Utilising Partial Momentum Refreshment In Separable Shadow Hamiltonian Hybrid Monte Carlo

WILSON TSAKANE MONGWE¹, RENDANI MBUVHA², AND TSHILIDZI MARWALA.¹, (Senior Member, IEEE)
¹School of Electrical Engineering, University of Johannesburg, Auckland Park, 2000, South Africa
²School of Statistics and Actuarial Science, University of Witwatersrand, Johannesburg, 2006, South Africa
Corresponding author: Wilson Tsakane Mongwe (e-mail: wilsonmongwe@gmail.com).

The work of Wilson Tsakane Mongwe and Rendani Mbuvha was supported by the Google Ph.D. Fellowships in Machine Learning. The work of Tshilidzi Marwala was supported by the National Research Foundation of South Africa.

ABSTRACT Sampling using integrator-dependent shadow Hamiltonian’s has been shown to produce improved sampling properties relative to Hamiltonian Monte Carlo. The shadow Hamiltonian’s are typically non-separable, requiring the expensive generation of momenta, with the recent trend being to utilise partial momentum refreshment. Separable shadow Hamiltonian Hybrid Monte Carlo (S2HMC) employs a canonical transformation which results in the Hamiltonian being separable and makes use of a processed leapfrog integrator. In this work, we combine the benefit of sampling using S2HMC with partial momentum refreshment to create the Separable Shadow Hamiltonian Hybrid Monte Carlo with Partial Momentum Refreshment (PS2HMC) algorithm which leaves the target distribution invariant. Numerical experiments across various targets show that the proposed algorithm outperforms S2HMC and Shadow Hamiltonian Monte Carlo with partial momentum refreshment. Comprehensive analysis is performed on the Banana shaped distribution, multivariate Gaussian distributions of various dimensions, Bayesian logistic regression and Bayesian neural networks.

INDEX TERMS Bayesian Neural Networks, Bayesian Logistic Regression, Hamiltonian Monte Carlo, Partial Momentum Refreshment, Shadow Hamiltonian Monte Carlo, Markov Chain Monte Carlo

I. INTRODUCTION
Markov Chain Monte Carlo (MCMC) methods have been successfully employed to sample from complex statistical and machine learning models [1, 2, 3, 4]. The first MCMC method to be introduced into the literature is the Metropolis-Hastings [5] method, which has since been enhanced using gradient-free MCMC techniques [6, 7, 8] as well as approaches that incorporate the gradient [9, 10, 11] information of the target posterior. MCMC methods have been applied in various contexts including health, renewable energy, finance and inverse problems [12, 13, 14, 15, 16].

A popular MCMC algorithm is Hamiltonian Monte Carlo (HMC) [9, 17, 18], which utilises Hamiltonian dynamics to sample from the target posterior. HMC improves on random walk samplers such as the Metropolis-Hastings algorithm by utilising the first-order gradient information of the unnormalised posterior distribution to guide its exploration. This results in lower auto-correlations between the generated samples when compared to random walk samplers.

There have been various improvements to the HMC algorithm first introduced by Duane et al. [9]. These variations include Magnetic Hamiltonian Monte Carlo [19, 11, 20, 21], which adds a magnetic field to HMC and leads to lower auto-correlations in the generated samples, Riemannian Manifold Hamiltonian Monte Carlo [10] which employs second order gradient information to explore the target, Wormhole Hamiltonian Monte Carlo [22] which efficiently samples from isolated modes of the target distribution by exploiting the Riemannian geometric properties of the target distribution, the No-U-Turn Sampler which addresses a key impediment of HMC which is the tuning of the step size and trajectory length parameters that are very difficult to manually tune [23, 14], as well as methods that use modified or shadow Hamiltonian’s to sample from high dimensional targets [1, 14, 24, 13].

Since the seminal work of Izaguirre and Hampton [1] on integrator-dependent shadow Hamiltonian’s, there has been
a proliferation of shadow Hamiltonian Monte Carlo methods in the literature. The shadow Hamiltonian methods are premised on the fact that shadow Hamiltonian’s are better conserved when compared to the true Hamiltonian’s [1]. This allows one to use larger step sizes, or perform sampling on problems with larger dimensions, without a significant decrease in the acceptance rates when compared to Hamiltonian Monte Carlo methods [25, 26, 27]. The authors introduce a constant, which determines how close the true and the shadow Hamiltonian’s are, to control the generation of the momentum. This increases overall computational time of the method.

Sweet et al. [24] improve on the work of Izaguirre and Hampton [1] by using a canonical transformation on the parameters and momentum. This canonical transformation is substituted into the non-separable Hamiltonian introduced in Izaguirre and Hampton [1] so that it now becomes separable. This results in a processed leapfrog integration scheme which is more computationally efficient when compared to the original shadow Hamiltonian Monte Carlo method of Izaguirre and Hampton [1], as computationally expensive momentum generation for the non-separable Hamiltonian is no longer required.

Partial momentum refreshment has been utilised by Radi-vojevic and Akhmatskay [26] and Akhmatskaya and Reich [25] to generate momenta in the context of non-separable Hamiltonian’s. Radivojevic and Akhmatskay [26] also consider higher order integrators and their corresponding shadow Hamiltonian’s and propose the Mix and Match Hamiltonian Monte Carlo algorithm which provides better sampling properties to HMC.

Heide et al. [27] derive a non-separable shadow Hamiltonian for the generalised leapfrog integrator used in Riemannian Manifold Hamiltonian Monte Carlo (RMHMC), which results in improved performance relative to sampling from the true Hamiltonian. The authors employed partial momentum refreshment to generate the momenta. Partial momentum refreshment has also been used by Horowitz [28] to improve the sampling properties of Hamiltonian Monte Carlo and by Mongwe et al. [21] to enhance the performance of Magnetic Hamiltonian Monte Carlo. The results showed the significant benefits that can be obtained by utilising partial momentum refreshment in Hamiltonian Monte Carlo methods [21, 28]. Employing partial momentum refreshment in S2HMC is yet to be explored in the literature. This manuscript aims to fill this gap in the literature.

