A Splitting Hamiltonian Monte Carlo Method for Efficient Sampling

Lei Li[1,2,3], Lin Liu[1,2,3,4] and Yuzhou Peng[1,*]

[1] *School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, P.R. China.*
[2] *Institute of Natural Sciences and MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, P.R. China.*
[3] *Shanghai Artificial Intelligence Laboratory, Shanghai, P.R. China.*
[4] *CMA-Shanghai and SJTU-Yale Joint Center for Biostatistics and Data Science, Shanghai Jiao Tong University, Shanghai 200240, P.R. China.*

Abstract. We propose a splitting Hamiltonian Monte Carlo (SHMC) algorithm, which can be computationally efficient when combined with the random mini-batch strategy. By splitting the potential energy into numerically nonstiff and stiff parts, one makes a proposal using the nonstiff part of U, followed by a Metropolis rejection step using the stiff part that is often easy to compute. The splitting allows efficient sampling from systems with singular potentials (or distributions with degenerate points) and/or with multiple potential barriers. In our SHMC algorithm, the proposal only based on the nonstiff part in the splitting is generated by the Hamiltonian dynamics, which can be potentially more efficient than the overdamped Langevin dynamics. We also use random batch strategies to reduce the computational cost to $\mathcal{O}(1)$ per time step in generating the proposals for problems arising from many-body systems and Bayesian inference, and prove that the errors of the Hamiltonian induced by the random batch approximation is $\mathcal{O}(\sqrt{\Delta t})$ in the strong and $\mathcal{O}(\Delta t)$ in the weak sense, where Δt is the time step. Numerical experiments are conducted to verify the theoretical results and the computational efficiency of the proposed algorithms in practice.

AMS subject classifications: 62M05, 65C05

Key words: Markov chain Monte Carlo, potential splitting, random-batch method, many body systems, Bayesian inference.

1 Introduction

Markov chain Monte Carlo (MCMC) [4, 22, 27, 47, 53, 58] methods are nowadays routinely used in a variety of scientific computing problems, including computing statistics

---

*Corresponding author. *Email addresses:* `leili2010@sjtu.edu.cn` (L. Li), `linliu@sjtu.edu.cn` (L. Liu), `pengyz@sjtu.edu.cn` (Y. Peng)

for many-body systems [1, 21], sampling from log-concave distributions [11, 13, 39, 43], parameter estimation in Bayesian statistics [2, 22, 25, 57] and Bayesian inverse problems [24, 50], just to name a few. Among MCMC methods, Hamiltonian Monte Carlo (HMC) [3, 7, 16, 49] has recently garnered a lot of attention in practice due to its scalability and efficiency in high-dimensional settings [6, 49]. Nonetheless, there are several situations where HMC can encounter difficulties. The first such scenario might be sampling from the Gibbs distribution

$$\mu(q) \propto \exp[-\beta U(q)] \tag{1.1}$$

of a many-body interacting particle system, where $\beta > 0$ is the (dimensionless) inverse temperature. It takes $\mathcal{O}(N^2)$ operations to compute the total potential energy

$$U(q) = \sum_{i=1}^{N} w_i V(q_i) + \sum_{i,j:i<j} w_i w_j \phi(q_i - q_j), \tag{1.2}$$

where $q_i \in \mathbb{R}^d$ is the position of the i-th particle and w_i denotes the weight. If one moves one particle per step, which is preferred in some applications [15, 21, 42], the computational cost of evolving the Hamiltonian dynamics and the Metropolis-Hastings correction step in HMC are both $\mathcal{O}(N)$. This fact makes HMC computationally expensive when sampling from the Gibbs distribution. Moreover, interaction potentials ϕ such as the Coulomb potential or the Lennard-Jones potential are usually singular [21]. Singularity in ϕ can introduce stiffness to the Hamiltonian system, which makes the numerical simulations difficult [62], and possibly leads to low acceptance rates [42], thus deteriorating the sampling efficiency of HMC.

As for another well-known example, let us consider Bayesian inference of a parameter θ based on its posterior distribution $p_{\text{post}}(\theta|\mathcal{D}_N)$ given the observed data $\mathcal{D}_N$

$$p_{\text{post}}(\theta|\mathcal{D}_N) \propto p_{\text{prior}}(\theta) \prod_{i=1}^{N} p(y_i; \theta), \tag{1.3}$$

where $\mathcal{D}_N = \{y_1, \ldots, y_N\}$ is a sample of size N drawn from the probabilistic model $p(\cdot; \theta)$. When $p(\cdot; \theta)$ is posited to be a mixture model, which is often the case in clustering and density estimation problems [20], the corresponding posterior distribution $p_{\text{post}}(\theta|\mathcal{D}_N)$ may be multimodal. One main reason for multimodality in mixture models is the non-identifiability of the parameters due to label switching [29]. When the target distribution (the posterior distribution in this case) is multimodal, most MCMC algorithms have difficulty moving between isolated modes and are therefore prone to generate biased samples. As a result, HMC could fail to explore the entire state space and lead to even worse performance than the simple Random Walk Metropolis (RWM) algorithm [44]. Indeed, the Hamiltonian simulation heavily relies on the gradient information of the potential U, easily resulting in samples trapped in one single well of U when U has multiple modes. A popular strategy of sampling from multimodal distributions is running multiple chains with over-dispersed initializations in parallel. Other methods include parallel

tempering [23,48], simulated tempering [45], Wormhole HMC [38] and the Wang-Landau algorithm [12,60,61].

In order to deal with the numerical stiffness in the potential for the Gibbs distributions of many-body particle systems and multi-modal distributions, like the singularity in Lennard Jones potential and the deep wells in the multi-modal distributions mentioned above, we adopt the idea of potential splitting [28,42,49,55,56] for this numerical challenge. Compared to the existing potential splitting for HMC [49,56], our motivation is different in the sense that we aim to separate out the numerically stiff part so that some standard methods can be used for generating the proposal samples. This motivation is similar to the Random Batch Monte Carlo (RBMC) method [42], but we choose to use the Hamiltonian dynamics to generate the samples for possible shorter mixing time. This leads us to a novel splitting Hamiltonian Monte Carlo (SHMC) method. In particular, consider a target distribution ρ with the effective potential energy

$$U = -\beta^{-1}\log\rho + const. \tag{1.4}$$

After splitting the potential U into two parts $U = U_1 + U_2$, a candidate state is first proposed by evolving the Hamiltonian system corresponding to U_1 and then is probabilistically accepted by a Metropolis rejection step determined by U_2. By combining with the random mini-batch strategy, the SHMC method could address the above issues regarding computational costs, singularity and multimodality to some extent. The benefits of our SHMC method can be summarized as follows:

(a) When sampling from the Gibbs distribution (1.1) of an interacting particle system, particularly with singular interaction, we decompose U into a nonstiff part U_1 and a part U_2 with short-range interactions. The nonstiffness of U_1 guarantees the feasibility of the Hamiltonian simulation while U_2 reduces the computational cost of the Metropolis correction step since only particles within a small neighborhood need to be considered when calculating the acceptance probability.

(b) There is often a big summation term in the potential energy (1.4), e.g. see Eq. (1.2). In this paper, we apply the random mini-batch idea [31,52] to further reduce the computation load due to summing over many terms, leading to more efficient algorithms. In particular, if only one randomly picked particle is updated [21] in each iteration, the cost can be reduced to $\mathcal{O}(1)$ per iteration.

(c) As for a multimodal distribution, we choose U_1 with a relatively flat landscape to enable the Markov chains traveling between isolated wells. Such U_1 can be constructed by either reducing the height of the barriers between different wells of U or filling the wells with "computational sand", an idea borrowed from the metadynamics approach [37] (for further details, see Remark 2.4).

(d) As we shall see later, our proposed SHMC method can possibly have shorter mixing time compared to the Langevin-based methods like the RBMC method.

We remark that the random batch strategy has recently been applied to interacting particle systems, resulting in several efficient algorithms [31, 34, 35, 41, 42]. Different from the mini-batch in stochastic gradient descent (SGD) [8,9,52] and stochastic gradient Hamiltonian Monte Carlo (SGHMC) [10], the random batch methods for interacting particle systems aim to grasp the dynamics as well. The rationale is a time averaging effect, by which RBM methods converge in a "law of large number" manner in time [30, 31]. Introducing random batch strategy into the Hamiltonian dynamics, however, will introduce extra effective noise, and the Hamiltonian is no longer conserved and long time simulation is not accurate. Nonetheless, in our proposed method, the Hamiltonian dynamics is never performed for a long time so this issue is nonessential in this paper; see Section 3.2 for the error estimates.

The rest of this paper is organized as follows. In Section 2, we propose our algorithm and provide some motivating applications. Specifically, we first elaborate the general framework of the splitting Monte Carlo method. Then we incorporate HMC into the general framework and present the splitting Hamiltonian Monte Carlo (SHMC) algorithm in Section 2.1. Finally, taking interacting particle systems and Bayesian inference problems as examples, we further introduce the random batch idea to reduce computational cost in Section 2.2. In Section 3, we prove an error bound of the value of the approximate Hamiltonian generated in random batches and show the advantage of our method over sampling algorithms based on overdamped Langevin equations. Finally, we perform some numerical experiments to verify the high numerical and computational efficiency of our algorithm in Section 4.

2   The Splitting Hamiltonian Monte Carlo method and its random batch variant

In this section, we propose the Splitting Hamiltonian Monte Carlo (SHMC) method to overcome the aforementioned difficulties that HMC could encounter in practice. To make SHMC scalable to large-scale problems such as many-body systems and Bayesian inference, we also incorporate the idea of random mini-batch into SHMC. We abbreviate the resulting algorithms RB-SHMC.

To start with, let us introduce the general framework of the splitting Monte Carlo method; see Algorithm 1. As mentioned in Section 1, common MCMC methods such as RWM and HMC behave poorly when sampling from the Gibbs distribution of interacting particle systems with singular potentials and sampling from multimodal distributions that often arise in Bayesian inference of mixture models. In both cases, there is some stiff part in the potential U, so the traditional gradient-based MCMC methods fail to sample from the target distribution efficiently. To alleviate this problem, we first construct a surrogate potential U_1 without stiffness and apply some generic sampling method $\mathcal{S}$ satisfying detailed balance with respect to the Gibbs distribution $\rho_1 \propto \exp(-\beta U_1)$. Then one obtains a proposed state which is a sample from ρ_1 and may be severely biased from

the target distribution $\rho \propto \exp(-\beta U)$. In the second step, a Metropolis rejection step using $U_2 := U - U_1$ is required to determine whether the aforementioned proposal shall be accepted as a new sample, thereby correcting the bias introduced in the first step.

The splitting Monte Carlo (Algorithm 1) can be viewed as a special case of the Metropolis-Hastings algorithm. Denote the transition probability of $\mathcal{S}$ from state x to state y as $q(y;x)$. The acceptance probability in the Metropolis rejection step is then given by

$$A(y;x) := \min\left(1, \frac{\exp[-\beta U(y)]q(x;y)}{\exp[-\beta U(x)]q(y;x)}\right), \tag{2.1}$$

and the detailed balance condition of $\mathcal{S}$ yields

$$\frac{q(x;y)}{q(y;x)} = \frac{\exp[-\beta U_1(x)]}{\exp[-\beta U_1(y)]}. \tag{2.2}$$

Substitute identity (2.2) into (2.1), one can see that the acceptance probability is only related to the U_2 component of U

$$\begin{aligned} A(y;x) &= \min\left(1, \frac{\exp[-\beta U(y)]\exp[-\beta U_1(x)]}{\exp[-\beta U(x)]\exp[-\beta U_1(y)]}\right) \\ &= \min\left(1, \exp\left[-\beta(U_2(y) - U_2(x))\right]\right). \end{aligned} \tag{2.3}$$

Algorithm 1 (General Splitting Monte Carlo algorithm)

Step 1. Propose a candidate state x^* using some generic sampling method $\mathcal{S}$ satisfying detailed balance with respect to the Gibbs distribution corresponding to U_1

$$\exp[-\beta U_1(x)]q(x;x^*) = \exp[-\beta U_1(x^*)]q(x^*;x). \tag{2.4}$$

Step 2. Set $x \leftarrow x^*$ with probability

$$A(x^*;x) := \min\left(1, \exp\left[-\beta(U_2(x^*) - U_2(x))\right]\right). \tag{2.5}$$

Otherwise, $x \leftarrow x$ remains unchanged.

2.1 Splitting Hamiltonian Monte Carlo method

We now present a splitting Monte Carlo method based on HMC, coined as the Splitting Hamiltonian Monte Carlo (SHMC) method, which evolves a Hamiltonian system to make a proposal in Step 1 of Algorithm 1.

