

RESEARCH ARTICLE

Quantum-Inspired Magnetic Hamiltonian Monte Carlo

Wilson Tsakane Mongwe^{1*}, Rendani Mbuyha², Tshilidzi Marwala¹

1 School of Electrical Engineering, University of Johannesburg, Johannesburg, South Africa, **2** School of Statistics and Actuarial Science, University of Witwatersrand, Johannesburg, South Africa

* wilsonmongwe@gmail.com

Abstract

Hamiltonian Monte Carlo (HMC) is a Markov Chain Monte Carlo algorithm that is able to generate distant proposals via the use of Hamiltonian dynamics, which are able to incorporate first-order gradient information about the target posterior. This has driven its rise in popularity in the machine learning community in recent times. It has been shown that making use of the energy-time uncertainty relation from quantum mechanics, one can devise an extension to HMC by allowing the mass matrix to be random with a probability distribution instead of a fixed mass. Furthermore, Magnetic Hamiltonian Monte Carlo (MHMC) has been recently proposed as an extension to HMC and adds a magnetic field to HMC which results in non-canonical dynamics associated with the movement of a particle under a magnetic field. In this work, we utilise the non-canonical dynamics of MHMC while allowing the mass matrix to be random to create the Quantum-Inspired Magnetic Hamiltonian Monte Carlo (QIMHMC) algorithm, which is shown to converge to the correct steady state distribution. Empirical results on a broad class of target posterior distributions show that the proposed method produces better sampling performance than HMC, MHMC and HMC with a random mass matrix.

OPEN ACCESS

Citation: Mongwe WT, Mbuyha R, Marwala T (2021) Quantum-Inspired Magnetic Hamiltonian Monte Carlo. PLoS ONE 16(10): e0258277. <https://doi.org/10.1371/journal.pone.0258277>

Editor: Mohammadreza Hadizadeh, Central State University & Ohio University, UNITED STATES

Received: July 20, 2021

Accepted: September 22, 2021

Published: October 5, 2021

Copyright: © 2021 Mongwe et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: UCI Machine Learning Repository, URL: <https://archive.ics.uci.edu/ml/index.php>.

Funding: Funding for this research was provided by Google PhD Fellowships and the National Research Fund of South Africa. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: We declare that this manuscript is original, has not been published before and is not currently considered for

1 Introduction

The inference of complex probabilistic models using Markov Chain Monte Carlo (MCMC) algorithms has become very common [1–7]. MCMC methods have been successfully applied in a variety of fields including health, finance and cosmology [1, 4, 5, 8–15]. The goal of MCMC methods is to construct a Markov chain that leaves the target posterior distribution invariant [3, 16]. This allows the generated samples from the chains to be used to estimate expectations with respect to the target posterior distribution. Examples of such constructions include the Metropolis-Hasting [17] method, slice sampling [18] and Gibbs sampling [19]. While these algorithms are able to construct Markov chains that asymptotically converge to the correct distribution, they still suffer from random-walk behaviour [3]. This results in high correlations between the generated samples and consequently low effective sample sizes.

For instances where the target posterior is differentiable on the Euclidean manifold, Hamiltonian Monte Carlo (HMC) provides a powerful mechanism to sample from differentiable

publication elsewhere. The authors also declare that there are no conflicts of interest.

target posterior distributions [20, 21]. HMC extends the parameter space into the phase space via the introduction of an auxiliary momentum variable which ensures that different energy levels are explored. HMC exploits the first-order gradient information of the target density to guide its exploration of the phase space. The use of gradient information reduces the random-walk behaviour typically associated with the Metropolis-Hastings algorithm [3, 10, 21]. HMC introduces the mass matrix for the momentum variable, trajectory length and step size parameters that need to be tuned for optimal results. Extensions of HMC include Riemannian Manifold Hamiltonian Monte Carlo (RMHMC) [1], the No-U-Turn Sampler (NUTS) [22] and Magnetic Hamiltonian Monte Carlo (MHMC) [6]. Furthermore, machine learning techniques have also been utilised to enhance the efficiency of Monte Carlo sampling algorithms. In Mcnaughton *et al.* [23], autoregressive neural networks are employed to boost the efficiency of Monte Carlo methods, while Levy *et al.* [24] generalises Hamiltonian Monte Carlo using neural networks.

RMHMC extends HMC by incorporating second-order gradient information of the target posterior. This allows RMHMC to take into account the local geometry of the target as it explores the phase space, which is important for ill-conditioned distributions [1, 5, 7, 25]. This results in the Hamiltonian being non-separable, which necessitates the use of implicit numerical integration schemes which are computationally expensive. On the other-hand, NUTS extends HMC by automatically tuning the trajectory length and step size parameters of HMC, which would need to be specified by the user [22, 25]. The trajectory length is tuned by iteratively doubling the trajectory length until either the parameters trace back, or the Hamiltonian becomes infinite [22]. The step size is tuned by employing the dual averaging methodology during the burn-in phase [22]. This removes the need for manual tuning of these parameters and the associated costly pilot runs, making HMC more accessible to non-experts. A recent extension to HMC is MHMC which adds a magnetic field to HMC [6]. This results in faster convergence to the correct target density, and much lower auto-correlations between the generated samples when compared to HMC [4, 6, 26]. When the magnetic component of MHMC is absent, MHMC and HMC have the same dynamics [4, 6, 26]. MHMC has similar execution times to HMC, which illustrates the close relationship between HMC and MHMC.

One of the parameters that needs to be set in HMC is the mass matrix of the auxiliary momentum variable. This mass matrix is typically set to equal the identity matrix [1, 6, 9, 21]. Although this produces good results, it is not necessarily the optimal choice across all target distributions. In RMHMC, the mass matrix is set to be the Hessian of the negative log-density—which is the fisher information metric [1, 25]. In Quantum-Inspired Hamiltonian Monte Carlo algorithm (QIHMC) [7], the mass matrix is set to be a stochastic process. This is motivated by the energy-time uncertainty relation from quantum mechanics, which allows a particle's mass to be stochastic to rather fixed [7].