In this work, we combine the separable Hamiltonian in S2HMC with partial momentum refreshment to create the Separable Shadow Hamiltonian Hybrid Monte Carlo With Partial Momentum Refreshment (PS2HMC) algorithm, which is shown to converge to the correct unnormalised posterior distribution. The performance of the proposed sampler is compared against Separable Shadow Hamiltonian Hybrid Monte Carlo and Shadow Hamiltonian Monte Carlo with partial momentum refreshment. The target posteriors considered are the Banana shaped distribution, multivariate Gaussian distributions with various dimensions, real world datasets modelled using Bayesian logistic regression and Bayesian neural networks.

The empirical results show that the PS2HMC method outperforms the other MCMC algorithms on all the target distributions based on the effective sample size, effective sample size normalised by execution time and the acceptance rate metrics.

The main contributions of this work can be summarised as follows:

- We introduce the Separable Shadow Hamiltonian Hybrid Monte Carlo with Partial Momentum Refreshment (PS2HMC) method, which utilises partial momentum refreshment in Separable Shadow Hamiltonian Hybrid Monte Carlo and converges to correct posterior distribution.
- Numerical experiments show that the PS2HMC algorithm outperforms Separable Shadow Hamiltonian Hybrid Monte Carlo and Shadow Hamiltonian Hybrid Monte Carlo with partial momentum refreshment on all the performance metrics considered.

The remainder of this manuscript proceeds as follows: Section II discuss the Markov Chain Monte Carlo considered in this work, Section III presents the proposed method, Section IV outlines the target densities considered, Section V outlines the experiments conducted, Section VI presents and discusses the results of the experiments and we provide the conclusion in Section VII.

II. SHADOW HAMILTONIAN MONTE CARLO

Hamiltonian Monte Carlo (HMC) employs Hamiltonian dynamics to efficiently explore the parameter space [9, 18, 29]. HMC adds an auxiliary momentum variable \( p \in \mathbb{R}^D \) to the parameter space \( w \in \mathbb{R}^D \), where \( D \) is the number of dimensions. The resultant Hamiltonian \( H : \mathbb{R}^{2D} \rightarrow \mathbb{R} \) from this dynamic system is written as [18]:

\[
H(w, p) = U(w) + K(p)
\]

(1)

where \( U(w) \) is the negative log-likelihood of a differentiable target posterior distribution and \( K(p) \) is the kinetic energy defined by the kernel of a Gaussian with a covariance matrix \( M \) [21, 18, 29]:

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

(2)

The trajectory vector field is defined by considering the parameter space as a physical system that follows Hamiltonian dynamics [21, 18]. The equations governing the trajectory of the chain are then defined by Hamilton’s equations at a fictitious time \( t \) as follows [18]:

\[
\frac{dw}{dt} = \frac{\partial H(w, p)}{\partial p}; \quad \frac{dp}{dt} = -\frac{\partial H(w, p)}{\partial w}.
\]

(3)

The evolution of this Hamiltonian system must preserve both volume and total energy [21]. As the Hamiltonian in equation (1) is separable, to traverse the space we use the leapfrog

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3126812, IEEE Access

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
The leapfrog integration scheme proceeds as follows: the next point on the trajectory is reached by taking a half step in the momentum direction, followed by a full step in the parameters and concluding with a half step in the momentum direction [21, 18]. Mathematically, this is expressed as:

$$\mathbf{p}_{t+\epsilon} = \mathbf{p}_t + \epsilon \frac{\partial H(\mathbf{w}_t, \mathbf{p}_t)}{\partial \mathbf{p}}$$

$$\mathbf{w}_{t+\epsilon} = \mathbf{w}_t + \epsilon \mathbf{M}^{-1} \mathbf{p}_{t+\epsilon}$$

$$\mathbf{p}_{t+\epsilon} = \mathbf{p}_{t+\epsilon} + \epsilon \frac{\partial H(\mathbf{w}_{t+\epsilon}, \mathbf{p}_{t+\epsilon})}{\partial \mathbf{w}}.$$ 

(4)

A Metropolis-Hastings acceptance step is then performed in order to take into account the discretisation errors introduced by the numerical integration scheme.

The leapfrog integrator for HMC only preserves the Hamiltonian up to second order [1, 24]. In order to increase accuracy and maintain the acceptance rate for larger systems, one could decrease the step size or design more accurate numerical integrators that preserve the Hamiltonian to a higher order [26, 27]. However, these approaches tend to be computationally expensive [26, 27]. The approach in this work relies on backwards error analysis to instead derive a shadow Hamiltonian, whose energy is more accurately conserved by the leapfrog algorithm. We thus instead target the corresponding modified density and employ importance sampling to correct the generated samples towards the true density [26, 27].

Shadow or modified Hamiltonian’s are perturbations of the Hamiltonian that are by design exactly conserved by the numerical integrator [26, 1, 3, 13]. In the case of shadow Hamiltonian Hybrid Monte Carlo, we sample from the importance distribution defined by the shadow Hamiltonian

$$\pi \propto \exp \left( - \tilde{H}^{[k]}(\mathbf{w}, \mathbf{p}) \right)$$

(5)

where $\tilde{H}^{[k]}$ is the shadow Hamiltonian defined using backwards error analysis of the numerical integrator up to the $k$th order [13, 3]. When performing backward error analysis, the shadow Hamiltonian can be defined by an asymptotic expansion in the powers of the discretisation step size $\epsilon$ around the Hamiltonian:

$$\tilde{H} = H_0 + \epsilon H_1 + \epsilon^2 H_2 + \epsilon^3 H_3 + \ldots.$$ 

(6)

This asymptotic expansion diverges in practice, however a $k$th order truncation of the expansion is used:

$$\tilde{H}^{[k]} = H + \epsilon H_2 + \epsilon^2 H_3 + \epsilon^3 H_4 + \ldots = \tilde{H} + \mathcal{O}(\epsilon^k)$$

(7)

The $H_k$ terms can be determined by matching the corresponding components of the Taylor series in terms of $\epsilon$ and the expanded exact flow of the modified differential equation of the Hamiltonian. These modified equations can be proved to be Hamiltonian for symplectic integrators such as the leapfrog integrator [24, 1, 3, 13].