We first recall some fundamental properties of the standard HMC, which obviously satisfies the detailed balance condition. HMC generates samples from the D-dimensional target distribution (1.1) by introducing an auxiliary variable $p \in \mathbb{R}^D$, namely the momentum, and sampling from the joint distribution $\tilde{\mu}(x,p) \propto \exp[-\beta H(x,p)]$. Traditionally, the

Hamiltonian $H(x,p) = U(x) + K(p)$ is separable with $K(p) = \|p\|^2/(2m)$, where m can be interpreted as "mass" in physics term. Hence, p is sampled from the isotropic multivariate Gaussian distribution $\mathcal{N}(0,(m/\beta)I_D)$. The Hamiltonian system satisfies many nice properties such as conservation of energy, reversibility, and symplecticity [3, 49], and therefore it is usually discretized by the leapfrog method, a symmetric symplectic integrator [40]. HMC proposes a new state in the phase space by simulating a Hamiltonian dynamics. The proposed momentum is flipped after the numerical simulation to guarantee the time reversibility and a correction step follows to correct the discretization error induced by the leapfrog integrator. In practice, the momentum flipping step is skipped since K is symmetric and the momentum should be resampled to ensure ergodicity. However, frequent resampling may result in random walk and thus multiple leapfrog steps are performed per iteration.

Sharing the same algorithmic structure as HMC but differing in some specific steps, SHMC numerically evolves a Hamiltonian system with a surrogate potential U_1 to propose a candidate state (x^*, p^*) and carries out a Metropolis rejection step only using $U_2 := U - U_1$ after the momentum flipping step. The same as HMC, flipping p^* ensures time reversibility but is usually omitted since the rejection step has nothing to do with the momentum and it should also be resampled in the next iteration to ensure ergodicity. Note that the candidate state (x^*, p^*) is proposed based on the surrogate potential U_1 and may not be a typical sample from the target distribution. Therefore, the rejection step for U_2 is critical to guarantee the empirical distribution of the samples to converge to the target distribution, beyond the sole purpose of correcting the discretization error as in HMC.

Remark 2.1. Recall that since the Hamiltonian system is numerically solved by the leapfrog integrator, which is symplectic and has a second-order accuracy, the discretization error would not be a great issue. Moreover, the method will have negligible systematic error if the step size Δt_n gradually decreases to zero and the Metropolis correction step in HMC to correct the errors introduced by the time discretization can be omitted to reduce computational cost. A related error bound can be found in Theorem 3.1 below. We also refer interested readers to the experiments in [63, Section 5] for further empirical evidence that decreasing step size helps to eliminate the systematic error.

For high-dimensional problems, especially the interacting particle systems, randomly picking a few entries of x to update per iteration can be more efficient [21]. In the following, we assume $x = [x_1, \ldots, x_N] \in \mathbb{R}^{d \times N}$ and only x_i, one randomly chosen entry of x, is updated per iteration. The detailed procedure of our SHMC method is given in Algorithm 2 and the splitting schemes are presented in Section 2.2.

Algorithm 2 (Splitting Hamiltonian Monte Carlo algorithm)

1: Split $U := U_1 + U_2$ such that U_1 is nonstiff and the remaining part U_2 contains the numerically stiff part of U and can be computed efficiently. Randomly generate the initial position $x^{(0)}$ and set N_s as the total number of samples.

2: **for** $n=1,\ldots,N_s$ **do**

3:     Randomly pick x_i with uniform probability. Sample a momentum
      $p_i \sim \mathcal{N}(0,(m/\beta)I_d)$ and set $L_n \geq 1, \Delta t_n > 0$.

4:     $(x_i^*, p_i^*) \leftarrow (x_i^{(n-1)}, p_i)$.

5:     **for** $\ell = 1,\ldots,L_n$ **do**

6:       Perform the following leapfrog discretization:

$$p_i^* \leftarrow p_i^* - \frac{\Delta t_n}{2}\nabla U_1(x_i^*),$$
$$x_i^* \leftarrow x_i^* + \Delta t_n \frac{p_i^*}{m}, \qquad\qquad (2.6)$$
$$p_i^* \leftarrow p_i^* - \frac{\Delta t_n}{2}\nabla U_1(x_i^*).$$

7:     **end for**

8:     Evaluate the following Metropolis acceptance probability:

$$A \leftarrow A(x_i^*;x_i) = \min\left(1, \exp\left[-\beta\left(U_2(x_i^*) - U_2(x_i^{(n-1)})\right)\right]\right).$$

9:     Generate a random number ζ from uniform distribution on $[0,1]$. If $\zeta \leq A$, set

$$x_i^{(n)} \leftarrow x_i^*.$$

    Otherwise, set

$$x_i^{(n)} \leftarrow x_i^{(n-1)}.$$

10: **end for**

2.2   Splitting Hamiltonian Monte Carlo method with random batch

In this section, we discuss how to apply the random batch idea [52] to SHMC for problems with big summation in the potential U. Typical examples include the Gibbs distribution arising from interacting particle systems and the posterior distribution from Bayesian inference problems. For these examples, any algorithm that needs to evaluate the full summation, such as the traditional Metropolis Hastings algorithm and the standard HMC, will be computationally inefficient.

The random batch idea is originated from the famous SGD algorithm [8, 52] and the stochastic gradient Langevin dynamics (SGLD) for Bayesian inference [63]. The random mini-batch based on random grouping strategy has recently been applied to many-body interacting systems [31–33,41,64]. Recently, a novel efficient molecular dynamics simulation method by building random batch and importance sampling into the Fourier space of Ewald sum has been proposed in [34]. In a nutshell, the random batch strategy approximates a big summation by summing over only a small random subset of size $\mathcal{O}(1)$ and therefore the complexity per time step is reduced to $\mathcal{O}(1)$ [31, 42]. The convergence of the random batch lies in the time average behavior and it can be intuitively interpreted as an extension of the "law of large number" from "over samples" to "over time".

Remark 2.2. (a) Introducing random batch strategy into the Hamiltonian dynamics will introduce extra noise and lead to the so-called "numerical heating" effects [30]. The Hamiltonian is no longer conserved and the long-run simulation will not be accurate. However, in either HMC or SHMC, the Hamiltonian dynamics are only performed for a short time so this issue is nonessential; see Section 3.2 for error bound estimates.

(b) The error in the Hamiltonian introduced by random batch, albeit small as proved in Section 3.2, will necessarily introduce systematic errors in the invariant measures of the Markov chain. One may correct this using a Metropolis rejection step as in the Metropolis-adjusted Langevin algorithm (MALA) [4, 54]. However, we choose not to do this simply because such a correction will bring back an $\mathcal{O}(N)$ computation cost that we have tried to avoid using random batch. As above in Remark 2.1, one may decrease Δt to make the systematic error negligible.

In the following, we present two implementations of the RB-SHMC algorithm.

2.2.1 Gibbs distribution of interacting particle system

For the Gibbs distribution of a N-particle interacting system, the potential (1.2) comprises the confining potential V and the interaction potential ϕ. Without loss of generality, we assume that the confining potential is smooth and the interaction potential is symmetric in the sense $\phi(q_i-q_j)=\phi(q_j-q_i)$. We also assume $w_i\equiv w$ for simplicity. In practice, the interaction between particles is of long range and usually singular. It is thus recommended that ϕ is decomposed into a smooth part ϕ_1 and a short-range part ϕ_2. Hence, one has

$$
\begin{aligned}
U_1(q) &= w\sum_{i=1}^{N}V(q_i)+w^2\sum_{i,j:i<j}\phi_1(q_i-q_j),\\
U_2(q) &= w^2\sum_{i,j:i<j}\phi_2(q_i-q_j),
\end{aligned}
\tag{2.7}
$$

where $q=[q_1,\dots,q_N]$ with $q_i\in\mathbb{R}^d$ denoting the position of the i-th particle and w represents the weight. Clearly, the short range nature of ϕ_2 is essential to reducing the computational cost because only particles within that short range need to be counted and therefore U_2 can be efficiently evaluated using suitable data structures like cell lists [21, Appendix F].

The Hamiltonian dynamics corresponding to U_1 defined in (2.7) are given by

$$
\begin{aligned}
\dot{q}_i &= \frac{p_i}{m},\\
\dot{p}_i &= -\left(\frac{\nabla V(q_i)}{w(N-1)}+\frac{1}{N-1}\sum_{j:j\neq i}\nabla\phi_1(q_i-q_j)\right).
\end{aligned}
\tag{2.8}
$$

Note that we have done a time rescaling $t\leftarrow w\sqrt{N-1}t$ and redefined the momentum by $p\leftarrow p/(w\sqrt{N-1})$ so that the interacting force on the i-th particle is $\mathcal{O}(1)$. The step of

resampling of p in Algorithm 2 now becomes

$$p_i \sim \mathcal{N}\left(0, \frac{m}{w^2(N-1)} I_d\right).$$

The acceptance rate is again given by $\min(1, \exp(-\Delta U_2))$.

Remark 2.3. Note that the time rescaling here is equivalent to setting

$$\tilde{U}_1 = \sum_{i=1}^{N} \frac{V(q_i)}{w(N-1)} + \frac{1}{N-1} \sum_{i,j:i<j} \phi_1(q_i - q_j),$$

$$\tilde{U}_2 = \frac{1}{N-1} \sum_{i,j:i<j} \phi_2(q_i - q_j),$$

and choosing $\beta = w^2(N-1)$ in Algorithm 2. The acceptance rate is then again based on $\beta \Delta \tilde{U}_2 = \Delta U_2$.

In the case of mean-field regime where $w \sim 1/N$, it becomes the usual ODE system for interacting particles in literature.

When random batch is applied, the total interacting force acting on i, which should be contributed by all the other particles $j \neq i$, is now approximated by that only from a random batch of particles and the Hamiltonian system is reduced to

$$\dot{\tilde{q}}_i = \frac{\tilde{p}_i}{m},$$
$$\dot{\tilde{p}}_i = -\left(\frac{\nabla V(\tilde{q}_i)}{w(N-1)} + \frac{1}{s} \sum_{j \in \xi} \nabla \phi_1(\tilde{q}_i - \tilde{q}_j)\right), \quad (2.9)$$

where $\xi = \{\xi_1, \dots, \xi_s\}$ is a random subset of $\{1, \dots, N\} \setminus \{i\}$, bookkeeping the indices of the particles selected into the small batch.

We remark that there exist other ways to implement the random batch strategy. For example, in the case of Coulomb interactions in a periodic box, the importance sampling in Fourier space is possible and the time scaling in (2.8) may be unnecessary [34]. Even if we do random batch in real space as indicated above in the molecular dynamics regime where $w = 1$, the time scaling to have $1/N$ factor in (2.7) is only a convenience made for establishing theoretical results. It will not change the intrinsic dynamics and is not necessary for implementation in practice [33].

2.2.2 Multimodal posterior in Bayesian inference

In Bayesian inference problems, the potential of the posterior is given by

$$U(\theta) = \frac{1}{\beta}\left[-\log p_{\text{prior}}(\theta) + \sum_{i=1}^{N} -\log p(y_i; \theta)\right], \quad (2.10)$$

where θ is the parameter of interest, $y_i, i=1,\dots,N$ are the observations and $\beta > 0$ is a scaling factor[†].

In many cases, p_{post} is multimodal and thus there are multiple wells in the landscape of U. The energy barriers between wells frustrate general MCMC methods and lead to biased samples trapped in one single well. In this situation, one can flatten the landscape of U to construct U_1 by adding "computational sand" G into the wells and thus $U_2 = -G$. One feasible choice of G is the Gaussian kernel.

Here, we take the potential U_1 to be

$$U_1(\theta) = -\log p_{\text{prior}}(\theta) + \sum_{i=1}^{N} -\log p(y_i; \theta) + G(\theta).$$

As above, we do a time rescaling $t \leftarrow \sqrt{N}t$ and the momentum is redefined as $p \leftarrow p/\sqrt{N}$. Then, we can resample the momentum from the isotropic Gaussian $\mathcal{N}(0, (m/N)I_d)$, and the momentum update in the RB-SHMC is given by

$$p^* \leftarrow p^* - \Delta t_\ell \left[-\frac{1}{N}\nabla \log p_{\text{prior}}(\theta^*) + \frac{1}{s}\sum_{i \in \xi_\ell} -\nabla \log p(y_i; \theta^*) + \frac{1}{N}\nabla G(\theta^*) \right], \qquad (2.11)$$

where $\xi_\ell = \{\xi_{\ell,1},\dots,\xi_{\ell,s}\} \subset \{1,\dots,N\}$ denotes the random batch of data chosen in the ℓ-th step of the current iteration to approximate the likelihood of the entire dataset. The acceptance rate is simply $\min(1, \exp(\Delta G))$.

The time rescaling here is again equivalent to choosing $\beta = N$

$$U_1 = -\frac{1}{N}\log p_{\text{prior}}(\theta) + \frac{1}{N}\sum_{i=1}^{N} -\log p(y_i; \theta) + \frac{1}{N}G, \quad U_2 = -\frac{1}{N}G$$

in Algorithm 2. The goal of time rescaling or choosing $\beta = N$ is to scale the summation term by $1/N$ so that the variance of

$$\mathcal{H}i^\ell := \frac{1}{s}\sum_{i \in \xi_\ell} -\nabla \log p(y_i; \theta^*) - \frac{1}{N}\sum_{i=1}^{N} -\nabla \log p(y_i; \theta^*)$$

is controlled independent of N. Indeed, choosing $\beta = N$ is only for convenience in the analysis and incurs no essential change in the physical interpretation [33]. In practice, one may perform the updates directly by

$$p^* \leftarrow p^* - \Delta t_\ell \left[-\nabla \log p_{\text{prior}}(\theta^*) + \frac{N}{s}\sum_{i \in \xi_\ell} -\nabla \log p(y_i; \theta^*) + \nabla G(\theta^*) \right].$$

[†]Note that the N here denotes the number of observed data and has nothing to do with the dimension of the variable θ.