QIHMC has been shown to improve performance when sampling from a broad class of distributions which occur in sparse modeling via bridge regression, image denoising and Bayesian neural network pruning [5, 7]. This is particularly important for spiky and multi-modal distributions, where HMC is inefficient [5, 7]. The use of a random mass matrix is yet to be considered in the context of MHMC in the literature. Given that MHMC is closely related to HMC, one would expect that making the mass matrix random would result in improved sampling performance when compared to MHMC with fixed mass.

In this work, we present the Quantum-Inspired Magnetic Hamiltonian Monte Carlo (QIMHMC) algorithm which uses a random mass matrix in the non-canonical Hamiltonian dynamics offered by MHMC. A proof that the proposed algorithm converges to the correct distribution is presented. Numerical experiments across various benchmarks including the Banana shaped distribution, multivariate Gaussian distributions and Bayesian logistic

regression show that the proposed algorithm outperforms HMC, MHMC and QIHMC. The main contributions of this work can be summarised as:

- We present the Quantum-Inspired Magnetic Hamiltonian Monte Carlo which employs a stochastic mass matrix in Magnetic Hamiltonian Monte Carlo.
- We prove that the proposed algorithm converges to the correct target density.
- Numerical experiments on various target posteriors show that the proposed method outperforms Hamiltonian Monte Carlo, Magnetic Hamiltonian Monte Carlo and Quantum-Inspired Hamiltonian Monte Carlo.

The remainder of this paper is structured as follows: Section 2 discusses the Markov Chain Monte Carlo methods that form the basis of the new method, Section 3 presents the proposed method, Section 4 outlines the target distributions considered, Section 5 outlines the experiments conducted, Section 6 presents and discusses the results of the experiments and we provide a conclusion in Section 7.

2 Hamiltonian Monte Carlo methods

Instead of directly sampling the parameters, Hamiltonian Monte Carlo (HMC) introduces an auxiliary momentum variable and then samples the parameters and momenta jointly. HMC improves upon the Metropolis-Hastings [17] algorithm by using first-order gradient information of the target to guide its exploration [9, 20], which leads to low auto-correlations between the generated samples. The parameter vector \mathbf{w} and momentum variable \mathbf{p} , which is typically chosen to be independent of \mathbf{w} , form a dynamic system whose Hamiltonian is written as:

$$H(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + K(\mathbf{p}) \tag{1}$$

where $U(\mathbf{w})$ is the negative log-likelihood of the target posterior distribution and $K(\mathbf{p})$ is the kinetic energy defined by the kernel of a Gaussian with mass matrix \mathbf{M} [10]:

$$K(\mathbf{p}) = \frac{1}{2} \log((2\pi)^D |\mathbf{M}|) + \frac{\mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}}{2}. \tag{2}$$

The equations governing the Hamiltonian dynamics step are defined by Hamilton’s equations at a fictitious time t as follows [8, 20]:

$$\frac{d\mathbf{w}}{dt} = \frac{\partial H(\mathbf{w}, \mathbf{p})}{\partial \mathbf{p}}; \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial H(\mathbf{w}, \mathbf{p})}{\partial \mathbf{w}}. \tag{3}$$

The evolution of this Hamiltonian system must preserve both volume and total energy [3, 21]. As the Hamiltonian in Eq (1) is separable, to traverse the space we can employ the leapfrog integrator [8, 20]. The update equations for the leapfrog integration scheme are:

$$\begin{aligned} \mathbf{p}_{t+\frac{\epsilon}{2}} &= \mathbf{p}_t + \frac{\epsilon}{2} \frac{\partial H(\mathbf{w}_t, \mathbf{p}_t)}{\partial \mathbf{w}} \\ \mathbf{w}_{t+\epsilon} &= \mathbf{w}_t + \epsilon \mathbf{M}^{-1} \mathbf{p}_{t+\frac{\epsilon}{2}} \\ \mathbf{p}_{t+\epsilon} &= \mathbf{p}_{t+\frac{\epsilon}{2}} + \frac{\epsilon}{2} \frac{\partial H(\mathbf{w}_{t+\epsilon}, \mathbf{p}_{t+\frac{\epsilon}{2}})}{\partial \mathbf{w}}. \end{aligned} \tag{4}$$

Due to the discretisation errors arising from the leapfrog integration, HMC utilises the

Metropolis-Hastings algorithm in which the parameters \mathbf{w}^* are accepted with probability:

$$P(\text{accept } \mathbf{w}^*) = \min \left(1, \frac{\exp(-H(\mathbf{w}^*, \mathbf{p}^*))}{\exp(-H(\mathbf{w}, \mathbf{p}))} \right). \tag{5}$$

The overall HMC sampling process follows a Gibbs sampling scheme, where we sample the momentum and then sample a new set of parameters given the drawn momentum. Algorithm 1 shows the pseudo-code for the HMC where ϵ is the discretisation step size and L is the trajectory length.

Algorithm 1: Hamiltonian Monte Carlo

Input: $N, \epsilon, L, w_{\text{init}}, H(w, p)$

Output: $(w)_{m=0}^N$

```

1:  $w_0 \leftarrow w_{\text{init}}$ 
2: for  $m \rightarrow 1$  to  $N$  do
3:    $p_{m-1} \sim \mathcal{N}(0, \mathbf{M})$ 
4:    $p_m, w_m = \text{Leapfrog}(p_{m-1}, w_{m-1}, \epsilon, L, H)$  in Eq (4)
5:    $\delta H = H(w_{m-1}, p_{m-1}) - H(w_m, p_m)$ 
6:    $\alpha_m = \min(1, \exp(\delta H))$ 
7:    $u_m \sim \text{Unif}(0, 1)$ 
8:    $w_m = \text{Metropolis}(\alpha_m, u_m, w_m, w_{m-1})$  in Eq (5)
9: end for

```

Although HMC improves on the Metropolis-Hastings algorithm and reduces random-walk behaviour, it still produces relatively high auto-correlations in the samples. Magnetic Hamiltonian Monte Carlo (MHMC) is a special case of non-canonical HMC corresponding to motion of a particle in a magnetic field [6, 27], which aims to improve the exploration of the target distribution [6]. This magnetic field adds an extra degree of freedom and can be tuned to attain significantly better sampling performance compared to HMC with very little extra computational cost [4, 6, 26].