In this work, we focus on a fourth-order truncation of the shadow Hamiltonian under the leapfrog-like integrator.

Since the leapfrog is second-order accurate ($\mathcal{O}(\epsilon^2)$), the fourth-order truncation is conserved with higher accuracy ($\mathcal{O}(\epsilon^4)$) than the true Hamiltonian. In theorem II.1, we derive the fourth-order shadow Hamiltonian Monte Carlo under the leapfrog integrator.

**Theorem II.1.** Let $H : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a smooth Hamiltonian function. The fourth-order shadow Hamiltonian function $\tilde{H} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ corresponding to the leapfrog integrator of HMC is given by:

$$\tilde{H}(\mathbf{w}, \mathbf{p}) = H(\mathbf{w}, \mathbf{p}) + \frac{\epsilon^2}{12} [K_p U_{\mathbf{w} \mathbf{w}} K_p] - \frac{\epsilon^2}{24} [U_{\mathbf{w}} K_{pp} U_{\mathbf{w}}] + \mathcal{O}(\epsilon^4)$$

(8)

**Proof.** The Hamiltonian vector field:

$$\tilde{H} = \nabla_{\mathbf{p}} H \nabla_{\mathbf{w}} + (-\nabla_{\mathbf{w}} + \nabla_{\mathbf{p}} H) \nabla_{\mathbf{p}} = \tilde{A} + \tilde{B}$$

(9)

will generate the exact flow corresponding to exactly simulating the HMC dynamics. We obtain the shadow density by simply exploiting the separability of the Hamiltonian. The leapfrog integration scheme in equation (4) splits the Hamiltonian as:

$$H(\mathbf{w}, \mathbf{p}) = H_1(\mathbf{w}) + H_2(\mathbf{p}) + H_3(\mathbf{w})$$

(10)

and exactly integrates each sub-Hamiltonian.

Via repeated application of the Baker-Campbell-Hausdorff formula we obtain [30]:

$$\Phi^{\text{frac}}_{\epsilon, H} = \Phi_{\epsilon, H_1} \circ \Phi_{\epsilon, H_2} \circ \Phi_{\epsilon, H_3}$$

$$= \exp \left( \frac{\epsilon}{2} \tilde{B} \right) \exp \left( \epsilon \tilde{A} \right) \exp \left( \frac{\epsilon}{2} \tilde{B} \right)$$

$$= H(\mathbf{w}, \mathbf{p}) + \frac{\epsilon^2}{12} \left( K, \{ K, U \} \right) - \frac{\epsilon^2}{24} \left( U, \{ U, K \} \right) + \mathcal{O}(\epsilon^4)$$

(11)

where the canonical Poisson brackets are defined as:

$$\{ f, g \} = \left[ \nabla_w f, \nabla_p g \right] \left[ \begin{array}{cc} 0 & I \\ -I & 0 \end{array} \right] \left[ \begin{array}{c} \nabla_w g \\ -\nabla_p g \end{array} \right]$$

(12)

The shadow Hamiltonian for the leapfrog integrator is then:

$$\tilde{H}(\mathbf{w}, \mathbf{p}) = H(\mathbf{w}, \mathbf{p}) + \frac{\epsilon^2}{12} [K_p U_{\mathbf{w} \mathbf{w}} K_p] - \frac{\epsilon^2}{24} [U_{\mathbf{w}} K_{pp} U_{\mathbf{w}}] + \mathcal{O}(\epsilon^4)$$

(13)

It is worth noting that the shadow Hamiltonian in (13) is conserved to fourth-order [27, 24, 26].

Separable Shadow Hamiltonian Hybrid Monte Carlo (S2HMC) utilises a processed leapfrog integrator to create a separable Hamiltonian [24, 13, 3]. The separable Hamiltonian in S2HMC is given as:

$$\tilde{H}(\mathbf{w}, \mathbf{p}) = U(\mathbf{w}) + K(\mathbf{p}) + \frac{\epsilon^2}{24} U^T M^{-1} U + \mathcal{O}(\epsilon^4)$$

(14)
which is obtained by substituting a canonical transformation \((w, p) = \mathcal{X}(w, p)\) into (13). The map should commute with reversal of momenta and should preserve phase space volume so that the resulting S2HMC ensures detailed balance [24, 13, 3]. Propagation of positions and momenta across various performance metrics. This approach is yet to be explored in the literature.

The aim of creating a new sampler than outperforms S2HMC [24, 13, 3]. In this work, we combine these two concepts with a \(\text{Separable Shadow Hamiltonian Hybrid Monte Carlo With Partial Momentum Refreshment (PS2HMC)}\) algorithm. This method combines the Separable Shadow Hamiltonian Hybrid Monte Carlo (S2HMC) sampler leaves the target density invariant [27, 25]. This necessitates computationally intensive momentum generation [1] or partial momentum refreshment [26, 25].