Remark 2.4. The idea of adding computational sand is inspired by metadynamics [37], which is informally described as "filling the free energy wells with computational sand". In some special cases, the modes of the posterior, and thus the locations of the wells in U, can be roughly estimated by the modes of the marginal distributions. Then an amenable G can be designed in advance. Such an example is presented in Section 4.3.2 below. Generally, however, the modes of the marginals fail to imply the location of the modes of the joint distribution. Hence, one may dynamically add a sequence of standard Gaussian kernels at the positions already visited by the proposals of the Hamiltonian simulation. In this case, Metropolis rejection steps are skipped in the first few trial runs. Then one collects all the added Gaussian kernels and obtains U_1, which is U plus the sum of these Gaussian kernels. Interested readers may refer to [37] for more details. We leave the theoretical justification of this proposal to future work.

3   Theoretical properties of SHMC and RB-SHMC

In this section, we provide some theoretical justifications for the SHMC and RB-SHMC algorithms. First, in Section 3.1, we give some informal justification on why SHMC could potentially be better than splitting Monte Carlo methods based on overdamped Langevin dynamics. Second, in Section 3.2, we provide error estimates of the values of the approximate Hamiltonian generated from RB-SHMC under some mild regularity conditions. These results give practitioners some guidance on when to expect SHMC or RB-SHMC to succeed in applications.

3.1   Benefits of Hamiltonian dynamics in sampling compared to the overdamped Langevin equations

As is well known, the Langevin sampling [5, 13, 51, 54] is another popular sampling scheme for the Gibbs distribution. Consider the overdamped Langevin equation

$$dX = -\nabla U(X)\,dt + \sqrt{\frac{2}{\beta}}\,dW, \tag{3.1}$$

where $\beta > 0$ is a known constant and W denotes the d-dimensional Wiener process. It converges exponentially to its invariant measure, the Gibbs distribution (1.1), under mild regularity conditions [46]. The Langevin sampler discretizes the overdamped Langevin equation (3.1), by the Euler-Maruyama scheme

$$X(t^{n+1}) = X(t^n) - \Delta t_n \nabla U\big(X(t^n)\big) + \sqrt{\frac{2\Delta t_n}{\beta}} z_n, \quad z_n \sim \mathcal{N}(0, I_d).$$

Hence, the Markov chain generated by the Langevin sampler also exhibits an exponentially fast convergence rate up to the discretization error of order $\mathcal{O}(\Delta t)$ [14].

In this section, we point out several possible benefits of the HMC-based samplers over the overdamped Langevin equation-based samplers like the RBMC and SGLD algorithms. In fact, the RBMC method uses the overdamped Langevin equation for generating the proposals in Step 1 of Algorithm 1, as the overdamped Langevin equation satisfies the detailed balance condition with respect to this invariant measure determined by U_1. This may provide some heuristic justification for why in certain cases RB-SHMC may be preferred compared to RBMC, such as the examples described in Sections 4.1 and 4.2.

In the discussion here, we assume that the potential U is smooth and for the sake of simplicity, there is no splitting so $U_1 \equiv U$. Recall that we perform L_n leapfrog steps with time step size Δt_n in the n-th iteration, which means that we evolve a Hamiltonian system with respect to $(X_{i^{(n)}}, P_{i^{(n)}})$ for $t \in [T_{i^{(n)}}^{(n-1)}, T_{i^{(n)}}^{(n)})$. Assume the particle index updated in the n-th iteration is $i \equiv i^{(n)}$, where we omit the superscript for convenience. Then we approximate the momentum $P_i(t')$ at time $t' \in (T_i^{(n-1)}, T_i^{(n)})$ by

$$P_i(t') = P_i(T_i^{(n-1)}) + \int_{T_i^{(n-1)}}^{t'} -\nabla U(X_i(\tau)) d\tau.$$

Thus

$$
\begin{aligned}
X_i(T_i^{(n)}) &= X_i(T_i^{(n-1)}) + \int_{T_i^{(n-1)}}^{T_i^{(n)}} m^{-1} P_i(t') dt' \\
&= X_i(T_i^{(n-1)}) + \frac{L_n \Delta t_n}{m} P_i(T_i^{(n-1)}) - \frac{1}{m} \int_{T_i^{(n-1)}}^{T_i^{(n)}} \int_{T_i^{(n-1)}}^{t'} \nabla U(X_i(\tau)) d\tau dt' \\
&= X_i(T_i^{(n-1)}) - \widehat{\Delta t_n} \widetilde{\nabla U} + \sqrt{\frac{2\widehat{\Delta t_n}}{\beta}} z_n
\end{aligned}
\tag{3.2}
$$

with

$$\widetilde{\nabla U} = \frac{2}{(L\Delta t_n)^2} \int_{T_i^{(n-1)}}^{T_i^{(n)}} \int_{T_i^{(n-1)}}^{t'} \nabla U(X_i(\tau)) d\tau dt'$$

and $\widehat{\Delta t_n} = (L_n \Delta t_n)^2/2m$ and $z_n \sim \mathcal{N}(0, I_d)$.

Roughly, the numerical evolution in the n-th iteration of SHMC is nearly equivalent to the simulation of an overdamped Langevin equation for time $\widehat{\Delta t_n}$. Hence, the effective time step of the overdamped Langevin equation is longer than the evolution time $L_n \Delta t_n$ of the corresponding Hamiltonian system, provided $L_n \Delta t_n > 2m$. This means that running the Hamiltonian system with $L_n \Delta t_n$ large has a longer effective dynamics so it might approximate the equilibrium faster than the overdamped Langevin dynamics. This property may be the reason why SHMC is preferred in some applications over RBMC (see Examples 4.1-4.2). Moreover, if the Hamiltonian step in (3.2) is accurately solved, it then exactly satisfies the detailed balance with respect to the target distribution. Discretizing the Hamiltonian using leapfrog might yield $\mathcal{O}(\Delta t^2)$ error, which is better compared to

the Euler-Maruyama discretization of overdamped Langevin dynamics with error $\mathcal{O}(\Delta t)$ if we use the same step size.

Another benefit of HMC is that it can take advantage of the symplectic integrators to conserve the Hamiltonian even when the system evolves for a long time. Usually, with fixed time step Δt, the discretization error would increase exponentially with the evolution time. However, the symplectic integration can be carried out for $e^{\mathcal{O}(1/\Delta t)}$ time steps such that the numerical trajectories would always oscillate around the exact Hamiltonian trajectories within this time interval [4, 26]. This guarantees that the Hamiltonian can be conserved within a reasonable tolerance for a long time, which is highly desired for HMC-based algorithms as the invariant measure is related to the Hamiltonian directly. Moreover, due to the time-reversibility and volume-preservation property of the leapfrog integrator, HMC exhibits better scalability compared to RWM and MALA in some cases. For instance, when the potential of the target distribution is a sum of i.i.d. terms such as (1.2) and satisfies mild regularity conditions, HMC can allow a larger time step $\Delta t = \mathcal{O}(N^{-1/4})$, and thus requires less steps to traverse the state space, as the number N of i.i.d. terms tends to infinity [6, 49].

3.2 Error estimates of the random batch method

In this section, we derive an error bound for the values of the Hamiltonian approximated by the random batch strategy in one iteration of RB-SHMC (the process between any two consecutive resampling steps of the momentum). We analyze how the random batch strategy affects the value of the Hamiltonian corresponding to U_1. Note that the error of the value of the Hamiltonian determines the deviation of the invariant measure so this estimate can provide us some insight on the impact of the random batch approximation on the invariant measures.

Without loss of generality, we denote the beginning of an iteration as $t = 0$, and consider the following Hamiltonian system:

$$
\begin{aligned}
\dot{x} &= \frac{p}{m}, \\
\dot{p} &= -\nabla V_1(x) + \frac{1}{N_J} \sum_{j \in \mathcal{J}} \nabla \psi_j(x)
\end{aligned}
\tag{3.3}
$$

and its counterpart after applying random batch

$$
\begin{aligned}
\dot{\tilde{x}} &= \frac{\tilde{p}}{m}, \\
\dot{\tilde{p}} &= -\nabla V_1(\tilde{x}) + \frac{1}{s} \sum_{j \in \xi} \nabla \psi_j(\tilde{x}).
\end{aligned}
\tag{3.4}
$$

Here N_J is the size of the index set $\mathcal{J}$ and ξ denotes a random subset of size s. We remark that introducing the notations V_1 and ψ_j enables us to unify the Hamiltonian

systems of the two examples discussed in Section 2.2. In particular, in the examples of interacting particle systems (Sections 4.1, 4.2 and 4.4), $V_1(\boldsymbol{x}) = V(\boldsymbol{x})/(w(N-1))$ is a scaling of the confining potential of the current particle i which is located at $\boldsymbol{x}$ and $\psi_j(\boldsymbol{x}) = \psi(\boldsymbol{x};q_j) = -\phi_1(\boldsymbol{x}-q_j)$ is the interaction potential between particle i and another particle $j \neq i$ located at q_j. When it comes to the example of the Bayesian posterior inference (Section 4.3.2), $V_1(\cdot) = (-\log p_{\mathrm{prior}}(\cdot) + G)/N$ is a scaling of the sum of the log prior density and the auxiliary Gaussian kernel added to flatten the landscape of the potential and ψ_j corresponds to the log-likelihood term $-\log p(y_j;\cdot)$.

We first clarify the notations used in the following analysis. Fix the number of leapfrog steps and the time step in each iteration to be L and Δt and denote $T := L\Delta t$. The initial values (i.e., the values at the beginning of one iteration) are denoted by X^0 and P^0. Denote the random batch selected in the ℓ-th leapfrog step as ξ_ℓ, $\ell = 1,\dots,L$. The Kolmogorov extension theorem [17] guarantees the existence of a probability space $(\Omega,\mathcal{F},\mathbb{P})$ such that $\{X^0,P^0,\xi_\ell, \ell = 1,\dots,L\}$ are independent random variables on this space. The $L^2(\Omega,\mathbb{P})$ norm is denoted as $\|\cdot\| = \sqrt{\mathbb{E}|\cdot|^2}$, where $\mathbb{E}$ is the expectation with respect to the probability measure $\mathbb{P}$.

Now we start to estimate the difference of $H(\tilde{\boldsymbol{x}}(T),\tilde{\boldsymbol{p}}(T))$ and $H(\boldsymbol{x}(T),\boldsymbol{p}(T))$, where $(\boldsymbol{x}(\cdot),\boldsymbol{p}(\cdot))$ is the solution of the fully coupled Hamiltonian system (3.3) and therefore $H(\boldsymbol{x}(T),\boldsymbol{p}(T)) = H(\boldsymbol{x}(0),\boldsymbol{p}(0))$.

For error bound estimates, we assume that the Hamiltonian is sufficiently smooth.

Assumption 3.1. $V_1 \in C^1$ and $\psi \in C^2$ with $\nabla V_1, \nabla \psi$ and $\nabla^2 \psi$ being bounded.

Remark 3.1. In cases where the actual confining potential has an unbounded gradient, one can still split V into a bounded and an unbounded part so the unbounded part can be used in the Metropolis rejection step. However, this may not be needed in practice as the particles usually move in a bounded domain so the unboundedness of V is not very relevant.

The main results are as follows.

Theorem 3.1. *Under Assumption 3.1 and assume Δt is sufficiently small such that $T(1+T^2)\Delta t \ll 1$. If $\boldsymbol{x}(0) = \tilde{\boldsymbol{x}}(0) = X^0$, and $\boldsymbol{p}(0) = \tilde{\boldsymbol{p}}(0) = P^0$, and the fourth moment of the initial momentum $\mathbb{E}[|P^0|^4]$ is bounded, then the error from using random batches can be bounded above as follows:*

$$\|H(\tilde{\boldsymbol{x}}(T),\tilde{\boldsymbol{p}}(T)) - H(\boldsymbol{x}(T),\boldsymbol{p}(T))\| \leq C\sqrt{T(1+T^2)\Delta t}, \tag{3.5}$$

and for any test function $\varphi \in C_b^\infty$,

$$\left| \mathbb{E}[\varphi(H(\tilde{\boldsymbol{x}}(T),\tilde{\boldsymbol{p}}(T))) - \varphi(H(\boldsymbol{x}(T),\boldsymbol{p}(T)))] \right| \leq C_\varphi T(1+T^2)\Delta t, \tag{3.6}$$

where $C,C_\varphi > 0$ are constants independent of T and N and C_φ depends on φ.

