A RANDOM-BATCH MONTE CARLO METHOD FOR MANY-BODY SYSTEMS WITH SINGULAR KERNELS*

LEI LI†, ZHENLI XU†, AND YUE ZHAO†

Abstract. We propose a fast potential splitting Markov chain Monte Carlo method which costs $O(1)$ time each step for sampling from equilibrium distributions (Gibbs measures) corresponding to particle systems with singular interacting kernels. We decompose the interacting potential into two parts; one is of long range but is smooth, and the other one is of short range but may be singular. To displace a particle, we first evolve a selected particle using the stochastic differential equation (SDE) under the smooth part with the idea of random batches, as commonly used in stochastic gradient Langevin dynamics. Then, we use the short range part to do a Metropolis rejection. Different from the classical Langevin dynamics, we only run the SDE dynamics with a random batch for a short duration of time so that the cost in the first step is $O(p)$, where p is the batch size and is often chosen to be $O(1)$. The cost of the rejection step is $O(1)$ since the interaction used is of short range. We justify the proposed random-batch Monte Carlo method, which combines the random batch and splitting strategies, both in theory and with numerical experiments. While giving comparable results for typical examples of the Dyson Brownian motion and Lennard-Jones fluids, our method can save more time when compared to the classical Metropolis-Hastings algorithm.

Key words. Markov chain Monte Carlo, Langevin dynamics, random-batch method, stochastic differential equations

AMS subject classifications. 82B80, 60H35, 65C05

DOI. 10.1137/19M1302077

1. Introduction. The statistics of many-body systems such as pressure, energy, and entropy at equilibrium can describe the physical properties of the systems [23]. The values of these quantities correspond to expectations of some functions under the equilibrium, which is often approximated by using Monte Carlo sampling methods [1, 8]. The Markov chain Monte Carlo (MCMC) methods [11, 9] are among the most popular Monte Carlo methods. By constructing Markov chains that have the desired distributions to be the invariant measures, one can obtain samples from the desired distributions by recording the states of the Markov chains. A typical MCMC algorithm is the Metropolis-Hastings algorithm [28, 12].

In this work, we revisit this classical problem, namely sampling from equilibrium distributions for many-body interacting particle systems. Suppose that we want to sample from the N-particle Gibbs distribution (or generally the Boltzmann distribution) [23]

$$\pi(\boldsymbol{X}) \propto \exp\left[-\beta H(\boldsymbol{X})\right], \tag{1.1}$$

with $\boldsymbol{X} = (x_1, \ldots, x_N) \in \mathbb{R}^{Nd}$ ($x_i \in \mathbb{R}^d$, and $d \geq 1$, $d \in \mathbb{N}$), β being a positive

---

constant, the N-body energy

$$(1.2) \qquad H(\boldsymbol{X}) := \sum_{i=1}^{N} w_i V(x_i) + \sum_{i,j:i<j} w_i w_j U(x_i - x_j),$$

and V being the external potential which is assumed to be smooth. Without loss of generality, the kernel for interaction potential $U(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is assumed to be of the symmetric form $U(x) = u(|x|)$ and is possibly singular and of long range. The weight w_i is often the mass or charge of the particle measuring how important the ith particle is. In this work, we will only consider $w_i \equiv w$, and the generalization of the proposed algorithm to nonuniform weights is straightforward.

As mentioned above, the classical algorithms to sample from the Gibbs distribution are MCMC algorithms, including the Metropolis algorithm. Later, there were some variants like the multiple time-step MC algorithm (MTS-MC) [13] using the splitting idea. Often, these algorithms update a randomly chosen particle per Markov jump in the state space, as updating one particle per iteration can be more efficient (see [8] for more explanation). A rejection step is performed following some random movement. The acceptance probability is computed using the difference between the energies of two configurations according to the detailed balance condition. Hence, computing the acceptance probability often takes $O(N)$ time after moving a randomly chosen particle. For this reason, throughout this paper, we will use "an iteration" or "a move" to mean such a Markov jump together with the rejection step.