In this work, we utilise the partial momentum refreshment technique outlined in [21, 25, 27] in which an auxiliary noise vector \(u \sim \mathcal{N}(0, M)\) is drawn and a momentum proposal is generated via the mapping:

\[
R(p, u) = \left( \rho p + \sqrt{1 - \rho^2} u, -\sqrt{1 - \rho^2} p + \rho u \right)
\]

The new parameter, which we refer to as the momentum refreshment parameter, \(\rho = \rho(w, p, u)\) takes values between zero and one and controls the extent of the momentum retention [21, 26]. When \(\rho\) is equal to one, the momentum is never updated and when \(\rho\) is equal to zero, the momentum is always updated [21]. The momentum proposals are then accepted according to the modified non-separable Shadow Hamiltonian density as in [21]. This raises the question of whether or not the method depends on the momentum and the position [25, 27, 21]. In Section V-C, we assess the sensitivity of the sampling results on the user specified value of \(\rho\).

An algorithmic description of the PS2HMC sampler is provided in Algorithm 1. The algorithm proceeds by first generating a proposal of the momentum, which is accepted or rejected via a Metropolis-Hastings step, and concludes by generating the positions before applying another Metropolis-Hastings step. Thus, the method employs a composition of two reversible Metropolis–Hastings steps, which implies that the resulting Markov chain is no longer reversible [27, 25, 26, 21]. This raises the question of whether or not the method converges to the correct target distribution [26, 21]. Theorem III.1 provides this guarantee.

**Theorem III.1.** The PS2HMC sampler leaves the target density invariant.

The proof of theorem III.1 is taken from [32, 26] after observing that the explicit form of the shadow density is not required for the proof [27, 21]. This means that the shadow density can be of any form.

It is worth noting that the momentum regeneration scheme used in Shadow Hamiltonian Monte Carlo with partial momentum refreshment (SHMC) in this work is the same as the one outlined in the Algorithm 1.

The proposed method involves a minimum (depending on how the gradients are calculated) of 14 likelihood or target posterior function evaluation to generate a single sample, this

\[\text{Hessian approximated as: } U_{ww} K_p = \frac{1}{2} \left[ U_w (w + \epsilon M^{-1} \hat{p}) - U_w (w - \epsilon M^{-1} \hat{p}) \right]\]
is much larger than the minimum of 4 evaluations required by standard HMC. However, it should be noted that the proposed PSHMC method only adds two more likelihood evaluations compared to S2HMC (on which the method is based) and S2HMC has already been shown in [24, 14] to outperform HMC. In this work, we show that our proposed PS2HMC method outperforms S2HMC and consequently outperforms HMC.

**Algorithm 1:** PS2HMC

| Input: maximum number of steps $L$, step size $\epsilon$, momentum update parameter $\rho$, number of Monte Carlo samples $N$ and initial values $(w_0, p_0)$. |
| --- |
| 1: for $i \rightarrow 1$ to $N$ do |
| 2: store: $(w, p) \leftarrow (w_{i-1}, p_{i-1})$. |
| 3: sample momentum update proposal: $u \sim \mathcal{N}(0, M)$. |
| 4: update: $p \leftarrow pp + \sqrt{1-\rho^2}u$ with probability $\gamma$ in equation (19). |
| 5: Apply the pre-processing mapping in equation (16): $(\tilde{w}, \tilde{p}) = \mathcal{X}(w, p)$. |
| 6: integrate Hamiltonian dynamics in equation (4): $(\tilde{w}, \tilde{p}) = \Phi^L_{\epsilon,H}(\tilde{w}, \tilde{p})$. |
| 7: Apply the post-processing mapping in equation (17): $(\tilde{w}, \tilde{p}) = \mathcal{X}^{-1}(\tilde{w}, \tilde{p})$. |
| 8: accept sample $(w_i, p_i) \leftarrow (\tilde{w}, \tilde{p})$ with probability $\beta$, and reject $(w_i, p_i) \leftarrow (w, -p)$ otherwise. Here, $\beta = \min(1, \exp(-\Delta H))$. |
| 9: $b_i = \exp(\tilde{H}(w_i, p_i) - H(w_i, p_i))$. |
| 10: end for |
| Output: $(w_i, p_i, b_i)_{i=0}^{N}$ |

**IV. THE TARGET POSTERIORS**

In this section, we outline the target distributions that were considered in this work. These target distributions are inline with those used in Mongwe et al. [21], with the additional target considered being Bayesian neural networks.

**A. BANANA SHAPED DISTRIBUTION**

The banana-shaped density is a 2-dimensional non-linear target which was first presented in Haario et al. [8]. The likelihood and prior distributions are given as:

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

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

$$\int_{i=1}^{N} p(y_k|w)p(w_1)p(w_2).$$

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

**B. MULTIVARIATE GAUSSIAN DISTRIBUTIONS**

We follow the approach of Mongwe et al. [21] and sample from $D$-dimensional Gaussian distributions $\mathcal{N}(0, \Sigma)$ with mean zero and covariance matrix $\Sigma$. For our purposes, we set the covariance matrix $\Sigma$ to be diagonal. We sample the standard deviations from a log-normal distribution with zero mean and unit standard deviation. We assess the case where the number of dimensions $D$ is in the set $\{10, 50, 100\}$.

**C. BAYESIAN LOGISTIC REGRESSION**

We utilise Bayesian logistic regression to model the real world binary classification datasets in Table 1. The negative log-likelihood $l(D|w)$ function for logistic regression is given as:

$$l(D|w) = \sum_{i} y_i \log(w^T x_i) + (1 - y_i) \log(1 - w^T x_i)$$

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

$$ln(p(w|D)) = l(D|w) + ln(p(w|\alpha))$$

where $ln(p(w|\alpha))$ is the log of the prior distribution on the parameters given the hyperparameters $\alpha$. The parameters $w$ are modelled as having Gaussian prior distributions with zero mean and standard deviation $\alpha = 10$.