Remark 3.2. As mentioned in Remark 2.2, the estimation of the error introduced by random batch guarantees that the systematic error of our method vanishes as $\Delta t_n \to 0$.

We also remark that the bounded fourth moment conditions on the momentum P^0 are trivially satisfied as it usually sampled from a multivariate Gaussian distribution. In the proof, we need to define the filtration $\{\mathcal{F}_\ell\}_{1\leq\ell\leq L}$ by

$$\mathcal{F}_\ell := \sigma\left(X^0, P^0, \{\xi_k\}_{1\leq k\leq\ell}\right), \tag{3.7}$$

the σ-algebra generated by the initial values X^0, P^0 and the first ℓ random batches ξ_k, $k=1,\dots,\ell$. Obviously, $\mathcal{F}_{\ell+1}=\sigma(\mathcal{F}_\ell\cup\sigma(\xi_{\ell+1}))$. Denote the error of the random approximation of the summation term by

$$\mathcal{H}i^\ell(t) := \frac{1}{s}\sum_{j\in\xi_\ell}\nabla\psi_j(\tilde{\boldsymbol{x}}(t)) - \frac{1}{N_J}\sum_{j\in\mathcal{J}}\nabla\psi_j(\tilde{\boldsymbol{x}}(t)). \tag{3.8}$$

Clearly, $\mathcal{H}i^\ell$ is $\mathcal{F}_\ell$-measurable and for any $\ell\geq 0, q>0$,

$$\sup_{t\in[t_\ell, t_{\ell+1})} \mathbb{E}\left[|\mathcal{H}i^\ell(t)|^q \,\big|\, \mathcal{F}_\ell\right] \leq 2^q\|\nabla\psi\|_\infty^q. \tag{3.9}$$

With these, the following upper bound, which will be used in the proof of Theorem 3.1, can be easily obtained.

Lemma 3.1. *Under conditions in Theorem* 3.1*, it holds that,*

$$\sup_{0\leq t\leq T} \mathbb{E}\left[|\tilde{\boldsymbol{p}}(t)|^4\right] \leq C(1+T^4), \tag{3.10}$$

where $C>0$ is a constant independent of $N, s, \xi_k, k=1,\dots,L$.

The proof of this simple lemma is deferred to Appendix A. We are now able to provide the proof of the error estimates for the Hamiltonian.

Proof of Theorem 3.1. Define

$$\Delta H(t) := H\left(\tilde{\boldsymbol{x}}(t), \tilde{\boldsymbol{p}}(t)\right) - H\left(\boldsymbol{x}(t), \boldsymbol{p}(t)\right) = H\left(\tilde{\boldsymbol{x}}(t), \tilde{\boldsymbol{p}}(t)\right) - H^0, \tag{3.11}$$

where $H^0 = H(X^0, P^0)$ is the initial value of the Hamiltonian since H is conserved under the evolution (3.3). Consequently, it is straightforward to find

$$\frac{d}{dt}\Delta H(t) = \frac{\tilde{\boldsymbol{p}}(t)}{m}\cdot\mathcal{H}i^{L_t}(t), \tag{3.12}$$

where

$$L_t = \left\lfloor \frac{t}{\Delta t} \right\rfloor. \tag{3.13}$$

Hence,

$$\frac{d}{dt}\|H(\tilde{\boldsymbol{x}}(t), \tilde{\boldsymbol{p}}(t)) - H(\boldsymbol{x}(t), \boldsymbol{p}(t))\|^2 = 2\mathbb{E}\left[\Delta H(t)\frac{\tilde{\boldsymbol{p}}(t)}{m}\cdot\mathcal{H}i^{L_t}(t)\right]$$

$$
\begin{aligned}
&= 2\mathbb{E}\left[\left(\Delta H(t_{L_t}) + \frac{\tilde{\boldsymbol{p}}(\hat{t})}{m}\cdot\mathcal{H}i^{L_t}(\hat{t})(t-t_{L_t})\right)\frac{\tilde{\boldsymbol{p}}(t)}{m}\cdot\mathcal{H}i^{L_t}(t)\right] \\
&=: I_1 + I_2,
\end{aligned}
\tag{3.14}
$$

where $\hat{t}\in(t_{L_t}, t)$ by the mean value theorem for one variable functions.

Step 1. Estimation of I_1: Note that $\Delta H(t_{L_t})\in\mathcal{F}_{L_t-1}$ and $\tilde{\boldsymbol{p}}(t_{L_t})\in\mathcal{F}_{L_t-1}$. Then one has

$$
\mathbb{E}\left[\Delta H(t_{L_t})\frac{\tilde{\boldsymbol{p}}(t_{L_t})}{m}\mathcal{H}i^{L_t}(t_{L_t})\right] = \mathbb{E}\left[\Delta H(t_{L_t})\frac{\tilde{\boldsymbol{p}}(t_{L_t})}{m}\cdot\mathbb{E}\left(\mathcal{H}i^{L_t}(t_{L_t})\Big|\mathcal{F}_{L_t-1}\right)\right] = 0.
$$

Therefore,

$$
\begin{aligned}
&\mathbb{E}\left[\Delta H(t_{L_t})\frac{\tilde{\boldsymbol{p}}(t)}{m}\mathcal{H}i^{L_t}(t)\right] \\
&= \mathbb{E}\left[\Delta H(t_{L_t})\left(\frac{\tilde{\boldsymbol{p}}(t)}{m}\mathcal{H}i^{L_t}(t) - \frac{\tilde{\boldsymbol{p}}(t_{L_t})}{m}\mathcal{H}i^{L_t}(t_{L_t})\right)\right] \\
&= \mathbb{E}\left[\Delta H(t_{L_t})\mathbb{E}\left(\frac{\tilde{\boldsymbol{p}}(t)}{m}\mathcal{H}i^{L_t}(t) - \frac{\tilde{\boldsymbol{p}}(t_{L_t})}{m}\mathcal{H}i^{L_t}(t_{L_t})\Big|\mathcal{F}_{L_t}\right)\right] \\
&= \mathbb{E}\left[\Delta H(t_{L_t})\int_{t_{L_t}}^t\mathbb{E}\left(\frac{\dot{\tilde{\boldsymbol{p}}}(t')}{m}\mathcal{H}i^{L_t}(t') + \frac{\tilde{\boldsymbol{p}}(t')}{m}\dot{\mathcal{H}i}^{L_t}(t')\Big|\mathcal{F}_{L_t}\right)dt'\right] \\
&\leq C(1+T^2)\Delta t\|\Delta H(t_{L_t})\| \\
&\leq C(1+T^2)\Delta t\left(\|\Delta H(t)\| + \left\|\frac{\tilde{\boldsymbol{p}}(\hat{t})}{m}\mathcal{H}i^{L_t}(\hat{t})(t-t_{L_t})\right\|\right) \\
&\leq C(1+T^2)\Delta t\left(\|\Delta H(t)\| + C(1+T)\Delta t\right) \\
&\leq C(1+T^2)\Delta t\|\Delta H(t)\|.
\end{aligned}
$$

Step 2. Estimation of I_2:

$$
\begin{aligned}
&\mathbb{E}\left[\frac{\tilde{\boldsymbol{p}}(\hat{t})}{m}\mathcal{H}i^{L_t}(\hat{t})\frac{\tilde{\boldsymbol{p}}(t)}{m}\mathcal{H}i^{L_t}(t)\right] \\
&\leq \left\{\mathbb{E}\left[\left|\frac{\tilde{\boldsymbol{p}}(\hat{t})}{m}\right|^4\right]\mathbb{E}\left[\left|\mathcal{H}i^{L_t}(\hat{t})\right|^4\right]\mathbb{E}\left[\left|\frac{\tilde{\boldsymbol{p}}(t)}{m}\right|^4\right]\mathbb{E}\left[\left|\mathcal{H}i^{L_t}(t)\right|^4\right]\right\}^{\frac{1}{4}} \leq C(1+T^2).
\end{aligned}
$$

Therefore, I_2 is controlled by $C(1+T^2)\Delta t$.

Combining the above two steps gives

$$
\frac{d}{dt}\|\Delta H(t)\|^2 \leq C(1+T^2)\Delta t\left(\|\Delta H(t)\| + 1\right).
\tag{3.15}
$$

Recall that Δt is assumed to be sufficiently small. Then one has

$$
\|H(\tilde{\boldsymbol{x}}(T), \tilde{\boldsymbol{p}}(T)) - H(\boldsymbol{x}(T), \boldsymbol{p}(T))\| \leq C\sqrt{T(1+T^2)\Delta t}.
\tag{3.16}
$$

Similarly, (3.6) follows from

$$
\begin{aligned}
&\left| \mathbb{E}\left[\varphi\left(H(\tilde{\boldsymbol{x}}(T),\tilde{\boldsymbol{p}}(T))\right) - \varphi\left(H(\boldsymbol{x}(T),\boldsymbol{p}(T))\right) \right] \right| \\
&= \left| \int_0^T \mathbb{E}\left[\varphi'\left(H(\tilde{\boldsymbol{x}}(t),\tilde{\boldsymbol{p}}(t))\right) \frac{\tilde{\boldsymbol{p}}(t)}{m}\mathcal{H}i^{L_t}(t) \right] dt \right| \\
&= \left| \int_0^T \mathbb{E}\left[\varphi'\left(H(\tilde{\boldsymbol{x}}(t),\tilde{\boldsymbol{p}}(t))\right) \frac{\tilde{\boldsymbol{p}}(t)}{m}\mathcal{H}i^{L_t}(t) - \varphi'\left(H(\tilde{\boldsymbol{x}}(t_{L_t}),\tilde{\boldsymbol{p}}(t_{L_t}))\right) \frac{\tilde{\boldsymbol{p}}(t_{L_t})}{m}\mathcal{H}i^{L_t}(t_{L_t}) \right] dt \right| \\
&\leq \left| \int_0^T \int_{t_{L_t}}^t \mathbb{E}\left[\varphi''\left(H(\tilde{\boldsymbol{x}}(t'),\tilde{\boldsymbol{p}}(t'))\right) \left(\frac{\tilde{\boldsymbol{p}}(t')}{m}\mathcal{H}i^{L_t}(t') \right)^2 \right] dt' dt \right| \\
&\quad + \left| \int_0^T \int_{t_{L_t}}^t \mathbb{E}\left[\varphi'\left(H(\tilde{\boldsymbol{x}}(t'),\tilde{\boldsymbol{p}}(t'))\right) \left(\frac{\dot{\tilde{\boldsymbol{p}}}(t')}{m}\mathcal{H}i^{L_t}(t') + \frac{\tilde{\boldsymbol{p}}(t')}{m}\dot{\mathcal{H}}i^{L_t}(t') \right) \right] dt' dt \right| \\
&\leq \left(\|\varphi'\|_\infty + \|\varphi''\|_\infty \right) CT(1+T^2)\Delta t.
\end{aligned}
$$

The proof is complete. □

4   Numerical examples

In this section, we conduct some numerical experiments to demonstrate the computational efficiency of SHMC and RB-SHMC. First, we give an artificial example with regular interactions to test the influence of the length of Hamiltonian dynamics in an iteration. In the second example, we simulate the Dyson Brownian motion for $N \gg 1$. The advantage of potential splitting and the high efficiency of random batch will manifest itself in this example, where the interaction is singular and of long range. Third, we consider a distribution with double well potential to demonstrate the benefit of potential splitting even in regular interaction cases, and then consider a classical example in Bayesian inference: estimating the locations of a two-dimensional Gaussian mixture model using SHMC and RB-SHMC. Finally, we consider the Gibbs distribution of a three-dimensional interacting particle system, the interaction potential of which is modelled by the Lennard-Jones potential. This example, which is of higher dimension than previous examples, further demonstrates the benefit of RB-SHMC in practice.

4.1   A test example

In the following toy example, we explore how L_n, the number of leapfrog steps in the n-th iteration, can influence the efficiency of the RB-SHMC algorithm. For the purpose of a relatively fair comparison between algorithms with different evolution time per iteration, we define the evolution time T_E up to the M-th iteration of Algorithm 2 by

$$
T_E(M) = \frac{1}{N}\sum_{n=1}^{M} L_n \times \Delta t_n. \tag{4.1}
$$

The rescaling factor $1/N$ arises from the fact that Algorithm 2 updates only one "particle" per iteration and thus on average N iterations are needed to update all the variables.