Among MCMC algorithms, the Langevin sampling methods make use of stochastic differential equations (SDEs) [2, 35, 6, 31]. In fact, if one moves X^i according to the SDE system [30]

$$(1.3) \qquad dX^i = -\gamma \left[w \nabla V(X^i) + w^2 \sum_{j:j\neq i} \nabla U(X^i - x_j) \right] dt + \sqrt{\frac{2\gamma}{\beta}} \, dW^i,$$

with $X^i(0)$ being arbitrary and $\{W^i\}$ being independent standard d-dimensional Wiener processes (Brownian motions) for $i = 1, \dots, N$, then the invariant measure is the Gibbs distribution (1.1). For the Langevin dynamics, one often desires the forcing term to be of order 1, so one may choose

$$(1.4) \qquad \gamma = \frac{1}{w^2(N-1)}.$$

For the equations with a big summation on the right-hand side, one may apply the minibatch ideas [34, 3, 4, 17] to reduce the cost. The random minibatch idea was famous by its application in the stochastic gradient descent (SGD) [34, 3]. Recently, it has been applied to interacting particle systems by Jin, Li, and Liu yielding a fast algorithm called the random-batch method (RBM) [17], which is asymptotic-preserving in the mean field limit regime. This algorithm not only computes the equilibrium distribution correctly, but it can also capture the dynamics of the interacting particle systems. Building the minibatch ideas into the Langevin sampling leads to the stochastic gradient Langevin dynamics (SGLD) [39, 25], which is a fast approximate Monte Carlo method and has been widely used for Bayesian inference problems [26]. Recently, some particle based variational inference sampling methods, which are not Monte Carlo methods, have been proposed (see [24, 32, 5]). These methods update particles by solving optimization problems, and each iteration is expected to make

progress. The Stein variational gradient descent [24] is a typical example. Applying the random-batch idea to it can yield an efficient sampling method [21].

In this paper, we combine the splitting idea (as in MTS-MC) and the random minibatch idea (as in SGLD) to get a fast MCMC sampling algorithm. We decompose the interacting potential into two parts; one is of long range but is smooth, and the other one is of short range but may be singular. As commented on in [8, Chap. 3] and mentioned above, we choose to update only one particle per move. To move a selected particle, we first evolve it using the SDE under the smooth part with the idea of random batches, as commonly used in SGLD. Then, we use the short range part to do a Metropolis rejection. Different from SGLD, we only run the SDE dynamics with a random batch for a short duration of time so that the cost in the first step is $O(p)$, where p is the batch size and is often chosen to be $O(1)$. The cost of the rejection step is $O(1)$ since the interaction used is of short range. The process of our method seems reminiscent of the Metropolis-adjusted Langevin algorithm (MALA) [2, 35], where a discretized SDE was run first and then a Metropolis rejection was performed. However, we point out that the Metropolis rejection in MALA is only to correct the small errors introduced by the discretization. In our approach, we have the splitting explicitly and the rejection step aims to avoid unphysical configurations where one particle is moved to the singular core of the nearby particle, which greatly improves the sampling efficiency.

Let us have a short discussion of the regimes to consider. In most cases, we focus on the molecular regime so that $w = 1$ and $\beta = O(1)$, and hence the number $\sqrt{2\gamma/\beta}$ by (1.4) is often small. In the mean field regime [37, 10, 20], one may have $w \sim N^{-1}$ (see the discussion below and the example in section 4.1). Each particle satisfies the following equation:

$$(1.5) \qquad dX^i = -\left[\nabla V(X^i) + \frac{1}{N}\sum_{j:j\neq i}\nabla U(X^i - x_j)\right] dt + \sqrt{\frac{2N}{\beta}}\, dW^i.$$

As is well known [16, 22], if $\beta \sim N$ and $\sqrt{2\gamma/\beta} = O(1)$, the stationary *single particle* distribution can be approximated by the solution to the nonlinear Fokker–Planck equation

$$(1.6) \qquad \nabla \cdot (\nabla V \rho) + \nabla \cdot (\rho \nabla U * \rho) + \Delta \rho = 0.$$

The rest of the paper is organized as follows. In section 2, we propose the splitting strategy and give its justifications. Based on this splitting, a move of the Markov chain consists of two parts; one is the evolution of an SDE under the smooth part of the potential, while the other one is the rejection under the short range part of the potential. We verify that the Gibbs distribution is the invariant measure of the constructed Markov chain. In section 3, we apply the idea of random minibatches to the SDE step and obtain our sampling algorithm. Section 4 is devoted to the numerical justifications, where we compare our method with the Metropolis-Hastings algorithm on the Dyson Brownian motion example and Lennard-Jones fluids. Conclusions are made in section 5.