MHMC utilises the same Hamiltonian as in HMC, but exploits non-canonical Hamiltonian dynamics where the canonical matrix now has a non-zero element on the diagonal [4, 6]. The MHMC dynamics are:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{w} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{w}} H(\mathbf{w}, \mathbf{p}) \\ \nabla_{\mathbf{p}} H(\mathbf{w}, \mathbf{p}) \end{bmatrix} \tag{6}$$

where \mathbf{G} is the magnetic field. The update equations for the leapfrog-like integration scheme in MHMC are given as [6]:

$$\begin{aligned}
 \mathbf{p}_{t+\frac{\epsilon}{2}} &= \mathbf{p}_t + \frac{\epsilon}{2} \frac{\partial H(\mathbf{w}_t, \mathbf{p}_t)}{\partial \mathbf{w}} \\
 \mathbf{w}_{t+\epsilon} &= \mathbf{w}_t + \mathbf{G}^{-1}(\exp(\mathbf{G}\epsilon) - \mathbf{I})\mathbf{M}^{-1}\mathbf{p}_{t+\frac{\epsilon}{2}} \\
 \mathbf{p}_{t+\frac{\epsilon}{2}} &= \exp(\mathbf{G}\epsilon)\mathbf{p}_{t+\frac{\epsilon}{2}} \\
 \mathbf{p}_{t+\epsilon} &= \mathbf{p}_{t+\frac{\epsilon}{2}} + \frac{\epsilon}{2} \frac{\partial H(\mathbf{w}_{t+\epsilon}, \mathbf{p}_{t+\frac{\epsilon}{2}})}{\partial \mathbf{w}}.
 \end{aligned} \tag{7}$$

Eq (7) shows that MHMC only differs from HMC dynamics through the presence of a non-zero \mathbf{G} [4, 6]. MHMC and HMC have exactly the same dynamics when the magnetic component is absent. This can be seen by performing a matrix Taylor series expansion of $\exp(\mathbf{G}\epsilon)$ in the $\mathbf{w}_{t+\epsilon}$ update in Eq (7) and then setting $\mathbf{G} = 0$.

As with HMC, these non-canonical dynamics in MHMC cannot be integrated exactly, and a numerical integration scheme with a Metropolis-Hastings acceptance step must be utilised to ensure detailed balance. The algorithm for MHMC is the same as the HMC algorithm in Algorithm 1, except that the leapfrog integration scheme is replaced by the leapfrog-like integration scheme corresponding to MHMC in Eq (7).

An issue that we are yet to address is the selection of the mass matrix \mathbf{M} in both HMC and MHMC. In practice, \mathbf{M} is commonly set to equal the identity matrix \mathbf{I} [4, 6, 10]. Although this produces good results, this approach is not always the optimal approach to use [7]. In this work, we set \mathbf{M} to be a stochastic process. When applied to HMC, this approach gives result to the Quantum-Inspired Hamiltonian Monte Carlo of Liu and Zhang [7]. A contribution of this work is that we utilise a random mass matrix in MHMC to create the Quantum-Inspired Hamiltonian Monte Carlo (QIMHMC) algorithm which we describe in the following section.

3 Quantum-Inspired Magnetic Hamiltonian Monte Carlo

We now present the Quantum-Inspired Magnetic Hamiltonian Monte Carlo (QIMHMC) method which employs a random mass matrix in MHMC. This proposed algorithm has the same dynamics as MHMC in Eqs (6) and (7), with the exception that the mass matrix is random and is re-sampled before generating the auxiliary momentum variable. The algorithmic description of QIMHMC is presented in Algorithm 2.

The proposed algorithm has been named Quantum-Inspired Magnetic Hamiltonian Monte Carlo as it utilises a random mass matrix, which is consistent with the behaviour of quantum particles. In quantum mechanics, a particle can have a mass which is random and has its own distribution, while in classical mechanics a particle has a fixed mass. The classical version of the proposed algorithm is the Magnetic Hamiltonian Monte Carlo method. When a random mass is utilised as inspired by quantum particles, the result is Quantum-Inspired Magnetic Hamiltonian Monte Carlo. Furthermore, this naming convention is consistent with that used by Liu and Zhang [7] for Hamiltonian Monte Carlo. Their work differs with ours in that now the quantum particle is subjected to a magnetic field as well.

As we have introduced another source of randomness in the form of a random mass matrix into MHMC, we need to show that the proposed algorithm converges to the correct steady state distribution. This is provided in Theorem 3.1, which guarantees that the proposed algorithm produces the correct steady state distribution [7].