Consider the simple one-dimensional Langevin equation

$$dX^i = P^i dt,$$

$$dP^i = \left(-\alpha X^i + \frac{1}{N-1} \sum_{j:j\neq i} \frac{X^i - X^j}{1 + |X^i - X^j|^2} - \gamma P^i \right) dt + \sqrt{\frac{2\gamma}{\beta}} dW^i. \tag{4.2}$$

Clearly, the interaction is smooth and bounded and its derivative is also bounded. The corresponding Gibbs distribution is

$$\mu \propto \exp\left[-\frac{\beta}{2} \left(\alpha \sum_i X_i^2 - \frac{1}{N-1} \sum_{i,j:i<j} \ln\left(1 + |X_i - X_j|^2\right) + \sum_i P_i^2 \right) \right]. \tag{4.3}$$

Consider a system of $N = 500$ particles sampled from the uniform distribution on $[-10, 10]$. Since the interaction kernel is regular, we move all the particles simultaneously using $\phi_1 := \phi$ and omit the Metropolis rejection step since $\phi_2 = 0$. Choosing $\alpha = 1, \beta = 1$ and fixing the batch size $s = 1$, the ℓ-th leapfrog step includes the following updates:

$$p \leftarrow p - \frac{\Delta t_n}{2} \left(\alpha x - \frac{x - \eta_\ell}{1 + |x - \eta_\ell|^2} \right),$$

$$x \leftarrow x + \Delta t_n p,$$

$$p \leftarrow p - \frac{\Delta t_n}{2} \left(\alpha x - \frac{x - \eta_\ell}{1 + |x - \eta_\ell|^2} \right),$$

where the i-th element of η_ℓ is $\eta_{\ell,i} = x_{\xi_i}$, with $\xi_i \in \{1,2,\ldots,N\} \setminus \{i\}$ for $i = 1,2,\ldots,N$. Fixing the time step $\Delta t_n \equiv 0.02$, we run RB-SHMC with $L_n \equiv 100$ and $L_n \equiv 10$ respectively for $T_E = 100$ units of evolution time to see how the efficiency can be influenced by different values of L_n. Fig. 1 shows the empirical densities at "evolution times" $T_E = 30,60,100$ obtained by the simple "bin-counting". Specifically, the empirical density in the j-th bin is approximated by $\bar{\mu}_j \approx N_j / (N_{tot} h)$, where N_j is the number of particles in the j-th bin during the entire sampling process and N_{tot} is the number of total particles (clearly, $N_{tot} = N_s N$ where N_s is the number of iterations in sampling). One can observe that RB-SHMC with larger L_n approaches the equilibrium faster, meaning that it can have shorter burn-in phase. This observation is consistent with the heuristic explanation provided in Section 3.1.

To quantify the effects of L_n in different stages of sampling, we use the following quantity (which we call "relative error") to gauge the error of the empirical density:

$$\tilde{U} := \sum_j \left| \frac{N_j}{N_{tot}} - \frac{\tilde{N}_j}{\tilde{N}_{tot}} \right|,$$

where j denotes the bin index, and the quantities with tildes are the reference quantities obtained by HMC (25 leapfrog steps with time step size $\Delta t = 0.01$ per sampling iterations, for a total of $4e8$ iterations).
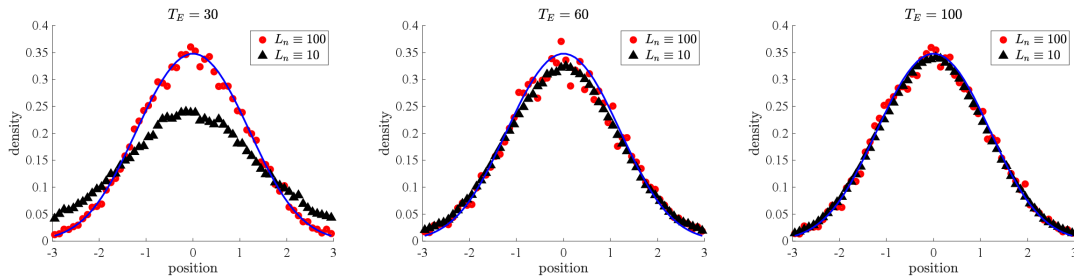
Figure 1: Empirical densities obtained by RB-SHMC with different L_n, numbers of leapfrog steps, at three different evolution time points $T_E = 30, 60, 100$. The red dots are for $L_n = 100$ leapfrog steps and the black triangles are for $L_n = 10$ leapfrog steps in each iteration. The blue curve is the reference density obtained by running HMC, 25 leapfrog steps with time step size $\Delta t = 0.01$ per iteration, for $4e8$ iterations. The empirical density obtained by RB-SHMC with large L_n (red dots), i.e. more leapfrog steps in each iteration, approximates the reference density better than the empirical density obtained by RB-SHMC with small L_n (black triangles).

As can be seen from Fig. 2, the experiment with larger L_n is more efficient when the evolution time is not very long as it has shorter burn-in phase. However, the systematic error becomes larger when one runs the sampling for longer time. Hence, an adaptive strategy can be used: one uses a large L_n in the early sampling phase, and when the distribution is close to equilibrium one can then switch to smaller L_n so that the results can be more accurate. Here, for comparison, we set $L_n = 100$ for $T_E \leq 100$ and then set $L_n = 10$. The error vs. the evolution time and CPU time are plotted in Fig. 2. One can see that running RB-SHMC with a large L_n for a few units of evolution time at the beginning accelerates the empirical density to approach its equilibrium state while reducing L_n to a smaller number after reaching a quasi-equilibrium yields a smaller error. The adaptive method can thus be efficient.
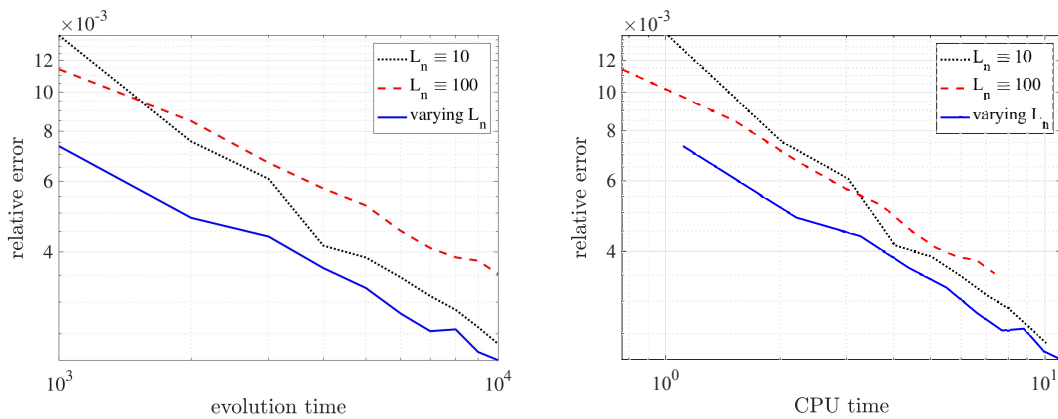

Figure 2: Left panel: relative error versus evolution time. Right panel: relative error versus CPU time. In both panels, blue curve is for RB-SHMC with varying L_n, red dashed curve is for RB-SHMC with $L_n \equiv 100$ while black dotted curve is for RB-SHMC with $L_n \equiv 10$. The relative error of the empirical density obtained by RB-SHMC with varying L_n (blue curve) decays faster than those of RB-SHMC with fixed L_n.

4.2 Dyson Brownian motion

In this section, we demonstrate the efficiency of RB-SHMC by simulating the Dyson Brownian motion with singular interaction kernels, which models the eigenvalues of certain random matrices [18, 19]

$$d\lambda_i(t) = \left(-\lambda_i(t) + \frac{1}{N-1} \sum_{j:j\neq i} \frac{1}{\lambda_i(t) - \lambda_j(t)} \right) dt + \sqrt{\frac{2}{N-1}} dW_i(t), \tag{4.4}$$

where $\{\lambda_i(t)\}$ denote the eigenvalues and $\{W_i(t)\}$ are independent standard Brownian motions. We have replaced N in the original model with $N-1$, which is nonessential when $N \gg 1$. It has been shown in [59] that in the $N \to \infty$ limit, these eigenvalues follow a distribution ρ that satisfies

$$\partial_t \rho(x,t) + \partial_x \big(\rho(u-x) \big) = 0, \quad u(x,t) = \pi(H\rho)(x,t) = \text{p.v.} \int_{\mathbb{R}} \frac{\rho(y,t)}{x-y} dy, \tag{4.5}$$

where $H(\cdot)$ is the Hilbert transform on $\mathbb{R}$ and p.v. is the standard notation for integrals evaluated using the Cauchy principal value. The mean field equation (4.5) has the following invariant measure:

$$\rho(x) = \frac{1}{\pi} \sqrt{2-x^2}, \tag{4.6}$$

which is the celebrated Wigner semicircle law. Clearly, (4.4) has an invariant measure

$$\mu \propto \exp\left[-\left(\frac{N-1}{2} \sum_i x_i^2 - \sum_{i,j:i<j} \ln|x_i - x_j| \right) \right],$$

where the interaction $\phi(x_i - x_j) = -\ln(|x_i - x_j|)$ is singular. For samples $(\lambda_1, \cdots, \lambda_N) \sim \mu$, we expect that the random empirical measure $\mu_N = (\sum_i \delta(x - \lambda_i))/N$ will be close to ρ in the weak topology. Below, we collect N_s such configurations $(\lambda_1, \cdots, \lambda_N)$ by sampling from μ using our sampling methods, and then compare the empirical measure of these $N_s N$ samples to the target Wigner semicircle law ρ. Following [42], we use the surrogate potential $\phi_1(x_i - x_j) = \ln(100) - 100|x_i - x_j| + 1$ when $0 < |x_i - x_j| < 0.01$ to remove the singularity while performing RB-SHMC sampling.

Recall that the example in Section 4.1 indicates that running L_n leapfrog steps with L_n large in the burn-in phase can accelerate the convergence to a quasi equilibrium state while a small L_n in later iterations can reduce the error. Hence, we adopt a dynamic leapfrog steps in the simulation for this example. Specifically, we choose $L_n = 100$ in the first 10^5 sampling iterations and $L_n = 20$ in the next 3×10^5 iterations. Then we reduce L_n to 10 and keep it fixed until the end of the sampling process. The corresponding time step size Δt_n also varies in different sampling phases (detailed in Table 1). We compare it to the RBMC which is a splitting Monte Carlo based on overdamped Langevin dynamics. With L_n denoting the number of discretization steps in the n-th iteration, we consider

Table 1: The specific choices of the number of discretization steps L_n and the stepsize Δt_n in each iteration of RB-SHMC, RBMC-fixed and RBMC-varying in the experiment for Dyson Brownian motion.

		L_n	Δt_n
	$n \leq 10^5$	100	2×10^{-4}
RB-SHMC	$10^5 < n \leq 4 \times 10^5$	20	2×10^{-4}
	$n > 4 \times 10^5$	10	10^{-4}
RBMC-fixed		10	10^{-4}
	$n \leq 2 \times 10^5$	100	
RBMC-varying	$2 \times 10^5 < n \leq 8 \times 10^5$	20	10^{-4}
	$n > 8 \times 10^5$	10	

both fixed Δt_n and L_n parameters (denoted as "RBMC-fixed") and varying L_n (denoted as "RBMC-varying"). Table 1 shows the specific choice of these parameters for the three methods considered in this example.

The same evolution time (4.1) is used again for this example. Fig. 3 presents the empirical densities obtained by running RB-SHMC (red dots) for 10^6 and 10^7 sampling iterations, corresponding to 7.6 and 25.6 units of evolution time respectively. The numerical results generated by running RBMC (black triangles) for the same evolution time are also presented for comparison. Obviously, RB-SHMC approaches the equilibrium much faster than RBMC which is consistent with our heuristic justification in Section 3.1.