2. Splitting Monte Carlo method. In practice, the interaction kernel between a pair of particles is often singular and possibly with long range. The typical example is the potential kernel of the form $U(x) = 1/|x|^\alpha$ for $\alpha > 0$. See section 4.1 for an example. Other examples include the famous Coulomb potential and the London

potential [15]. If we do sampling directly using the Metropolis-Hastings algorithm [12] (see also Algorithm 2 in section 4), to get a proposed move, one must consider the interaction with all other $N - 1$ particles, which makes the computational cost of the algorithm $O(N)$ per move. Moreover, due to the singular interaction, the acceptance rate can be lower.

Inspired by the MTS-MC method [13], we consider a splitting strategy for the binary potential U such that the single-variable function $u(r)$ becomes the summation of $u_1(r)$ and $u_2(r)$. Let $U_i(x) := u_i(|x|)$, and correspondingly we have

$$(2.1) \qquad U(x) = U_1(x) + U_2(x),$$

where we suppose that U_1 has long range but is smooth and bounded, and U_2 is singular and compactly supported. The splitting is convenient because one can apply the Langevin dynamics [35, 6, 31] for the U_1 part. The singular U_2 part can then be used for the rejection.

There are many ways to split the potential. In the adiabatic nuclear and electronic sampling Monte Carlo (ANES-MC) method [27], the splitting is to decompose U into "fast" and "slow" dynamics. In the MTS-MC method, the potential is decomposed into short-range and long-range parts; the long-range part is evaluated less frequently than the short range part to improve the efficiency. In our method, we also split the potential into short-range and long-range parts, but our motivation is different from MTS-MC. In fact, we aim to build the random minibatch idea in SGD and SGLD into the algorithm to reduce the cost, but the minibatch method works well for smooth potentials. Naturally, we do the splitting and ask the long-range part U_1 to be smooth and use the singular part to do rejection. In section 4.2, we will see that for Lennard-Jones potential, our method will have comparable CPU speedup with that of MTS-MC in [13] when $N = 500$. Since our algorithm is $O(1)$ per iteration, we expect that the cost to run a cycle of samples will be linear in N as N grows. In our current implementation, the splitting is kind of far from the best. How to optimize the splitting to gain a better spectral gap of the semigroup for the Markov chain is an interesting question left for future.

Below, we aim to justify this splitting by showing that the desired Gibbs distribution is an invariant measure of the Markov chain.

Suppose there are N particles located at x_j for $j = 1, \ldots, N$. Let us consider the following method for a Markovian jump:

Step 1. Randomly choose a particle i.

Step 2. Move the particle using U_1 with an overdamped Langevin equation:

$$(2.2) \qquad dX^i = -\left(\frac{\nabla V(X^i)}{w(N-1)} + \frac{1}{N-1} \sum_{j:j \neq i} \nabla U_1(X^i - x_j) \right) dt + \sqrt{\frac{2}{(N-1)w^2 \beta}} \, dW^i,$$

$$X^i(0) = x_i,$$

where x_j's are fixed. Evolve this SDE with some time $t > 0$ and obtain $X^i(t) \to x_i^*$ as a candidate position of particle i for the new sample.

Step 3. Use U_2 to do the Metropolis rejection. Define

$$(2.3) \qquad acc(x_i, x_i^*) = \min \left\{ 1, \exp\left[-\beta \sum_{j:j \neq i} w^2 (U_2(x_i^* - x_j) - U_2(x_i - x_j)) \right] \right\}.$$

With probability $acc(x_i, x_i^*)$, we accept x_i^* and set $x_i \leftarrow x_i^*$. Otherwise, x_i is unchanged. We then obtain a new sample $\{x_1, \ldots, x_N\}$ in the Markov chain.

In comparison with the multiple time-step splitting algorithm, we do not have rejection in Step 2, while Step 3 can be done in $O(1)$ time since U_2 is local. The correctness of this splitting is guaranteed by Theorem 2.2. Before stating this theorem, we have the following lemma.

LEMMA 2.1. *The SDE* (2.2) *has invariant measure*

$$(2.4) \qquad \pi_1(z) \propto \exp\left[-\beta\left(wV(z) + w^2 \sum_{j:j \neq i} U_1(z - x_j)\right)\right].$$

Moreover, if $p(z, t; x_i)$ denotes the transition density from x_i, then the detailed balance condition holds:

$$(2.5) \qquad \pi_1(x_i)p(x_i^*, t; x_i) = \pi_1(x_i^*)p(x_i, t; x_i^*).$$

The claims here are standard in SDE theory, and we attach a proof in Appendix A for the convenience of the readers.

