{m}_n$ using the density transformation formula with the determinant of the Jacobian.

However, in this case, there is somewhat more illustrative way. Consider

$$\boldsymbol{r} = \sqrt{(1-a)T}\boldsymbol{M}^{-1/2}\boldsymbol{\epsilon}, \quad \text{and thus} \quad \boldsymbol{r} \sim \mathcal{N}(\boldsymbol{0}, (1-a)T\boldsymbol{M}), \tag{12}$$

and similarly for $\boldsymbol{r}'$ and $\boldsymbol{\epsilon}'$. Note that $\boldsymbol{r} = \boldsymbol{m}_{n+1/4} - \sqrt{a}\boldsymbol{m}_n$, and $\boldsymbol{r}' = \boldsymbol{m}_{n+1} - \sqrt{a}\boldsymbol{m}_{n+3/4}$. By substituting $\boldsymbol{r}, \boldsymbol{r}'$ into eqs. (B.1) and (B.2), we can write $\boldsymbol{\theta}_{n+1}, \boldsymbol{m}_{n+1}$ as

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + h\boldsymbol{M}^{-1}\left(\sqrt{a}\boldsymbol{m}_n - \frac{h}{2}\nabla_{\boldsymbol{\theta}}U(\boldsymbol{\theta}_n) + \boldsymbol{r}\right), \tag{13}$$

$$\boldsymbol{m}_{n+1} = a\boldsymbol{m}_n - \frac{\sqrt{a}h}{2}(\nabla_{\boldsymbol{\theta}}U(\boldsymbol{\theta}_n) + \nabla_{\boldsymbol{\theta}}U(\boldsymbol{\theta}_{n+1})) + \sqrt{a}\boldsymbol{r} + \boldsymbol{r}'. \tag{14}$$

The reciprocal of the determinant of the Jacobian is thus

$$\begin{vmatrix} \mathrm{d}\boldsymbol{\theta}_{n+1}/\mathrm{d}\boldsymbol{r} & \mathrm{d}\boldsymbol{\theta}_{n+1}/\mathrm{d}\boldsymbol{r}' \\ \mathrm{d}\boldsymbol{m}_{n+1}/\mathrm{d}\boldsymbol{r} & \mathrm{d}\boldsymbol{m}_{n+1}/\mathrm{d}\boldsymbol{r}' \end{vmatrix}^{-1} = \begin{vmatrix} h\boldsymbol{M}^{-1} & 0 \\ \sqrt{a} & \boldsymbol{I} \end{vmatrix}^{-1} = h^{-1}|\boldsymbol{M}|, \tag{15}$$

which is a constant, so it will be eliminated when dividing by the backward transition density.

Putting this all together, the forward transition density is

$$\begin{aligned} g(\boldsymbol{\theta}_{n+1}, \boldsymbol{m}_{n+1} \,|\, \boldsymbol{\theta}_n, \boldsymbol{m}_n) = h^{-1}|\boldsymbol{M}|\mathcal{N}\big(\boldsymbol{m}_{n+1/4} - \sqrt{a}\boldsymbol{m}_n \,\big|\, \boldsymbol{0}, (1-a)T\boldsymbol{M}\big) \\ \mathcal{N}\big(\boldsymbol{m}_{n+1} - \sqrt{a}\boldsymbol{m}_{n+3/4} \,\big|\, \boldsymbol{0}, (1-a)T\boldsymbol{M}\big). \end{aligned} \tag{16}$$