In Fig. 4, we visualize the relative error of the three methods vs. the evolution time and CPU time respectively. The same "relative error" as in Example 4.1 has been used. Clearly, RB-SHMC is far more efficient than RBMC and is also superior to RBMC-v2.
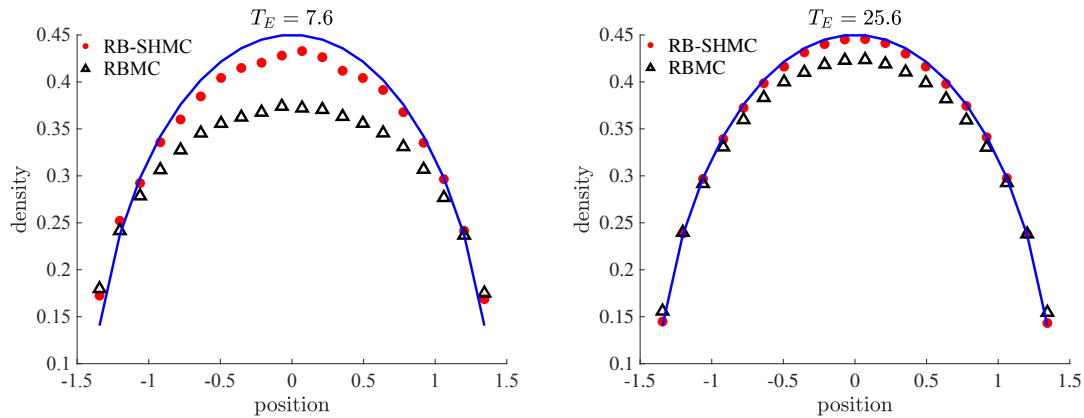

Figure 3: The empirical densities obtained by RB-SHMC and RBMC at different evolution times. Specifically, the left panel is for evolution time $T_E = 7.6$, or equivalently 10^6 iterations, and the right panel is for $T_E = 25.6$, or equivalently 10^7 iterations. The red dots and the black triangles are the empirical densities obtained by RB-SHMC and RBMC-fixed respectively. The blue curve is the equilibrium semicircle law (4.6). The empirical density obtained by RB-SHMC approaches faster than that of RBMC-fixed.
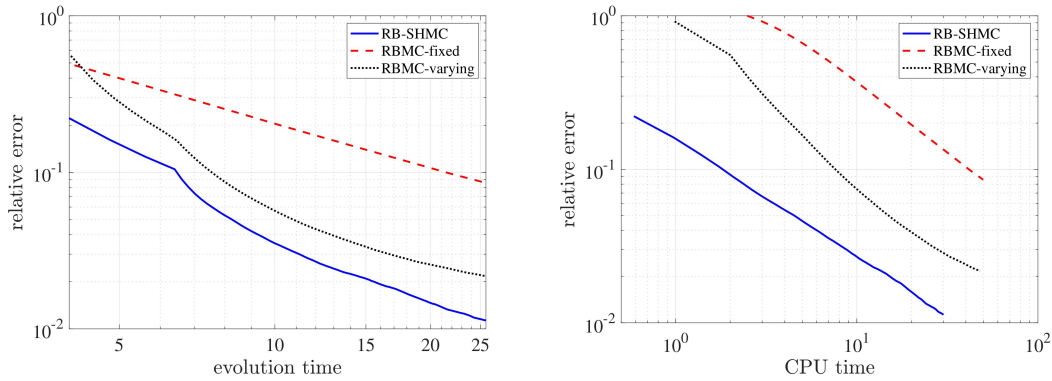
Figure 4: Left panel: relative error versus evolution time. Right panel: relative error versus CPU time. In both panels, the blue curve is for RB-SHMC, the red dashed curve is for RBMC-fixed while the black dotted curve is for RBMC-varying. The relative error of the empirical density obtained by RB-SHMC (blue curve) decays faster than those of the other two versions of RBMC considered in this example.

4.3   Bimodal distribution

The two examples given below will show that an appropriate potential splitting effectively prevents the problem of generating biased samples when the target distribution is multimodal. These examples showcase the advantage of SHMC and RB-SHMC even when there is no singularity in the potential, further expanding the application territory of the framework of SHMC and RB-SHMC where they can be superior to many other existing methods.

Intuitively, if we can "flatten" the landscape of U, then the samples can escape from one local minimum, cross the barrier and visit other local minima more efficiently. Without loss of generality, we only consider the bimodal distributions for simplicity.

4.3.1   Double well potential

Consider the following one-dimensional double well potential:

$$U(x) = \frac{H}{W^4}(x^2 - W^2)^2, \tag{4.7}$$

where $H = 20/\beta$ is the height of the barrier and $W = 1$ is the half width between the two wells. Clearly, the Gibbs distribution corresponding to (4.7) is

$$\mu(x) = \frac{1}{Z}\exp\left[-\beta\frac{H}{W^4}(x^2 - W^2)^2\right], \tag{4.8}$$

where Z is the normalizing constant.

We first apply the basic HMC algorithm to sample (4.8). There are $L = 40$ leapfrog steps in each iteration and the time step is $\Delta t = 2W/L = 0.05$. The initial position of the samples is generated randomly on $[-W, W]$. Drawing 10^5 samples from (4.8), the results plotted in Fig. 5 clearly show that samples are trapped in the local minimum $x = -W$.
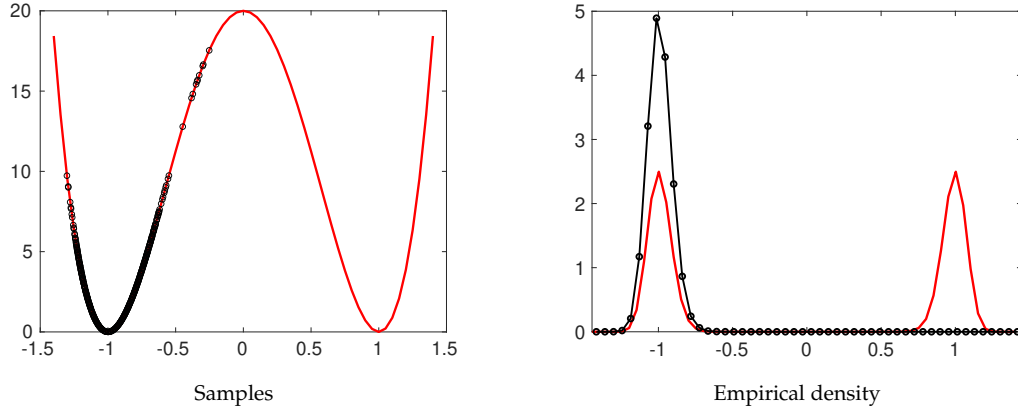
Figure 5: Samples and the empirical density obtained by HMC with $L=40$ leapfrog steps and time step size $\Delta t=2W/L=0.05$ in each iteration. In the left panel, the black dots are the samples obtained by HMC while the red curve is the exact potential (4.7). In the right panel, the black dots are the empirical density obtained by HMC while the red curve is the analytic density (4.8).

To fix the above problem, we decompose U into the sum of U_1, a double well potential energy with lower barrier to flatten the landscape and U_2, the difference between U and U_1. Specifically, we choose

$$U_1(x)=\begin{cases}\lambda U(x), & |x|<W, \\ U(x), & |x|\geq W,\end{cases} \tag{4.9}$$

and

$$U_2(x)=\begin{cases}(1-\lambda)U(x), & |x|<W, \\ 0, & |x|\geq W,\end{cases} \tag{4.10}$$

where $\lambda\in(0,1)$ is a positive constant being small enough. Here, we choose $\lambda=0.05$. We apply SHMC to draw 10^5 samples from (4.8). We again set $L=40$ leapfrog steps with time step $\Delta t=2W/L$ in each iteration. The samples we obtained and the resulting empirical density are presented in Fig. 6. Clearly, the splitting strategy can successfully overcome the local barrier.

4.3.2 Gaussian mixture model

As can be seen from the previous example, an appropriate splitting easily alleviates the problem of generating biased samples in the vanilla HMC algorithm when we have some knowledge about the landscape, e.g. where the local minima are located. However, in many problems including problems in Bayesian inference, the locations of the local minima are generally unknown. In this section, we consider such examples and propose a potential solution by modifying SHMC and RB-SHMC.
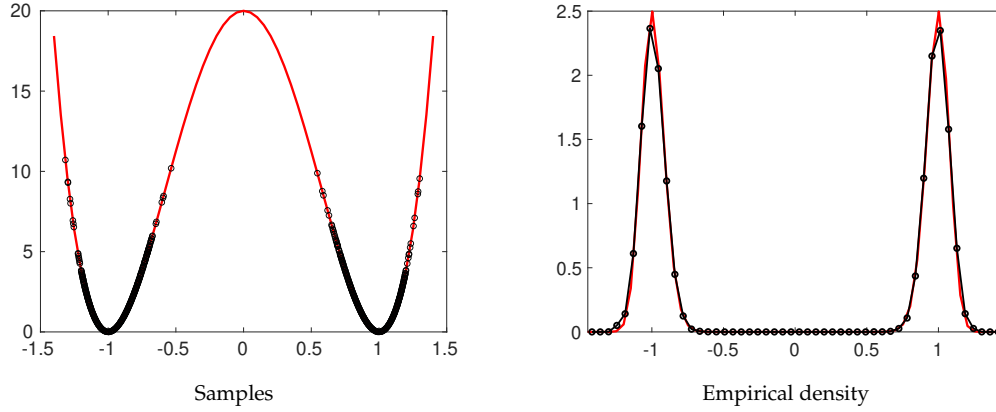
Figure 6: Samples and the empirical density obtained by SHMC using the surrogate potential U_1 given by (4.9) with $\lambda = 0.05$. There are $L = 40$ leapfrog steps with time step size $\Delta t = 2W/L = 0.05$ in each iteration. In the left panel, the black dots are the samples obtained by SHMC while the red curve is the exact potential (4.7). In the right panel, the black dots are the empirical density obtained by SHMC while the red curve is the analytic density (4.8).

To further illustrate our proposed solution, we slightly modify the first example in [63, Section 5.1], which is a classical Bayesian inference problem of mixture models. We will evaluate the performance of sampling from the posterior distribution using SHMC and RB-SHMC. To be specific, we consider the following Gaussian mixture model:

$$\theta_1 \sim \mathcal{N}(0,\sigma_1^2), \quad \theta_2 \sim \mathcal{N}(0,\sigma_2^2),$$
$$y_i \sim \frac{1}{2}\mathcal{N}(\theta_1,\sigma_y^2) + \frac{1}{2}\mathcal{N}(\theta_1+\theta_2,\sigma_y^2), \tag{4.11}$$

where $\sigma_1^2 = 10, \sigma_2^2 = 1$ and $\sigma_y^2 = 0.5$. $N = 100$ data points are drawn from the model with $(\theta_1,\theta_2) = (0,2)$. The left panel of Fig. 7 shows the potential U corresponding to the target posterior distribution, from which one can imagine that the barrier between the two wells of U will prevent samples from moving between these two wells. Indeed, the samples generated by HMC are trapped in one well; see the left panel of Fig. 8.

Inspired by the metadynamics approach [37], we add scaled Gaussian kernels in the wells to raise the altitudes of the landscape and obtain a new potential energy U_1. Defining $U_2 := U - U_1$, we get a decomposition $U = U_1 + U_2$. The locations of the wells are approximated by $\mathrm{mode}^{mar}(\theta_1)$ and $\mathrm{mode}^{mar}(\theta_2)$, the modes[‡] of the marginal distributions of θ_1 and θ_2 (see the right panel of Fig. 7), while h_b, the height of the barrier in U, is estimated by the difference between the altitude at the mid-point between the two wells and the average altitudes between the two wells. From our experiments, the scaled Gaussian kernel with height $h_G = h_b + 10/\beta$ and covariance matrix I_2 turns out to be a good choice

---

[‡]One may obtain the modes of the marginal distributions using any off-the-shelf mode-finding methods: e.g. the function 'findpeaks()' in Matlab.

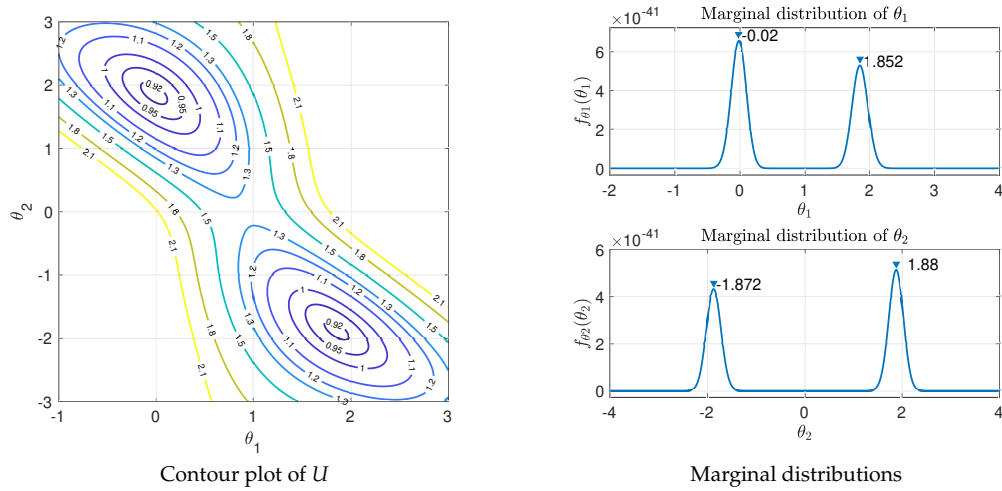Contour plot of U Marginal distributions

Figure 7: Left panel: The contour plot of the potential corresponding to the posterior of model (4.11). Right panel: The marginal distributions of the target posterior distribution.

for this specific example. In particular,

$$U_1(\theta) = U(\theta) + 2\pi h_G \left[\mathcal{N}\left(\text{mode}^{mar}(\theta_1), I_2\right) + \mathcal{N}\left(\text{mode}^{mar}(\theta_2), I_2\right) \right]. \qquad (4.12)$$

We choose $\beta = N$ such that the scaling factor of the big summation in U is $1/N$

$$U(\theta) = \frac{1}{N}\left(\frac{\theta_1^2}{2\sigma_1^2} + \frac{\theta_2^2}{2\sigma_2^2}\right) - \frac{1}{N}\sum_{i=1}^{N}\log\left[\exp\left(-\frac{(\theta_1 - y_i)^2}{2\sigma_x^2}\right) + \exp\left(-\frac{(\theta_1 + \theta_2 - y_i)^2}{2\sigma_x^2}\right)\right]. \qquad (4.13)$$

Under the specific setting of our experiment, the distance between the two wells in U is $d_w = 4.1931$ and the height of the barrier is $h_b = 0.4054$. The energy barriers for two wells are reduced to 0.0131 and 0.0113 respectively after the above two scaled Gaussian