**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   |
| South African fraud [33, 34] | 14     | 1 560 | 15   |
| German credit | 24       | 1 000 | 25   |

**D. BAYESIAN NEURAL NETWORKS**

Artificial neural networks are learning machines that have been extensively employed as universal approximators of complex systems with great success [13, 3]. This work focuses on Multilayer Perceptrons (MLP) with one hidden layer, an example of which is shown in Figure 1. In this paper, MLPs are used to model the real world datasets outlined in Table 2. These datasets are regression datasets, with the negative log-likelihood being the sum of squared errors.

It was shown in [35] that MPLs can be utilised to approximate any arbitrary function if the MLP has enough hidden units [36]. The outputs of a network with a single output as depicted in Figure 1 are defined as:

$$f_k(x) = b_k + \sum_j v_{jk} h_j(x)$$

$$h_j(x) = \Psi(a_j + \sum_i w_{ij} x_i)$$

where $w_{ij}$ is the weight connection for the $i^{th}$ input to the $j^{th}$ hidden unit and $v_{jk}$ is the weight connection between...
In practice, functions are examples of common activation functions used between inputs and outputs. The Sigmoid and ReLU activation functions are required to approximate complex non-linear relationships. The \( k = 1 \) hidden unit to the \( k^{th} \) output. The activation function \( \Psi \) provides the non-linearity required to approximate complex non-linear relationships between inputs and outputs. The Sigmoid and ReLU activation functions are examples of common activation functions used in practice [37, 13].

The Bayesian framework provides a principled approach for the inference of neural network model parameters. The Bayesian framework is strongly tied to Bayes theorem. Employing Bayes theorem for a neural network model with architecture \( H \), weights \( w \) and training dataset \( D \), we have [38, 39, 13]:

\[
P(w|D, H) = \frac{P(D|w, H)P(w|H)}{P(D|H)} \tag{26}
\]

where \( P(w|D, H) \) is the target posterior probability of the weights given the data and model architecture, \( P(D|w, H) \) is the likelihood of the data given the model and \( P(w|H) \) is the prior probability of the weights. \( P(D|H) \) is the probability of the data given the model [13, 38].

**TABLE 2**: Real world datasets modelled using Bayesian neural networks. \( N \) represents the number of observations. \( D \) represents the number of model parameters.

| Dataset  | Features | \( N \)   | \( D \) |
|---------|----------|----------|--------|
| Airfoil | 5        | 1503     | 36     |
| Concrete| 8        | 1030     | 51     |

**V. EXPERIMENTAL SETUP**

In this section we outline the experimental setup. We present the settings used for the experiments, the performance metrics employed in the analysis and we present the sensitivity analysis of the sampling results for a user chosen value of \( \rho \).

**A. EXPERIMENT SETTINGS**

In the experiments, we assess the performance of PS2HMC when compared to S2HMC and Shadow Hamiltonian Monte Carlo with partial momentum refreshment. The MCMC methods are compared using the effective sample size, effective sample size per second and the acceptance rate metrics. We set the momentum refreshment parameter \( \rho \) to 0.9 across all the targets. We further assess the effect of \( \rho \) on the sampling results on Section V-C.

We set the trajectory length for the three MCMC methods considered in this work to 100 across all the target densities. A step size of 0.1 was used for the Banana distribution, 0.02 for the Bayesian logistic regression datasets, 0.005 for the Bayesian neural network datasets while step sizes of 0.1, 0.07 and 0.05 were used for each value of \( D \) in that order for the multivariate Gaussian distributions.

A total of ten independent chains were run for each MCMC algorithm across all the target posterior distributions. For the Bayesian neural network targets, 5 000 samples were generated after 2 500 samples of burn-in. For the other targets, we generated 3 000 samples for each target, with the first 1 000 being discarded as the burn-in. These settings were sufficient for the considered MCMC methods to converge across all the posteriors. All experiments were conducted on a 64bit CPU using PyTorch.

**B. EFFECTIVE SAMPLE SIZE**

This work employs the multivariate effective sample size metric developed by Vats et al. [40] 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 [3, 40, 10, 4]. 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 [40, 3]. The multivariate ESS is calculated as:

\[
mESS = N \times \left( \frac{|\Lambda|}{|\Sigma|} \right)^{\frac{1}{2}}
\]

where \( N \) is the number of generated samples, \( D \) is the number of parameters, \( \Lambda \) is the sample covariance matrix and \( \Sigma \) is the estimate of the Markov chain standard error. When \( D = 1 \), the multivariate ESS is equivalent to the univariate ESS measure.

We now address the effective sample size calculation for Markov chains that have been re-weighted via importance sampling, such is the case for the MCMC algorithms considered in this paper [26, 27, 3, 13]. For \( N \) samples re-weighted by importance sampling, the common approach is to use the approximation by [41] given by

\[
ESS_{IMP} = \frac{1}{\left( \sum_{j=1}^{N} \tilde{b}_j^2 \right)} \tag{27}
\]
where \( \bar{b}_j = b_j / \sum_{k=1}^{N} b_k \). This accounts for the possible non-uniformity in the importance sampling weights. In order to account for both the effects of sample auto-correlation and re-weighting via importance sampling, we approximated the effective sample size under importance sampling by:

\[
\text{ESS} := \frac{\text{ESS}_{IMP}}{N} \times m\text{ESS} = \frac{1}{\left(\sum_{j=1}^{N} \bar{b}_j^2\right)^{\frac{1}{2}}} \left(\frac{|A|}{|\Sigma|}\right)^\frac{1}{2} \tag{28}
\]

### C. IMPACT OF PARTIAL MOMENTUM REFRESHMENT

In this section, we assess the impact of the momentum refreshment parameter \( \rho \) on the sampling results. We ran a total of ten independent chains of SHMC and PS2HMC on the ten-dimensional multivariate Gaussian distribution for \( \rho \in \{0.1, 0.3, 0.5, 0.6, 0.7, 0.9\} \). Each chain utilised the same simulation parameters as outlined in Section V. The results are displayed in Figure 2. The results show that SHMC and PS2HMC produce the same acceptance rates regardless of the value of \( \rho \). PS2HMC outperforms SHMC across all the considered performance metrics.