Theorem 3.1. *Consider continuous-time Hamiltonian dynamics with a deterministic time-varying positive-definite mass matrix $\mathbf{M}(t)$ in Eq (7). The marginal density $\pi_{\mathbf{w}}(\mathbf{w}) \propto \exp(-U(\mathbf{w}))$ is a unique steady state distribution in the \mathbf{w} space if momentum re-sampling steps $p_{\mathbf{p}}(\mathbf{p}) \propto \exp\left(\frac{\mathbf{p}^T \mathbf{M}(t)^{-1} \mathbf{p}}{2}\right)$ are included.*

Proof. Consider the joint distribution of $(\mathbf{w}, \mathbf{p}, \mathbf{M})$ given by $\pi(\mathbf{w}, \mathbf{p}, \mathbf{M})$. Here we have dropped the explicit dependence of \mathbf{M} on t because $\mathbf{M}(t)$ obeys the mass distribution $\mathbf{P}_{\mathbf{M}}(\mathbf{M})$ for all t . Employing Bayes theorem we have that:

$$\begin{aligned} \pi(\mathbf{w}, \mathbf{p}, \mathbf{M}) &= p(\mathbf{w}, \mathbf{p}|\mathbf{M})\mathbf{P}_{\mathbf{M}}(\mathbf{M}) \\ \pi(\mathbf{w}, \mathbf{p}, \mathbf{M}) &\propto \exp(-U(\mathbf{w})) \exp\left(\frac{\mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}}{2}\right) \\ \Rightarrow \pi(\mathbf{w}) &= \int_{\mathbf{p}} \int_{\mathbf{M}} \pi(\mathbf{w}, \mathbf{p}, \mathbf{M}) d\mathbf{w} d\mathbf{M} \\ \pi(\mathbf{w}) &\propto \exp(-U(\mathbf{w})) \end{aligned} \tag{8}$$

which means that the marginal steady state distribution $\pi(\mathbf{w})$ is the correct posterior distribution.

An aspect that we are yet to address is which distribution $\mathbf{P}_M(\mathbf{M})$ should be used for the mass matrix. This is still an open area of research [5, 7]. In this paper, we consider the simple case where \mathbf{M} is a diagonal matrix with the entries being sampled from a log-normal distribution where the mean is zero and the standard deviation, which we refer to as the vol-of-vol, is equal to a tunable parameter α . We present the sensitivity analysis to the chosen value of α in Section 5.3.

Note that the algorithm for QIHMC used in this paper is the same as that outlined in Algorithm 2, except that in step 5 we use the leapfrog integration scheme in Eq (4).

Algorithm 2: Quantum-Inspired Magnetic Hamiltonian Monte Carlo

```

Input:  $N, \varepsilon, L, w_{init}, H(w, p), \mathbf{G}$ 
Output:  $(w)_{m=0}^N$ 
1:  $w_0 \leftarrow w_{init}$ 
2: for  $m \rightarrow 1$  to  $N$  do
3:    $\mathbf{M} \sim \mathbf{P}_M(\mathbf{M}) \leftarrow$  re-sample mass matrix
4:    $p_{m-1} \sim \mathcal{N}(0, \mathbf{M})$ 
5:    $p_m, w_m =$  Integrator ( $p_{m-1}, w_{m-1}, \varepsilon, L, H, \mathbf{G}$ ) in Eq (7)
6:    $\delta H = H(w_{m-1}, p_{m-1}) - H(w_m, p_m)$ 
7:    $\alpha_m = \min(1, \exp(\delta H))$ 
8:    $u_m \sim \text{Unif}(0, 1)$ 
9:    $w_m =$  Metropolis ( $\alpha_m, u_m, w_m, w_{m-1}$ ) in Eq (5)
10: end for

```

4 The target posteriors

In this section, we outline the target distributions that we considered in this work. These target posterior densities have been extensively used in the literature [1, 4–6, 12, 28, 29], including the seminal work of Girolami and Caldehead [1].

4.1 Banana shaped distribution

The Banana shaped density is a 2-dimensional non-linear target which was first presented in Haario *et al.* [30]. The likelihood and prior distributions are given as:

$$y|\mathbf{w} \sim \mathcal{N}(w_1 + w_2^2 = 1, \sigma_y^2), \quad w_1, w_2 \sim \mathcal{N}(0, \sigma_w^2) \tag{9}$$

We generated one hundred data points for y with $\sigma_y^2 = 4$ and $\sigma_w^2 = 1$. Due to independence of the data and parameters, the posterior distribution is proportional to:

$$\prod_{i=1}^{i=N} p(y_i|\mathbf{w})p(w_1)p(w_2). \tag{10}$$

where $N = 100$ is the number of observations.

4.2 Multivariate Gaussian distributions

The objective is to sample from D -dimensional Gaussian distributions $\mathcal{N}(0, \Sigma)$. The covariance matrix Σ is set to be diagonal, with the standard deviations simulated from a log-normal distribution with mean zero and unit standard deviation. In this work, we consider the number of dimensions D to be in the set {50, 100}.

4.3 Bayesian logistic regression

We model the real world binary classification datasets shown in Table 1 using Bayesian logistic regression. The negative log-likelihood $l(D|w)$ function associated with logistic regression is given by:

$$l(D|w) = \sum_i^N y_i \log(w^T x_i) + (1 - y_i) \log(1 - w^T x_i) \tag{11}$$

where D is the data and N is the number of observations. Thus, the target unnormalised posterior log distribution is given as:

$$\ln p(w|D) = l(D|w) + \ln p(w|\beta) \tag{12}$$

where $\ln p(w|\beta)$ is the log of the prior distribution on the parameters given the hyperparameters β . We model the parameters w as having a Gaussian prior with each parameter having zero mean and standard deviation $\beta = 10$. These settings are the same as those used in Girolami and Caldehead [1].

5 Experimental setup

We now outline the settings used for the experiments, the performance metrics used and we perform a sensitivity analysis to the vol-of-vol parameter α .

5.1 Settings

In all our experiments, we compare the performance of QIMHMC to HMC, MHMC and QIHMC using the following metrics: acceptance rate of the generated samples, the effective sample size and the effective sample size normalised by execution time.

The acceptance rate metric measures the number of generated samples that are accepted in the Metropolis-Hastings acceptance step of the algorithm. The higher the number of accepted samples for the same step size, the more preferable the method. The effective sample size metric is a commonly used metric for assessing the sampling efficiency of an MCMC algorithm. It indicates the number of effectively uncorrelated samples out of the total number of generated samples. The larger the effective sample size, the better the performance of the MCMC method. The effective sample size normalised by execution time metric takes into account the computational resources required to generate the samples and penalises MCMC methods that require more computational resources to generate the same number of uncorrelated samples. The larger this metric, the better the efficiency of algorithm.