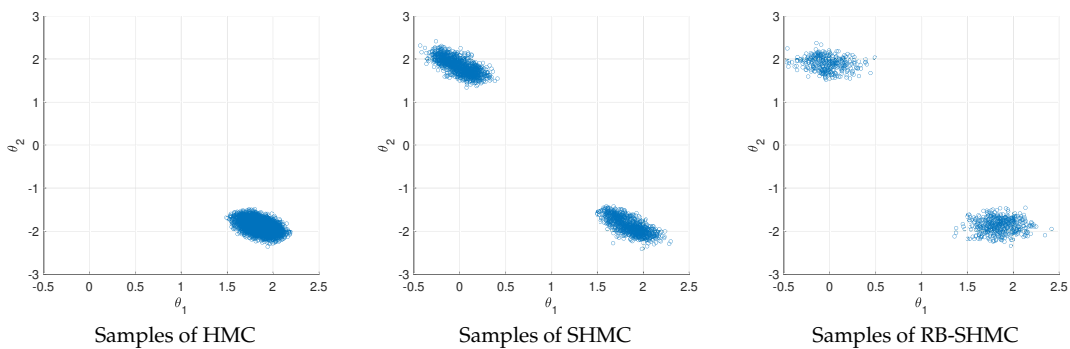Samples of HMC   Samples of SHMC   Samples of RB-SHMC

Figure 8: Samples from the posterior distribution obtained by HMC, SHMC and RB-SHMC with batch size $s = 10$. For all the 3 methods, the time step is fixed to be $\Delta t = 0.01$ and evolution time per iteration is $0.4d_w$.

kernels are added to the wells. Fix the evolution time of each iteration $L\Delta t = 0.4d_w$ and time step size $\Delta t = 0.01, 0.001$, we apply HMC, SHMC and RB-SHMC with batch size $s = 10$ to sample from the posterior. For each method, we collect 10^4 samples after 10^3 burn-in iterations. The scatter plot of the samples obtained by these three methods are shown in Fig. 8. Clearly, SHMC and RB-SHMC can sample from the two modes of the posterior without multiple initializations while the samples of HMC are trapped in one single well. For HMC, we even run the algorithm with longer evolution time per iteration $L\Delta t = 2d_w$. But still, the samples generated by HMC fail to escape from a single well. We have also tried to use SGLD [63], the naive version of SGHMC [10] and split HMC with "the splitting of data" strategy in [56]; however, all of them yield severely biased samples trapped in one single well since both the distance between the two wells and the height of the barrier between the two wells are large (larger than those in the similar experiment conducted in [63]). In summary, one can see that a suitable splitting strategy is powerful in the problem of sampling from a bimodal distribution.

Table 2 shows the average CPU time and acceptance rate of five runs of SHMC and RB-SHMC, with the corresponding standard deviations recorded in the parentheses. The CPU time spent in the sampling phase of RB-SHMC is less than 40% of that of SHMC and t_g, the CPU time to evaluate the summation term in ∇U_1, of RB-SHMC exhibits a significant advantage over that of SHMC, which clearly shows the benefit of random batch in terms of computational efficiency. In the experiment with $\Delta t = 0.01$, the acceptance rate of RB-SHMC is comparatively low since the difference between the exact dynamics and the dynamics using random batch has a relatively large variance when Δt is not small enough [42]. Nevertheless, the acceptance rate of RB-SHMC is comparable to that of SHMC when the time step is sufficiently small ($\Delta t = 0.001$).

Finally, we remark that the application of random batch does inject some noise into the original dynamics. As a result, the invariant measure cannot be preserved exactly. In particular, the random batch approximation may not be accurate enough for some complicated problems due to this extra variance by randomization. Hence, we suggest that practitioners exploit the random batch strategy when computational cost is high whereas the demand on accuracy is relatively low. A theoretical understanding of such computational-accuracy trade-off even for general random batch methods will be an interesting research direction to explore in future works.

Table 2: The CPU time of evaluating the summation term in ∇U_1, denoted by t_g, the CPU time of the sampling phase and the acceptance rate of SHMC and RB-SHMC. The evolution time per iteration is fixed to be $L\Delta t = 0.4d_w$ and $L = 168$ and $L = 1678$ leapfrog steps are perform respectively for the two choices of Δt.

		t_g (s)	Sampling time (s)	Acceptance rate
$\Delta t = 0.01$	SHMC	28.68 (0.4597)	36.00 (0.5848)	0.2411 (0.0070)
	RB-SHMC	6.62 (0.1006)	14.31 (0.1730)	0.0856 (0.0018)
$\Delta t = 0.001$	SHMC	289.56 (5.3467)	361.72 (6.8459)	0.2386 (0.0045)
	RB-SHMC	65.00 (1.0334)	139.62 (2.7512)	0.1903 (0.0010)

4.4 Lennard-Jones fluids

Now we apply RB-SHMC to sample from the Gibbs distribution of a three-dimensional interacting particle system whose interaction potential is modelled by the Lennard-Jones potential

$$\phi(r_i - r_j) = \Phi(r_{ij}) = 4\left[\left(\frac{1}{r_{ij}}\right)^{12} - \left(\frac{1}{r_{ij}}\right)^{6}\right], \tag{4.14}$$

where $r_{ij} = |r_i - r_j|$ is the distance between the two particles located at r_i and r_j in $\mathbb{R}^3$. Conventionally, we use periodic boxes with side length L to approximate the fluids in the numerical simulation. Therefore, each particle i interacts with not only all the other particles $j \neq i$ in the system but also their periodic images (including its own periodic image). The corresponding Gibbs distribution is given by

$$\mu \propto \exp\left[-\frac{\beta}{2}\sum_n{}'\sum_{i,j}\phi(r_i - r_j + nL)\right], \tag{4.15}$$

where $\beta = 1/T$ is the inverse temperature (all quantities have been scaled according to some basic units so that they are dimensionless) and $n \in \mathbb{Z}^3$. Here, we introduce the notation $\sum_n{}'$ to mean that the term with $n=0$ is excluded from the outer summation if $i=j$. Note that the Lennard-Jones potential (4.14) decays rapidly as the distance between the two particles increases so that one may approximate the discrete sum by the continuous integral over $r_i, r_j : |r_i - r_j + nL| \geq R_c$ for some chosen cutoff $R_c > 0$. Thus the pressure of the system can be approximated by

$$P = \frac{\rho}{\beta} + \frac{8}{V}\sum_{i=1}^{N}\sum_{\substack{j:j>i, \\ r_{ij}^* < R_c}}\left[2\left(\frac{1}{r_{ij}^*}\right)^{12} - \left(\frac{1}{r_{ij}^*}\right)^{6}\right] + \frac{16}{3}\pi\rho^2\left[\frac{2}{3}\left(\frac{1}{R_c}\right)^{9} - \left(\frac{1}{R_c}\right)^{3}\right], \tag{4.16}$$

where $\rho = N/L^3$ is the particle density and $r_{ij}^* = |r_i - r_j + n_{ij}^*L|$ is the distance between particle i and the nearest image of j. One may refer to [21,42] for more details. Moreover, with this cutoff, we only need to consider the interactions between particles and images that have distance less than R_c during our sampling algorithms using RB-SHMC and RBMC.

In this example, we take the cutoff length $R_c = L/2$ and compare our proposed RB-SHMC algorithm with RBMC [42]. We use similar settings to those in [42]. Specifically, the system consists of $N=500$ particles and the length of the periodic boxes is varied with the given density $\rho \in [0,1]$ by

$$L = \left(\frac{N}{\rho}\right)^{\frac{1}{3}}. \tag{4.17}$$

We adopt a simpler splitting strategy than that in [42], given by

$$\Phi(r) = \Phi_1(r) + \Phi_2(r), \tag{4.18}$$

with

$$\Phi_1(r) = \begin{cases} -2^{-\frac{1}{6}}r, & 0 < r < 2^{\frac{1}{6}}, \\ 4\left(r^{-12} - r^{-6}\right), & r \geq 2^{\frac{1}{6}}, \end{cases} \tag{4.19}$$

$$\Phi_2(r) = \begin{cases} 4\left(r^{-12} - r^{-6}\right) + 2^{-\frac{1}{6}}r, & 0 < r < 2^{\frac{1}{6}}, \\ 0, & r \geq 2^{\frac{1}{6}}. \end{cases} \tag{4.20}$$

In the experiment, we let the number of leapfrog steps and the timestep for RB-SHMC gradually decrease. Specifically, the number of leapfrog steps is chosen to be

$$L_n = \max\left\{ \left\lceil \frac{\log 2 \cdot L_1}{\log(1+k_n)} \right\rceil, L_1 - k_n + 1, 3 \right\}$$

with $k_n = \lceil n/2000 \rceil$ and the initial value $L_1 = 20$. By this strategy, L_n decreases as $21 - n/2000$ in the earlier stage of iterations (roughly when $n \leq 3.2 \times 10^4$) and as $20\log 2 / \log(1 + n/2000)$ in the later stage (roughly when $n \geq 3.2 \times 10^4$). Moreover, we make sure $L_n \geq 3$ so that the displacement is not too small. The timestep is chosen to be $\Delta t_n = \Delta t_1 n^{-\gamma}$ with the initial step size $\Delta t_1 = 0.2$ and the decaying exponent $\gamma = 0.06$. Following [42], the number of discretization steps and timestep size of RBMC are fixed at $L_n \equiv 9$ and $\Delta t_n \equiv 0.01$ in each iteration. For both RB-SHMC and RBMC, the batch size is chosen to be $s = 1$ and the first 2×10^5 iterations are taken to be the burn-in phase. Only the samples generated after the burn-in phase are collected. The pressure obtained over M iterations is computed in the following way: we first compute the pressure (4.16) of each sample configuration collected after the burn-in phase, and then we take the sample average of the pressures over these M iterations. It can be seen from Fig. 9 that the pressures obtained by 10^7 iterations of both RB-SHMC (red dots) and RBMC (black triangles) capture the reference solution (blue curve) given by the fitting curve provided in [36] well.

Following [42], the numerical results are evaluated by the relative error in the ℓ^2 norm

$$\sqrt{\sum_{i=1}^{9} \frac{1}{9}\left(P_i - P(\rho_i)\right)^2} \bigg/ \sqrt{\sum_{i=1}^{9} \frac{1}{9}P(\rho_i)^2}, \tag{4.21}$$

where P_i denotes the pressure at different density values $\rho = \rho_i$ (here we choose 9 different density values as shown in Fig. 9) and $P(\cdot)$ represents the reference solution given by the fitting curves in [36]. Fig. 10 presents how the relative errors of RB-SHMC and RBMC vary over the CPU time used. Here, the value of the pressure P_i used to compute the relative error (4.21) is averaged over the pressures generated by 10 independent chains to further reduce the impact of randomness. One can clearly see that both the relative errors of RB-SHMC and RBMC quickly decrease to below 1% and the relative error of RB-SHMC decays even faster than that of RBMC.
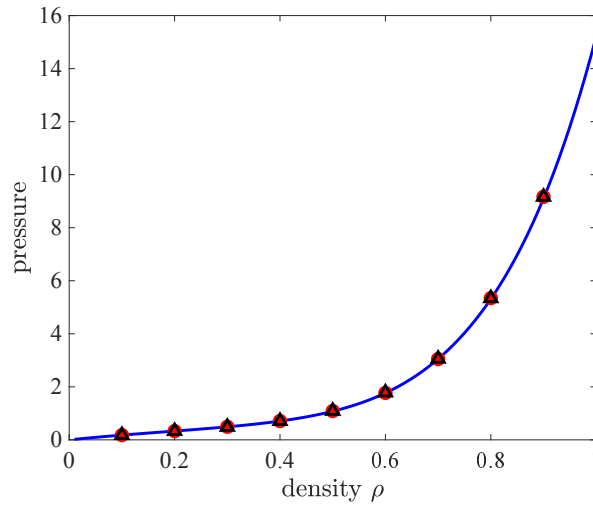
Figure 9: The pressure obtained by 10^7 iterations of RB-SHMC (red dots) and RBMC (black triangles) respectively. The samples are collected after the burn-in phase, defined as the first 2×10^5 iterations of the two methods. The blue curve is the reference solution given by the fitting curve in [36].
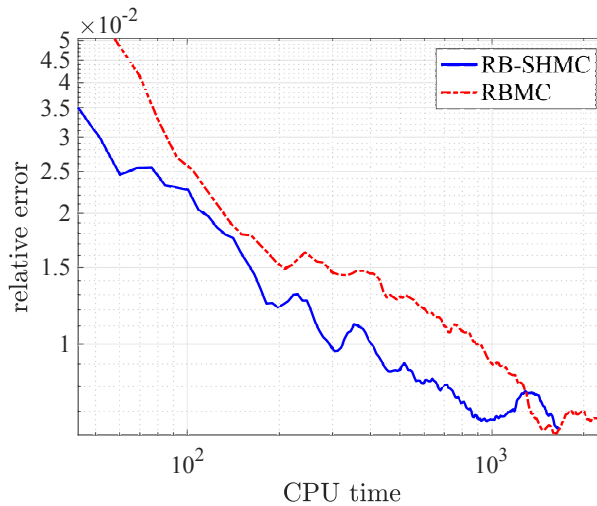

Figure 10: The relative ℓ^2 error of the pressures obtained by RB-SHMC (blue curve) decays faster than that of RBMC (red dashed curve). The value of the relative error (4.21) is computed by the pressure averaged over 10 independent chains.