The MCMC methods show a general trend of increasing ESS and normalised ESS with increasing \( \rho \). The optimal \( \rho \) depends on the target distribution and requires manual tuning by the user. The tuning of this parameter is still an open research problem [21]. We plan to address the automatic tuning of this parameter in future work. As a guideline, higher values of \( \rho \) seem to be correlated with higher effective sample sizes. These results were also observed by Mongwe et al. [21] in the context of sampling from Magnetic Hamiltonian Monte Carlo.

### VI. RESULTS AND DISCUSSION

We now present and discuss the results of the experiments outlined in Section V. The performance of the MCMC methods across the various metrics are presented in Figure 3 and Tables 3 to 6. Note that the results in Figure 3 are presented as follows: the plots on the first row for each target distribution show the effective sample size while the plots on the second row show the effective sample size normalised by execution time \( t \). The displayed results are for a total of ten independent runs of each MCMC algorithm.

The execution time \( t \) in Figure 3 and Tables 3 to 6 is in seconds. The results in Tables 3 to 6 are the mean results over the ten runs for each MCMC algorithm. Note that we use the mean values over the ten runs in Tables 3 to 6 to form our conclusions about the performance of the MCMC algorithms.

The results in Figure 3 and Tables 3 to 6 show that the proposed PS2HMC method outperforms the other MCMC methods considered in this work on an effective sample size basis across all the target posteriors considered. Furthermore, it outperforms all the other methods on a normalised effective sample size basis, or produces similar results to the other methods.

The proposed method has a higher computational burden when compared to the other MCMC methods due to the extra evaluations of the shadow Hamiltonian. This results in the normalised effective sample size performance being affected - although still outperforming or producing similar performance to the other MCMC methods. We intend to address this short coming in future work by utilising surrogate methods such as Gaussian processes by learning the shadow Hamiltonian during the burn-in phase [42].

These empirical results show the significant benefit that can be derived from utilising partial momentum refreshment in S2HMC. However, the full benefits of employing partial momentum refreshment in S2HMC are only obtained when the momentum refreshment parameter \( \rho \) has been optimally tuned.

### VII. CONCLUSION

This work introduced the Separable Shadow Hamiltonian Hybrid Monte Carlo With Partial Momentum Refreshment (PS2HMC) sampler which combines the benefits of sampling posteriors using Separable Shadow Hamiltonian Hybrid Monte Carlo (S2HMC) with partially updating the auxiliary momentum variable. This results in significant sampling improvements over S2HMC without partial momentum refreshment. The new method is compared to S2HMC and Shadow Hamiltonian Monte Carlo utilising partial momenta.

---

**TABLE 3:** 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.

| Metric         | S2HMC | SHMC | PS2HMC |
|----------------|-------|------|--------|
| Acceptance     | 0.90  | 0.88 | **0.94** |
| ESS            | 1 137 | 1 237| **1 376** |
| ESS/t          | 7.48  | 7.91 | **8.65** |

**TABLE 4:** 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.

| Metric         | S2HMC | SHMC | PS2HMC |
|----------------|-------|------|--------|
| Acceptance     | 0.83  | 0.85 | **0.89** |
| ESS            | 1 589 | 3 111| **3 682** |
| ESS/t          | 2.50  | 4.01 | **4.78** |

| Metric         | S2HMC | SHMC | PS2HMC |
|----------------|-------|------|--------|
| Acceptance     | 0.76  | 0.79 | **0.88** |
| ESS            | 1 649 | 2 749| **3 661** |
| ESS/t          | 2.24  | 2.71 | **3.78** |
FIGURE 2: Acceptance rates, ESS and ESS/t for ten runs of SHMC (blue) and PS2HMC (orange) on the 10 dimensional Gaussian distribution with varying choices of ρ.

FIGURE 3: 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.
performs the momentum refreshment. We plan to address
is due to the evaluation on the shadow Hamiltonian when
time when compared to S2HMC. The larger execution time
algorithms.

The empirical results show that the new method outper-
that specific metric.

| Metric         | S2HMC | SHMC | PS2HMC |
|---------------|-------|------|--------|
| Acceptance    | 0.88  | 0.85 | 0.90   |
| ESS           | 1 364 | 2 912| 3 067  |
| ESS/t         | 4.17  | 8.16 | 8.71   |

| Metric         | S2HMC | SHMC | PS2HMC |
|---------------|-------|------|--------|
| Acceptance    | 0.88  | 0.86 | 0.89   |
| ESS           | 2 570 | 2 768| 3 755  |
| ESS/t         | 1.60  | 1.84 | 2.17   |

| Metric         | S2HMC | SHMC | PS2HMC |
|---------------|-------|------|--------|
| Acceptance    | 0.77  | 0.76 | 0.84   |
| ESS           | 1 019 | 1 611| 2 089  |
| ESS/t         | 0.42  | 0.82 | 0.85   |

| Metric         | S2HMC | SHMC | PS2HMC |
|---------------|-------|------|--------|
| Acceptance    | 0.78  | 0.72 | 0.83   |
| ESS           | 1 413 | 2 372| 3 126  |
| ESS/t         | 0.68  | 1.33 | 1.52   |

tum refreshment. The methods are compared on the Banana
shaped distribution, multivariate Gaussian distributions and
on real world datasets modelled using Bayesian logistic
regression and Bayesian neural networks. The empirical results show that the new method outper-
that all the target posteriors considered. Furthermore, it
outperforms all the other methods on a normalised effective
sample size basis, or produces similar results to the other
algorithms.