The vol-of-vol parameter α is set depending on the particular target density. The sensitivity analysis for α is presented in Section 5.3. We set α to 0.1 for the Banana shaped distribution and α was set to 0.3 for the other targets.

The trajectory length for all the MCMC methods considered in this work was set to 100 across all the targets. For the Banana shaped distribution, we used a step size of 0.1 for all the

Table 1. Real world benchmark datasets. N represents the number of observations. D represents the number of model parameters.

Dataset	Features	N	D
Heart	13	270	14
Australian credit	14	690	15
German credit	24	1 000	25

<https://doi.org/10.1371/journal.pone.0258277.t001>

MCMC methods. Step sizes of 0.07 and 0.05 were used for each value of D , in that order, for the multivariate Gaussian distributions, with the Bayesian logistic regression datasets all using a step size of 0.02. Ten independent chains were run for each method on each target distribution. A total of 3 000 samples were generated for each target, with the first 1 000 samples discarded as burn-in. This was sufficient for all the algorithms to converge on all the targets. All experiments were conducted on a 64bit CPU using PyTorch.

As mentioned previously, the matrix \mathbf{G} in the MHMC and QIMHMC methods provides an extra degree of freedom which results in better sampling behavior than HMC [4, 6, 31]. It is not immediately clear how this matrix should be set—this is still an open area of research [4, 6, 27]. In this work, we select only a few dimensions to be influenced by the magnetic field. In particular, \mathbf{G} was set such that $\mathbf{G}_{ii} = g$, $\mathbf{G}_{i1} = -g$ and zero elsewhere for $g = 0.2$. These settings are used across all the targets.

It is worth highlighting that, in this work, the mass matrix \mathbf{M} is set to the identity matrix for the HMC and MHMC methods. This is what is commonly used in practice [3, 4, 6, 26].

5.2 Multivariate effective sample size

This work employs the multivariate effective sample size metric developed by Vats *et al.* [32] instead of the minimum univariate ESS metric typically used in analysing MCMC results. The minimum univariate ESS measure is not able to capture the correlations between the different parameter dimensions, while the multivariate ESS metric is able to incorporate this information [1, 4, 5, 32]. The minimum univariate ESS calculation results in the estimate of the ESS being dominated by the parameter dimensions that mix the slowest, and ignoring all other dimensions [4, 5, 32]. The multivariate ESS is calculated as:

$$\text{ESS} = n \times \left(\frac{|\Lambda|}{|\Sigma|} \right)^{\frac{1}{D}} \quad (13)$$

where n is the number of generated samples, D is the number of parameters, Λ is the sample covariance matrix and Σ is the estimate of the Markov chain standard error. When $D = 1$, the multivariate ESS is equivalent to the univariate ESS measure.

It is worth noting that for the case where $D > 1$, the disadvantage of the multivariate ESS approach is with regards to the estimation of Λ and Σ . If non-stable approaches are used, they can lead to unreliable results [32]. This is particularly important when D is large.

5.3 Sensitivity to the vol-of-vol

In this section, we present the sensitivity analysis for the chosen vol-of-vol parameter α . We considered values of $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, with all the other settings being the same as those outlined in Section 5.1. The results are presented in Fig 1. The results show that the ESS has a tendency of decreasing for both QIHMC and QIMHMC with increasing α on both the Australian and German credit datasets. Furthermore, QIMHMC outperforms QIHMC for all values of α on both an ESS and normalised ESS basis, showing the robust results that can be obtained from QIMHMC.

6 Results and discussion

We present the performance of the MCMC methods outlined above across all the targets distributions using different performance metrics in Fig 2 and Tables 2 to 4. In Fig 2, the plots on the first row for each dataset show the effective sample size, and the plots on the second row show the effective sample size normalised by execution time. The results are for the 10 runs of