Remark 4.1. The three examples arising from interacting particle systems (Example 4.1, Example 4.2 and Example 4.4) clearly exhibit the benefits of RB-SHMC. First, for an interacting particle system with N particles, the application of the random batch strategy efficiently reduces the computational cost per timestep from $\mathcal{O}(N)$ to $\mathcal{O}(1)$. A second benefit of RB-SHMC comes from the splitting strategy. Note that the interaction potentials

in the examples of Dyson Brownian motion and Lennard-Jones fluids are singular, popular gradient-based sampling methods, such as SGLD and standard HMC, are doomed to fail. Moreover, a clever splitting scheme can further reduce the computational cost. Recall that we perform the splitting by truncating the pairwise interaction potential at a small cutoff radius $r = r_c$ so that the number of particles within the range of r_c to the current particle is $\mathcal{O}(1)$ and thus the computational cost of the Metropolis rejection step is reduced from $\mathcal{O}(N)$ to $\mathcal{O}(1)$.

Appendix A. Proof of Lemma 3.1

Proof. Denote C_q as a constant depending on q and C_q may change from line to line. For any $t \in [0,T]$,

$$
\begin{aligned}
\frac{d}{dt}\mathbb{E}\big(|\tilde{\boldsymbol{p}}(t)|^q \,\big|\, \mathcal{F}_L\big) &= q\mathbb{E}\left(|\tilde{\boldsymbol{p}}(t)|^{q-1} \left| -\nabla V_1(\tilde{\boldsymbol{x}}(t)) + \frac{1}{s}\sum_{j\in\tilde{\xi}_{L_t}} \nabla\psi_j(\tilde{\boldsymbol{x}}) \right| \,\bigg|\, \mathcal{F}_L \right) \\
&\le q\big(\|\nabla V_1\|_\infty + \|\nabla\psi\|_\infty\big)\mathbb{E}\big(|\tilde{\boldsymbol{p}}(t)|^{q-1}\,\big|\,\mathcal{F}_L\big) \\
&\le C_q\mathbb{E}\big(|\tilde{\boldsymbol{p}}(t)|^q\,\big|\,\mathcal{F}_L\big)^{\frac{q-1}{q}},
\end{aligned}
\tag{A.1}
$$

where the last line follows from Hölder's inequality. By integration, one has

$$
\mathbb{E}\big(|\tilde{\boldsymbol{p}}(t)|^q\,\big|\,\mathcal{F}_L\big) \le C_q\big(|\tilde{\boldsymbol{p}}(0)|^q + t^q\big) \le C_q\big(|\tilde{\boldsymbol{p}}(0)|^q + T^q\big),
\tag{A.2}
$$

and thus Eq. (3.10) follows from taking expectation with respect to the randomness in $\mathcal{F}_L$ on both sides and setting $q=4$. □

Acknowledgements

References

[1] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, 1987.

[2] S. Banerjee, B. P. Carlin, and A. E. Gelfand, *Hierarchical Modeling and Analysis for Spatial Data*, CRC Press, 2014.

[3] A. Barbu and S.-C. Zhu, *Hamiltonian and Langevin Monte Carlo*, In: *Monte Carlo Methods*, Springer, 281–325, 2020.

[4] G. Benettin and A. Giorgilli, *On the Hamiltonian interpolation of near-to-the identity symplectic mappings with application to symplectic integration algorithms*, J. Stat. Phys., 74(5-6):1117–1143, 1994.

[5] J. E. Besag, *Comments on "Representations of knowledge in complex systems" by U. Grenander and M. I. Miller*, J. R. Stat. Soc. Series B Stat. Methodol., 56(4):549–581, 1994.

[6] A. Beskos, N. Pillai, G. Roberts, J.-M. Sanz-Serna, and A. Stuart, *Optimal tuning of the hybrid Monte Carlo algorithm*, Bernoulli, 19(5A):1501–1534, 2013.

[7] M. Betancourt, *A conceptual introduction to Hamiltonian Monte Carlo*, arXiv:1701.02434.

[8] L. Bottou, *On-line Learning and Stochastic Approximations*, In: *On-Line Learning in Neural Networks (Publications of the Newton Institute)*, Cambridge University Press, 9–42, 1999.

[9] S. Bubeck, *Convex optimization: Algorithms and complexity*, Found. Trends Mach. Learn., 8(3-4):231–357, 2015.

[10] T. Chen, E. Fox, and C. Guestrin, *Stochastic gradient Hamiltonian Monte Carlo*, In: *International Conference on Machine Learning*, PMLR, 1683–1691, 2014.

[11] Y. Chen, R. Dwivedi, M. J. Wainwright, and B. Yu, *Fast mixing of Metropolized Hamiltonian Monte Carlo: Benefits of multi-step gradients*, J. Mach. Learn. Res., 2020.

[12] C. Dai and J. S. Liu, *Wang-Landau algorithm as stochastic optimization and its acceleration*, Phys. Rev. E, 101(3):033301, 2020.

[13] A. S. Dalalyan, *Theoretical guarantees for approximate sampling from smooth and log-concave densities*, J. R. Stat. Soc. Series B Stat. Methodol., 79(3):651–676, 2017.

[14] A. S. Dalalyan and A. Karagulyan, *User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient*, Stoch. Process Their Appl., 129(12):5278–5311, 2019.

[15] Z. Ding, Q. Li, J. Lu, and S. Wright, *Random coordinate underdamped Langevin Monte Carlo*, In: *International Conference on Artificial Intelligence and Statistics*, PMLR, 2701–2709, 2021.

[16] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, *Hybrid Monte Carlo*, Phys. Lett. B, 195(2):216–222, 1987.

[17] R. Durrett, *Probability: Theory and Examples*, Vol. 49, Cambridge University Press, 2019.

[18] F. J. Dyson, *A Brownian-motion model for the eigenvalues of a random matrix*, J. Math. Phys., 3(6):1191–1198, 1962.

[19] L. Erdos and H.-T. Yau, *A Dynamical Approach to Random Matrix Theory*, Courant Lecture Notes in Mathematics, 28, 2017.

[20] C. Fraley and A. E. Raftery, *Model-based clustering, discriminant analysis, and density estimation*, J. Am. Stat. Assoc., 97(458):611–631, 2002.

[21] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, Vol. 1, Elsevier, 2001.

[22] S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Mach. Intell., 6:721–741, 1984.

[23] C. J. Geyer, *Markov Chain Monte Carlo maximum likelihood*, In: *Computing Science and Statistics: Proceedings of 23rd Symposium on the Interface Interface Foundation*, 156–163, 1991.

[24] M. Giordano and R. Nickl, *Consistency of Bayesian inference with Gaussian process priors in an*

elliptic inverse problem, Inverse Probl., 36(8):085001, 2020.

[25] P. J. Green, *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*, Biometrika, 82(4):711–732, 1995.

[26] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Vol. 31, Springer Science & Business Media, 2006.

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

[28] B. Hetenyi, K. Bernacki, and B. J. Berne, *Multiple "time step" Monte Carlo*, J. Chem. Phys., 117(18):8203–8207, 2002.

[29] A. Jasra, C. C. Holmes, and D. A. Stephens, *Markov Chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling*, Stat. Sci., 20(1):50–67, 2005.

[30] S. Jin and L. Li, *Random batch methods for classical and quantum interacting particle systems and statistical samplings*, arXiv:2104.04337.

[31] S. Jin, L. Li, and J.-G. Liu, *Random batch methods (RBM) for interacting particle systems*, J. Comput. Phys., 400(1):108877, 2020.

[32] S. Jin, L. Li, and J.-G. Liu, *Convergence of the random batch method for interacting particles with disparate species and weights*, SIAM J. Numer. Anal., 59(2):746–768, 2021.

[33] S. Jin, L. Li, and Y. Sun, *On the random batch method for second order interacting particle systems*, arXiv:2011.10778.

[34] S. Jin, L. Li, Z. Xu, and Y. Zhao, *A random batch Ewald method for particle systems with Coulomb interactions*, arXiv:2010.01559.

[35] S. Jin and X. Li, *Random batch algorithms for quantum Monte Carlo simulations*, arXiv:2008.12990.

[36] J. K. Johnson, J. A. Zollweg, and K. E. Gubbins, *The Lennard-Jones equation of state revisited*, Mol. Phys., 78(3):591–618, 1993.

[37] A. Laio and M. Parrinello, *Escaping free-energy minima*, Proc. Natl. Acad. Sci., 99(20):12562–12566, 2002.

[38] S. Lan, J. Streets, and B. Shahbaba, *Wormhole Hamiltonian Monte Carlo*, In: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28, 2014.

[39] Y. T. Lee and S. S. Vempala, *Convergence rate of Riemannian Hamiltonian Monte Carlo and faster polytope volume computation*, In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 1115–1121, 2018.

[40] B. Leimkuhler and S. Reich, *Simulating Hamiltonian dynamics*, Vol. 14, Cambridge University Press, 2004.

[41] L. Li, Y. Li, J.-G. Liu, Z. Liu, and J. Lu, *A stochastic version of Stein variational gradient descent for efficient sampling*, Comm. App. Math. Comp. Sci., 15(1):37–63, 2020.

[42] L. Li, Z. Xu, and Y. Zhao, *A random-batch Monte Carlo method for many-body systems with singular kernels*, SIAM J. Sci. Comput., 42(3):A1486–A1509, 2020.

[43] L. Lovász and M. Simonovits, *Random walks in a convex body and an improved volume algorithm*, Random Struct. Algorithms, 4(4):359–412, 1993.

[44] O. Mangoubi, N. S. Pillai, and A. Smith, *Does Hamiltonian Monte Carlo mix faster than a random walk on multimodal densities?* arXiv:1808.03230.

[45] E. Marinari and G. Parisi, *Simulated tempering: a new Monte Carlo scheme*, EPL, 19(6):451, 1992.

[46] P. A. Markowich and C. Villani, *On the trend to equilibrium for the Fokker-Planck equation: An interplay between physics and functional analysis*, Math. Contemp., 19:1–31, 2000.

[47] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of*

state calculations by fast computing machines, J. Chem. Phys., 21(6):1087–1092, 1953.

[48] B. Miasojedow, E. Moulines, and M. Vihola, *An adaptive parallel tempering algorithm*, J. Comput. Graph. Stat., 22(3):649–664, 2013.

[49] R. M. Neal, *MCMC using Hamiltonian dynamics*, Handbook of Markov Chain Monte Carlo, 2(11):2, 2011.

[50] R. Nickl and S. Wang, *On polynomial-time computation of high-dimensional posterior measures by Langevin-type algorithms*, arXiv:2009.05298.

[51] M. Raginsky, A. Rakhlin, and M. Telgarsky, *Non-convex learning via stochastic gradient Langevin dynamics: A nonasymptotic analysis*, In: *Conference on Learning Theory*, PMLR, 1674–1703, 2017.

[52] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Stat., 22(3):400–407, 1951.

[53] G. O. Roberts and J. S. Rosenthal, *General state space Markov chains and MCMC algorithms*, Probab. Surv., 1:20–71, 2004.

[54] G. O. Roberts and R. L. Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli, 2(4):341–363, 1996.

[55] J. C. Sexton and D. H. Weingarten, *Hamiltonian evolution for the hybrid Monte Carlo algorithm*, Nucl. Phys. B, 380(3):665–677, 1992.

[56] B. Shahbaba, S. Lan, W. O. Johnson, and R. M. Neal, *Split Hamiltonian Monte Carlo*, Stat. Comput., 24(3):339–349, 2014.

[57] A. Smith, *Sequential Monte Carlo Methods in Practice*, Springer Science & Business Media, 2013.

[58] R. H. Swendsen and J.-S. Wang, *Replica Monte Carlo simulation of spin-glasses*, Phys. Rev. Lett., 57(21):2607, 1986.

[59] T. Tao, *Topics in Random Matrix Theory*, Vol. 132, AMS, 2012.

[60] F. Wang and D. P. Landau, *Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram*, Phys. Rev. E, 64(5):056101, 2001.

[61] F. Wang and D. P. Landau, *Efficient, multiple-range random walk algorithm to calculate the density of states*, Phys. Rev. Lett., 86(10):2050, 2001.

[62] G. Wanner and E. Hairer, *Solving Ordinary Differential Equations II*, Vol. 375, Springer, 1996.

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

[64] X. Ye and Z. Zhou, *Efficient sampling of thermal averages of interacting quantum particle systems with random batches*, J. Chem. Phys., 154(20):204106, 2021.