A limitation of the proposed method is the larger execution
time when compared to S2HMC. The larger execution time
is due to the evaluation on the shadow Hamiltonian when
performing the momentum refreshment. We plan to address
this in future work by utilising surrogate approaches for the
shadow Hamiltonian, which should reduce the execution time
without a large reduction in the performance of the algorithm. Another limitation of the new algorithm is the need for the
user to manually specify the momentum refreshment parameter. The user would be required to perform trial and error runs
to select an appropriate value for the parameter. The results in
this work suggest that larger values of the momentum refreshment parameter tend to improve the effective sample size. However, a more theoretically robust approach to the selection of the parameter is required.

This work can be improved upon by investigating an
automated approach to optimally tune the momentum refreshment parameter and thus removing the need for the user
to perform trial and error. As future work, we plan to compare
the performance of the proposed algorithm against manifold
based shadow Hamiltonian methods on larger datasets.
Furthermore, we plan on using surrogate model approaches to
reduce the execution time of the proposed method.

TABLE 6: Bayesian neural network 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.

| Metric       | S2HMC | SHMC | PS2HMC |
|--------------|-------|------|--------|
| Acceptance   | 0.95  | 0.95 | 0.96   |
| ESS          | 231   | 601  | 615    |
| ESS/t        | 0.17  | 0.22 | 0.31   |

| Metric       | S2HMC | SHMC | PS2HMC |
|--------------|-------|------|--------|
| Acceptance   | 0.96  | 0.96 | 0.97   |
| ESS          | 278   | 606  | 611    |
| ESS/t        | 0.38  | 0.38 | 0.67   |

ACKNOWLEDGEMENTS

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.

REFERENCES

[1] J. A. Izaguirre and S. S. Hampton, “Shadow hybrid
monte carlo: An efficient propagator in phase space of
macromolecules,” J. Comput. Phys., vol. 200, no. 2, pp.
581–604, Nov. 2004.
[2] J. A. Brofos and R. R. Lederman, “Magnetic
manifold hamiltonian monte carlo,” arXiv preprint
arXiv:2010.07753, 2020.
[3] W. T. Mongwe, R. Mbuva, and T. Marwala, “Antithetic
magnetic and shadow hamiltonian monte carlo,” IEEE
Access, vol. 9, pp. 49 857–49 867, 2021.
[4] ——, “Antithetic riemannian manifold and quantum-
inspired hamiltonian monte carlo,” arXiv preprint
arXiv:2107.02070, 2021.
[5] W. K. Hastings, “Monte carlo sampling methods using
markov chains and their applications,” 1970.
[6] M. Bédard, R. Douc, and E. Moulines, “Scaling anal-
ysis of multiple-try mcmc methods,” Stochastic Pro-
cesses and their Applications, vol. 122, no. 3, pp.
758–786, 2012.
[7] L. Martino, “A review of multiple try mcmc algo-
rithms for signal processing,” Digital Signal Process-
ing, vol. 75, pp. 134–152, 2018.
[8] H. Haario, E. Saksman, and J. Tamminen, “Adaptive
proposal distribution for random walk metropolis al-
gorithm,” Computational Statistics, vol. 14, no. 3, pp.
375–395, 1999.
[9] S. Duane, A. D. Kennedy, B. J. Pendleton, and
D. Roweth, “Hybrid monte carlo,” Physics letters B,
vol. 195, no. 2, pp. 216–222, 1987.
[10] M. Girolami and B. Calderhead, “Riemann manifold langevin and hamiltonian monte carlo methods,” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 73, no. 2, pp. 123–214, 2011.

[11] N. Tripuraneni, M. Rowland, Z. Ghahramani, and R. Turner, “Magnetic hamiltonian monte carlo,” in International Conference on Machine Learning. PMLR, 2017, pp. 3453–3461.

[12] R. Mbuvha and T. Marwala, “Bayesian inference of covid-19 spreading rates in south africa,” PloS one, vol. 15, no. 8, p. e0237126, 2020.

[13] R. Mbuvha, W. T. Mongwe, and T. Marwala, “Separable shadow hamiltonian hybrid monte carlo for bayesian neural network inference in wind speed forecasting,” Energy and AI, p. 100108, 2021.

[14] W. T. Mongwe, R. Mbuvha, and T. Marwala, “Adaptively setting the path length for separable shadow hamiltonian hybrid monte carlo,” IEEE Access, vol. 9, pp. 138 598–138 607, 2021.

[15] G. Bal, I. Langmore, and Y. Marzouk, “Bayesian inverse problems with monte carlo forward models,” Inverse Problems & Imaging, vol. 7, no. 1, p. 81, 2013.

[16] Y. Xia and N. Zabaras, “Bayesian multiscale deep generative model for the solution of high-dimensional inverse problems,” arXiv preprint arXiv:2102.03169, 2021.

[17] S. Duane and J. B. Kogut, “The theory of hybrid stochastic algorithms,” Nuclear Physics B, vol. 275, no. 3, pp. 398–420, 1986.

[18] R. M. Neal, Bayesian learning for neural networks. Springer Science & Business Media, 2012, vol. 118.

[19] W. T. Mongwe, R. Mbuvha, and T. Marwala, “Quantum-inspired magnetic hamiltonian monte carlo,” Plos one, vol. 16, no. 10, p. e0258277, 2021.

[20] J. A. Brofos and R. R. Lederman, “Non-canonical hamiltonian monte carlo,” arXiv preprint arXiv:2008.08191, 2020.

[21] W. T. Mongwe, R. Mbuvha, and T. Marwala, “Magnetic hamiltonian monte carlo with partial momentum refreshment,” IEEE Access, vol. 9, pp. 108 009–108 016, 2021.

[22] S. Lan, J. Streets, and B. Shahbaba, “Wormhole hamiltonian monte carlo,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, no. 1, 2014.

[23] M. D. Hoffman and A. Gelman, “The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo,” Journal of Machine Learning Research, vol. 15, no. 1, pp. 1593–1623, 2014.