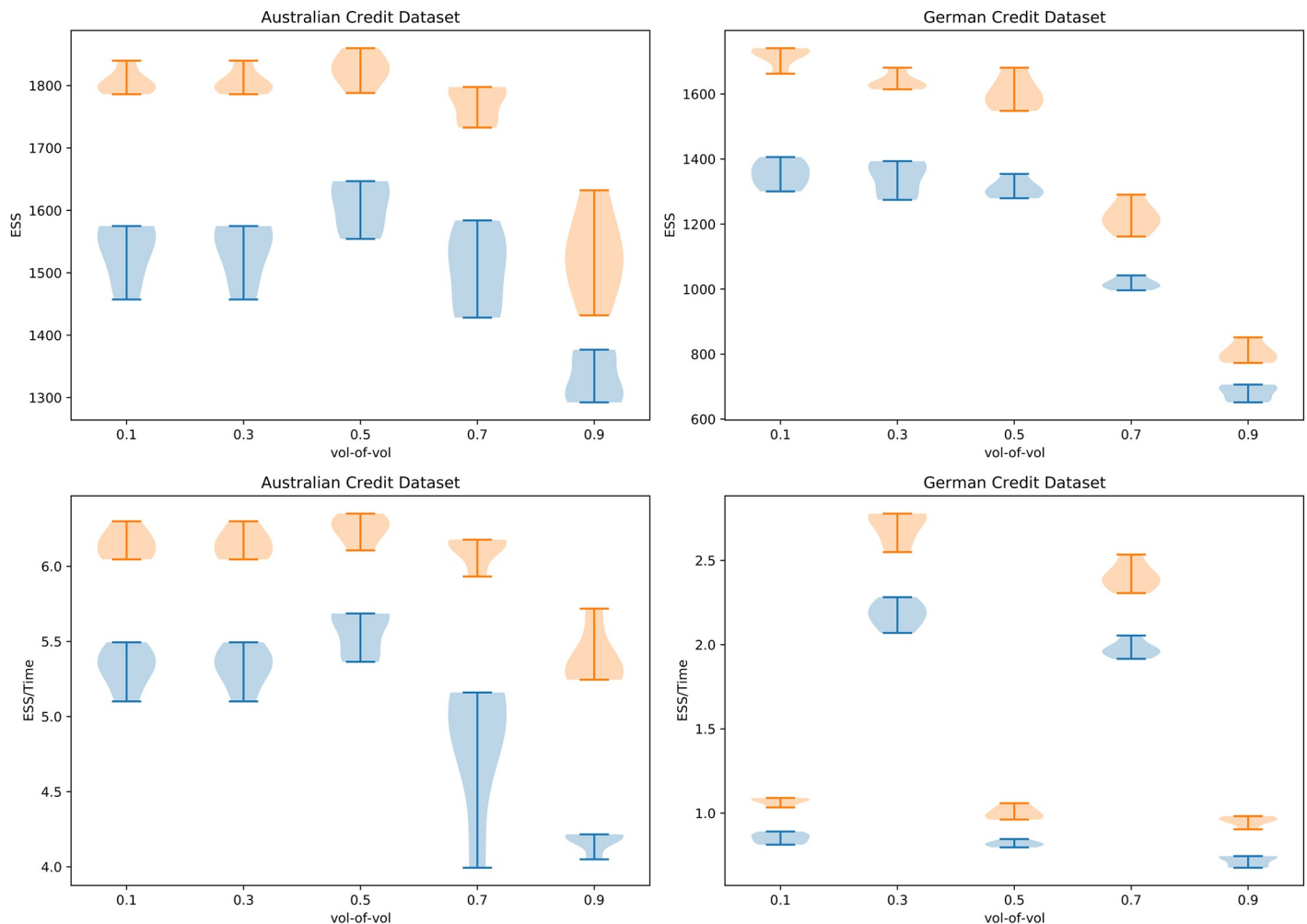


Fig 1. ESS and ESS/ t for ten runs of QIHMC (blue) and QIMHMC (orange) on the Australian and German credit datasets with varying choices of the vol-of-vol parameter.

<https://doi.org/10.1371/journal.pone.0258277.g001>

each algorithm. In Tables 2 to 4, each column corresponds with the results for a particular MCMC method. Values that are in **bold** indicate that the MCMC method outperforms on that particular metric.

The execution time t in Fig 2 and Tables 2 to 4 is in seconds. The results in Tables 2 to 4 are the mean results over the 10 runs for each algorithm. We use the mean values over the 10 runs in Tables 2 to 4 to form our conclusions about the performance of the algorithms.

The results in Fig 2 and Tables 2 to 4 show that the proposed QIMHMC method outperforms all the other methods across the ESS and normalised ESS performance metrics and on all the targets. The outperformance improving with increasing dimensionality of the problem.

As expected, MHMC always outperforms HMC on all the targets and across all the metrics. This is in line with what has been previously observed [4, 6, 27]. MHMC and QIMHMC produce similar performance on the acceptance rate metric.

QIHMC outperforms HMC across all the targets. This confirms the results observed by Liu and Zhang [7] using different target posteriors to those considered in this paper. This shows the significant benefit that utilising a random mass can provide to the sampling properties of HMC based samplers. However, the real performance gains are only realised when the vol-of-