By substituting $\boldsymbol{\theta}_{n+1} \leftrightarrow \boldsymbol{\theta}_n$, $\boldsymbol{m}_n \to -\boldsymbol{m}_{n+1}$, $\boldsymbol{m}_{n+1/4} \to -\boldsymbol{m}_{n+3/4}$, $\boldsymbol{m}_{n+3/4} \to -\boldsymbol{m}_{n+1/4}$, $\boldsymbol{m}_{n+1} \to -\boldsymbol{m}_n$, we can obtain the backward transition probability

$$\begin{aligned} g(\boldsymbol{\theta}_n, -\boldsymbol{m}_n \,|\, \boldsymbol{\theta}_{n+1}, -\boldsymbol{m}_{n+1}) = h^{-1}|\boldsymbol{M}|\mathcal{N}\big(-\boldsymbol{m}_{n+3/4} + \sqrt{a}\boldsymbol{m}_{n+1} \,\big|\, \boldsymbol{0}, (1-a)T\boldsymbol{M}\big) \\ \mathcal{N}\big(-\boldsymbol{m}_n + \sqrt{a}\boldsymbol{m}_{n+1/4} \,\big|\, \boldsymbol{0}, (1-a)T\boldsymbol{M}\big). \end{aligned} \tag{17}$$

Dividing them, the constant terms from the Gaussians and $h^{-1}|\boldsymbol{M}|$ disappear, and we obtain

$$\begin{aligned} &\log\frac{g(\boldsymbol{\theta}_n, -\boldsymbol{m}_n \,|\, \boldsymbol{\theta}_{n+1}, -\boldsymbol{m}_{n+1})}{g(\boldsymbol{\theta}_{n+1}, \boldsymbol{m}_{n+1} \,|\, \boldsymbol{\theta}_n, \boldsymbol{m}_n)} = \\ &\quad -\frac{1}{(1-a)T}\Big(K\big(-\boldsymbol{m}_{n+3/4} + \sqrt{a}\boldsymbol{m}_{n+1}\big) + K\big(-\boldsymbol{m}_n + \sqrt{a}\boldsymbol{m}_{n+1/4}\big) \\ &\qquad\qquad - K\big(\boldsymbol{m}_{n+1/4} - \sqrt{a}\boldsymbol{m}_n\big) - K\big(\boldsymbol{m}_{n+1} - \sqrt{a}\boldsymbol{m}_{n+3/4}\big)\Big) \end{aligned} \tag{18}$$

where $K(\boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^\mathsf{T}\boldsymbol{M}^{-1}\boldsymbol{v}$ is the kinetic energy defined in section 3. This simplifies greatly, to

$$\log \frac{g(\boldsymbol{\theta}_n, -\boldsymbol{m}_n \,|\, \boldsymbol{\theta}_{n+1}, -\boldsymbol{m}_{n+1})}{g(\boldsymbol{\theta}_{n+1}, \boldsymbol{m}_{n+1} \,|\, \boldsymbol{\theta}_n, \boldsymbol{m}_n)} = -\frac{1}{T}\big(K(\boldsymbol{m}_n) - K(\boldsymbol{m}_{n+1}) + K\big(\boldsymbol{m}_{n+3/4}\big) - K\big(\boldsymbol{m}_{n+1/4}\big)\big). \tag{19}$$

The log ratio of the Boltzmann densities (eq. (1)) is

$$\log \frac{\bar{P}(\boldsymbol{\theta}_{n+1}, \boldsymbol{m}_{n+1})}{\bar{P}(\boldsymbol{\theta}_n, \boldsymbol{m}_n)} = -\frac{1}{T}(U(\boldsymbol{\theta}_{n+1}) - U(\boldsymbol{\theta}_n) + K(\boldsymbol{m}_{n+1}) - K(\boldsymbol{m}_n)). \tag{20}$$

Now, to obtain the acceptance probability in eq. (2) we add eqs. (19) and (20), obtaining

$$\log \mathrm{MH}\,(\boldsymbol{\theta}_{n+1}, \boldsymbol{m}_{n+1} \,|\, \boldsymbol{\theta}_n, \boldsymbol{m}_n) = -\frac{1}{T}\big(U(\boldsymbol{\theta}_{n+1}) - U(\boldsymbol{\theta}_n) + K(\boldsymbol{m}_{n+3/4}) - K(\boldsymbol{m}_{n+1/4})\big) \tag{9}$$

which is exactly the same as eq. (9).