[24] C. R. Sweet, S. S. Hampton, R. D. Skeel, and J. A. Izaguire, “A separable shadow hamiltonian hybrid monte carlo method,” The Journal of chemical physics, vol. 131, no. 17, p. 174106, 2009.

[25] E. Akhmatskaya and S. Reich, “The targeted shadowing hybrid monte carlo (tshmec) method,” in New Algorithms for Macromolecular Simulation. Springer, 2006, pp. 141–153.

[26] T. Radivojević and E. Akhmatskaya, “Mix & match hamiltonian monte carlo,” arXiv preprint arXiv:1706.04032, 2017.

[27] C. Heide, F. Roosta, L. Hodgkinson, and D. Kroese, “Shadow manifold hamiltonian monte carlo,” in International Conference on Artificial Intelligence and Statistics. PMLR, 2021, pp. 1477–1485.

[28] A. Horowitz, “Stochastic quantization in phase space,” Physics Letters B, vol. 156, no. 1-2, pp. 89–92, 1985.

[29] M. Betancourt, “A conceptual introduction to hamiltonian monte carlo,” arXiv preprint arXiv:1701.02434, 2017.

[30] E. Hairer, M. Hochbruck, A. Iserles, and C. Lubich, “Geometric numerical integration,” Oberwolfach Reports, vol. 3, no. 1, pp. 805–882, 2006.

[31] A. M. Horowitz, “A generalized guided monte carlo algorithm,” Physics Letters B, vol. 268, no. 2, pp. 247–252, 1991.

[32] Y. Fang, J.-M. Sanz-Serna, and R. D. Skeel, “Compressible generalized hybrid monte carlo,” The Journal of chemical physics, vol. 140, no. 17, p. 174108, 2014.

[33] W. T. Mongwe and K. M. Malan, “The efficacy of financial ratios for fraud detection using self organising maps,” in 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2020, pp. 1100–1106.

[34] ——, “A survey of automated financial statement fraud detection with relevance to the south african context,” South African Computer Journal, vol. 32, no. 1, jul 2020.

[35] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” Mathematics of Control, Signals, and Systems (MCSS), vol. 2, no. 4, pp. 303–314, 1989.

[36] L. Mthembu, T. Marwala, M. I. Friswell, and S. Adhikari, “Model selection in finite element model updating using the bayesian evidence statistic,” Mechanical Systems and Signal Processing, vol. 25, no. 7, pp. 2399–2412, 2011.

[37] S. Marsland, Machine learning: an algorithmic perspective. CRC press, 2015.

[38] R. Mbuvha, “Bayesian neural networks for short term wind power forecasting,” 2017.

[39] D. J. MacKay, “Probable networks and plausible predictions—a review of practical bayesian methods for supervised neural networks,” Network: computation in neural systems, vol. 6, no. 3, pp. 469–505, 1995.

[40] D. Vats, J. M. Flegal, and G. L. Jones, “Multivariate output analysis for Markov chain Monte Carlo,” Biometrika, vol. 106, no. 2, pp. 321–337, 04 2019.

[41] L. Kish, Survey sampling, 1965, no. 04; HN29, K5.

[42] S. Cohen, R. Mbuvha, T. Marwala, and M. Deisenroth, “Healing products of gaussian process experts,” in International Conference on Machine Learning. PMLR, 2020, pp. 2068–2077.
WILSON TSAKANE MONGWE was born in Tembisa (Gauteng, South Africa). He holds a BSc in Computing from the University of South Africa, BBusSci in Actuarial Science from the University of Cape Town and a Masters in Mathematical Finance from the University of Cape Town. He is a recipient of the Google Ph.D. fellowship in machine learning which supports his current Ph.D. research at the University of Johannesburg. His research interests include Bayesian machine learning and Markov Chain Monte Carlo methods.

RENDANI MBUVHA was born in Venda (Limpopo, South Africa) is a lecturer in Statistics and Actuarial Science at the University of Witwatersrand. He is a qualified Actuary and a holder of the Chartered Enterprise Risk Actuary designation. He holds a BSc with Honours in Actuarial Science and Statistics from the University of Cape Town and an MSc in Machine Learning from KTH, Royal Institute of Technology in Sweden. He is a recipient of the Google Ph.D. fellowship in machine learning which supports his current Ph.D. research at the University of Johannesburg.

TSHILIDZI MARWALA (Senior Member, IEEE) was born in Duthuni, Venda, Limpopo, South Africa, in 28 July 1971. He received the bachelor’s degree in mechanical engineering from Case Western Reserve University, Cleveland, OH, USA, in 1995, the master’s degree in mechanical engineering from the University of Pretoria, Pretoria, South Africa, in 1997, and the Ph.D. degree in engineering from the University of Cambridge (St Johns College), Cambridge, U.K., in 2000. He is currently a registered Professional Engineer (Pr. Eng) with the Engineering Council of South Africa. He has also completed other leadership courses from Columbia Business School, National University of Singapore, GIBS University of Pretoria, Harvard Business School, and University of South Africa. Some of his accomplishments include being a fellow of Cambridge Commonwealth Trust (1997), CSIR (2005), South African Academy of Engineering (2007), TWAS, The World Academy of Science (2010), African Academy of Science (2013), South African Institute of Electrical Engineers (2016). He has won many awards including notably, the Bronze Order of Mapungubwe awarded by the President of the Republic of South Africa, in 2004, and the TWAS-AAS-Microsoft award for Young Scientists(2009). His contributions in artificial intelligence field come in forms of over 15 books he has authored, over 50 peer-reviewed chapters, over 50 journal publications and over 150 conference publications. He also holds the Deputy Chairman of the Presidential Fourth Industrial Revolution Commission of South Africa position and has contributed to over 50 articles to local press and journals on the subject of Fourth Industrial Revolution and Economics. He is the current Vice-Chancellor and Principal of the University of Johannesburg.

***