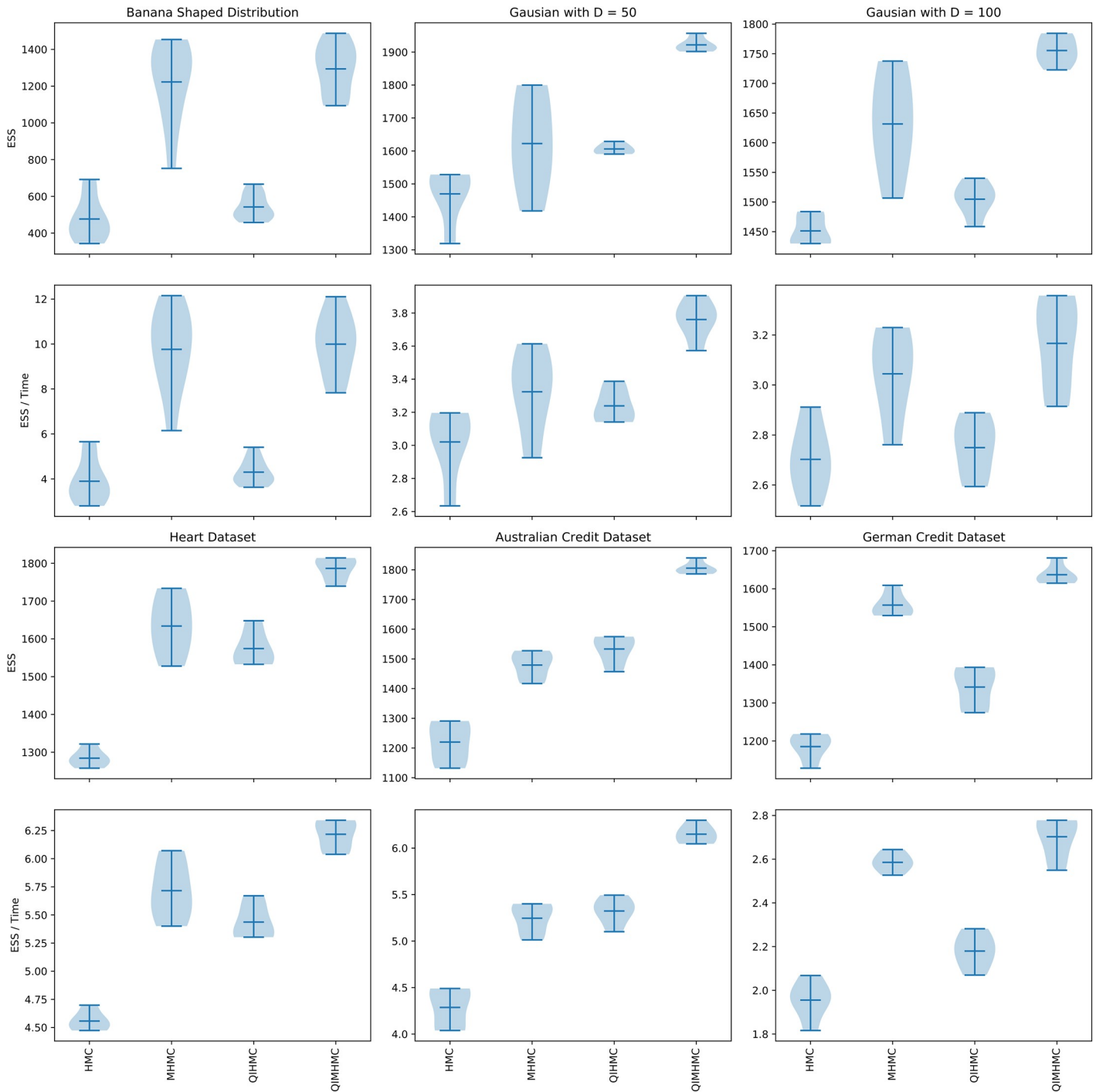


Fig 2. Results for the datasets over 10 runs of each method. For each dataset, the plots on the first row show the multivariate effective sample size and the plots on the second row show the multivariate effective sample size normalised by execution time (in seconds). For all the plots, the larger the value the better the method. The dark horizontal line in each violin plot represents the mean value over 10 runs of each algorithm.

<https://doi.org/10.1371/journal.pone.0258277.g002>

Table 2. Banana shaped distribution results averaged over 10 runs. The time t is in seconds. The values in bold indicate that the particular method outperforms the other methods on that specific metric.

Banana shaped distribution				
Metric	HMC	MHMC	QIHMC	QIMHMC
Acceptance	77.42	99.81	75.52	99.10
ESS	476	1 222	542	1 293
ESS/ t	3.89	9.76	4.30	9.99

<https://doi.org/10.1371/journal.pone.0258277.t002>

Table 3. Multivariate Gaussian distribution results averaged over 10 runs. The time t is in seconds. The values in bold indicate that the particular method outperforms the other methods on that specific metric.

Gaussian with D = 50				
Metric	HMC	MHMC	QIHMC	QIMHMC
Acceptance	78.90	83.03	79.84	89.3
ESS	1 469	1 622	1 606	1 921
ESS/ t	3.02	3.32	3.23	3.76
Gaussian with D = 100				
Metric	HMC	MHMC	QIHMC	QIMHMC
Acceptance	71.82	77.50	71.08	80.24
ESS	1 451	1 631	1 504	1 755
ESS/ t	2.70	3.04	2.75	3.17

<https://doi.org/10.1371/journal.pone.0258277.t003>

Table 4. Bayesian logistic regression results averaged over 10 runs. The time t is in seconds. The values in bold indicate that the particular method outperforms the other methods on that specific metric.

Heart dataset				
Metric	HMC	MHMC	QIHMC	QIMHMC
Acceptance	92.90	99.00	92.48	98.92
ESS	1 284	1 633	1 574	1 786
ESS/ t	4.55	5.71	5.43	6.21
Australian credit dataset				
Metric	HMC	MHMC	QIHMC	QIMHMC
Acceptance	89.25	96.70	89.39	96.58
ESS	1 220	1 479	1 533	1 805
ESS/ t	4.28	5.24	5.32	6.15
German credit dataset				
Metric	HMC	MHMC	QIHMC	QIMHMC
Acceptance	77.92	89.08	75.89	86.27
ESS	1 185	1 557	1 341	1 636
ESS/ t	1.95	2.58	2.17	2.70

<https://doi.org/10.1371/journal.pone.0258277.t004>

vol parameter α has been appropriately tuned. The approach for tuning α is still an open research problem, which we aim to address in future work.

7 Conclusion

We present the Quantum-Inspired Magnetic Hamiltonian Monte Carlo (QIMHMC) method which employs a random mass matrix in the non-canonical dynamics of Magnetic

Hamiltonian Monte Carlo (MHMC). This results in significant sampling improvements over MHMC. The new method is compared to Hamiltonian Monte Carlo (HMC), MHMC and Hamiltonian Monte Carlo with a random mass matrix. The methods are compared on the Banana shaped distribution, multivariate Gaussian distributions and on real world datasets modelled using Bayesian logistic regression.

The empirical results show that the new method outperforms all the other methods on a normalised effective sample size basis across all the targets. Furthermore, HMC with a random mass matrix outperforms vanilla HMC. This shows the significant benefit provided by using a random mass matrix to the sampling properties of HMC based samplers. A limitation of the method is the need to tune the vol-of-vol parameter. Although typically smaller values of the parameter improve the effective sample sizes, a more robust approach to the selection of the parameter is still required.

This work can be improved upon by establishing a heuristic or an automated approach to tune the vol-of-vol parameter. In addition, the tuning of the magnetic component could also be of interest. In future work, we plan to apply the proposed method to the inference of Bayesian neural networks using larger datasets such as MNIST.

Acknowledgments

The computations in this work were performed on resources provided by the Center for High Performance Computing (CHPC) at the Council of Scientific and Industrial Research (CSIR) South Africa.

Author Contributions

Conceptualization: Wilson Tsakane Mongwe.

Data curation: Wilson Tsakane Mongwe.

Formal analysis: Wilson Tsakane Mongwe.

Investigation: Wilson Tsakane Mongwe.

Methodology: Wilson Tsakane Mongwe.

Project administration: Rendani Mbuva, Tshilidzi Marwala.

Software: Wilson Tsakane Mongwe.

Supervision: Rendani Mbuva, Tshilidzi Marwala.

Validation: Wilson Tsakane Mongwe, Rendani Mbuva, Tshilidzi Marwala.

Visualization: Wilson Tsakane Mongwe.

Writing – original draft: Wilson Tsakane Mongwe.

Writing – review & editing: Wilson Tsakane Mongwe, Rendani Mbuva, Tshilidzi Marwala.

funding-acquisition: Tshilidzi Marwala.

resources: Wilson Tsakane Mongwe.

References

1. Girolami M, Calderhead B. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 2011; 73(2):123–214. <https://doi.org/10.1111/j.1467-9868.2010.00765.x>

2. Izaguirre JA, Hampton SS. Shadow Hybrid Monte Carlo: An Efficient Propagator in Phase Space of Macromolecules. *J Comput Phys*. 2004; 200(2):581–604. <https://doi.org/10.1016/j.jcp.2004.04.016>
3. Betancourt M. A conceptual introduction to Hamiltonian Monte Carlo. arXiv preprint arXiv:170102434. 2017.
4. Mongwe WT, Mbuva R, Marwala T. Antithetic Magnetic And Shadow Hamiltonian Monte Carlo. *IEEE ACCESS*. 2021; 9:1–11. <https://doi.org/10.1109/ACCESS.2021.3069196>
5. Mongwe WT, Mbuva R, Marwala T. Antithetic Riemannian Manifold And Quantum-Inspired Hamiltonian Monte Carlo. arXiv preprint arXiv:210702070. 2021.
6. Tripuraneni N, Rowland M, Ghahramani Z, Turner R. Magnetic hamiltonian monte carlo. In: *International Conference on Machine Learning*. PMLR; 2017. p. 3453–3461.
7. Liu Z, Zhang Z. Quantum-Inspired Hamiltonian Monte Carlo for Bayesian Sampling. arXiv preprint arXiv:191201937. 2019.
8. Neal RM. Bayesian learning via stochastic dynamics. In: *Advances in neural information processing systems*; 1993. p. 475–482.
9. Neal RM, et al. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*. 2011; 2(11):2.
10. Neal RM. Bayesian learning for neural networks. vol. 118. Springer Science & Business Media; 2012.
11. Tsaih RH, Lin WY, Huang SY. Exploring Fraudulent Financial Reporting with GHSOM. In: *Intelligence and Security Informatics*. Springer Berlin Heidelberg; 2009. p. 31–41.
12. Radivojević T, Akhmatkaya E. Mix & Match Hamiltonian Monte Carlo. arXiv preprint arXiv:170604032. 2017.
13. Mbuva R, Marwala T. Bayesian inference of COVID-19 spreading rates in South Africa. *PloS one*. 2020; 15(8):e0237126. <https://doi.org/10.1371/journal.pone.0237126> PMID: 32756608
14. Mongwe WT, Malan KM. The Efficacy of Financial Ratios for Fraud Detection Using Self Organising Maps. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE; 2020. p. 1100–1106.
15. Mongwe WT, Malan KM. A Survey of Automated Financial Statement Fraud Detection with Relevance to the South African Context. *South African Computer Journal*. 2020; 32(1). <https://doi.org/10.18489/sacj.v32i1.777>
16. Graham MM, Storkey AJ. Continuously tempered hamiltonian monte carlo. arXiv preprint arXiv:170403338. 2017.
17. Hastings WK. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*. 1970;. <https://doi.org/10.1093/biomet/57.1.97>
18. Neal RM. Slice sampling. *Annals of statistics*. 2003; p. 705–741.
19. Gelfand AE, Smith AF. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*. 1990; 85(410):398–409. <https://doi.org/10.1080/01621459.1990.10476213>
20. Duane S, Kennedy AD, Pendleton BJ, Roweth D. Hybrid monte carlo. *Physics letters B*. 1987; 195(2):216–222. [https://doi.org/10.1016/0370-2693\(87\)91197-X](https://doi.org/10.1016/0370-2693(87)91197-X)
21. Neal RM. Probabilistic inference using Markov chain Monte Carlo methods. University of Toronto, Department of Computer Science; 1993. CRG-TR-93-1.
22. Hoffman MD, Gelman A. The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*. 2014; 15(1):1593–1623.
23. McNaughton B, Milošević M, Perali A, Pilati S. Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks. *Physical Review E*. 2020; 101(5):053312. <https://doi.org/10.1103/PhysRevE.101.053312> PMID: 32575304
24. Levy D, Hoffman MD, Sohl-Dickstein J. Generalizing Hamiltonian Monte Carlo with Neural Networks. In: *International Conference on Learning Representations*; 2018.
25. Betancourt MJ. Generalizing the No-U-Turn Sampler to Riemannian Manifolds; 2013.
26. Gu M, Sun S. Neural Langevin Dynamical Sampling. *IEEE Access*. 2020; 8:31595–31605. <https://doi.org/10.1109/ACCESS.2020.2972611>
27. Brofos JA, Lederman RR. Non-Canonical Hamiltonian Monte Carlo. arXiv preprint arXiv:200808191. 2020.
28. Mongwe WT, Mbuva R, Marwala T. Magnetic Hamiltonian Monte Carlo With Partial Momentum Refreshment. *IEEE Access*. 2021; 9:108009–108016.
29. Heide C, Roosta F, Hodgkinson L, Kroese D. Shadow Manifold Hamiltonian Monte Carlo. In: *International Conference on Artificial Intelligence and Statistics*. PMLR; 2021. p. 1477–1485.

30. Haario H, Saksman E, Tamminen J. Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*. 1999; 14(3):375–395. <https://doi.org/10.1007/s001800050022>
31. Brofos JA, Lederman RR. Magnetic Manifold Hamiltonian Monte Carlo. arXiv preprint arXiv:201007753. 2020.
32. Vats D, Flegal JM, Jones GL. Multivariate output analysis for Markov chain Monte Carlo. *Biometrika*. 2019; 106(2):321–337. <https://doi.org/10.1093/biomet/asz002>