Now we have the following theorem on the detailed balance of the algorithm composed of Steps 1–3, namely, the Langevin dynamics step using U_1 and the Metropolis rejection step using U_2 for a randomly chosen particle.

THEOREM 2.2. *Let $\boldsymbol{X} = (x_1, \ldots, x_N)$ be a point in the state space of the N particles. Let $T(\boldsymbol{X}, \boldsymbol{X}^*)$ be the transition probability density for Steps* 1–3. *The splitting Monte Carlo satisfies the following detailed balance:*

$$(2.6) \qquad \pi(\boldsymbol{X})T(\boldsymbol{X}, \boldsymbol{X}^*) = \pi(\boldsymbol{X}^*)T(\boldsymbol{X}^*, \boldsymbol{X}),$$

where $\pi(\boldsymbol{X})$ is given by (1.1).

Proof. Note that the method consists of Steps 1–3 and only one particle is updated. Hence, $T(\boldsymbol{X}, \boldsymbol{X}^*)$ is nonzero only if $\boldsymbol{X}$ and $\boldsymbol{X}^*$ are different by one component at most. Hence, we only have to check two states where the ith component is different.

Without loss of generality, we assume $\boldsymbol{X}^* = (x_1, \ldots, x_i^*, \ldots, x_N)$ and $x_i \neq x_i^*$. Clearly,

$$(2.7) \qquad T(\boldsymbol{X}, \boldsymbol{X}^*) = \frac{1}{N}p(x_i^*, t; x_i)acc(x_i, x_i^*),$$

where $1/N$ is the probability for choosing the random particle.

Note that

(2.8)

$$\pi(\boldsymbol{X}) = \frac{1}{Z_N} \exp\left[-\beta\left(\sum_i wV(x_i) + \sum_{j,k:j<k} w^2 U(x_j, x_k)\right)\right]$$

$$= \frac{\pi_1(x_i)\pi_2(x_i)}{Z_3} \exp\left[-\beta \sum_{\ell:\ell \neq i} wV(x_\ell)\right] \exp\left[-\beta \sum_{j<k:j,k \neq i} w^2 U(x_j - x_k)\right],$$

where

$$(2.9) \qquad \pi_2(z) \propto \exp\left[-\beta \sum_{j:j \neq i} w^2 U_2(z - x_j)\right],$$

and Z_N and Z_3 are some normalizing constants. Consequently, the detailed balance condition (2.6) is reduced to

$$(2.10) \qquad \pi_1(x_i)\pi_2(x_i)p(x_i^*,t;x_i)acc(x_i,x_i^*) = \pi_1(x_i^*)\pi_2(x_i^*)p(x_i,t;x_i^*)acc(x_i^*,x_i).$$

Using the detailed balance (2.5) for the SDE, (2.10) is then reduced to

$$(2.11) \qquad \pi_2(x_i)acc(x_i,x_i^*) = \pi_2(x_i^*)acc(x_i^*,x_i),$$

which clearly holds by the formula of the Metropolis rejection probability (2.3). $\qquad\square$

Remark 2.3. The detailed balance (2.5) allows us to solve the SDE on a fixed time interval, far before reaching the equilibrium. If the SDE $dX = b(X)\,dt + \sigma(X)dW$ does not have detailed balance (or $-b\pi + \frac{1}{2}\nabla \cdot (\sigma\sigma^T\pi) \neq 0$), then Step 2 should be run for a long time so that $p(x,t;x_i) \approx \pi_1(x)$. This is a problem since the Euler–Maruyama scheme can be guaranteed to converge on a finite time interval, but it may not converge for an infinite time interval.

3. Random-batch sampling algorithm. Note that Step 3 can be done in $O(1)$ operations using some standard data structures, such as the cell list [8, Appendix F], so we want to reduce the time cost in Step 2 to $O(1)$ as well. The idea is to use the minibatch approach, similar to the idea in SGLD [39, 25].

3.1. Discretization of the SDE with random batch. We now focus on Step 2 and discretize the SDE with the Euler–Maruyama scheme [19, 29]. The interaction force is approximated by that within the minibatch. This then gives the following discrete approximation.

Pick a batch size $p > 1$. Let $X^i \leftarrow x_i$. Run the following approximation for m steps ($k = 0, \ldots, m-1$), where $\boldsymbol{\xi}_k$ is a random set with size $p-1$ at step k, which is a subset of $\{1, 2, \ldots, N\} \setminus \{i\}$:

$$(3.1) \quad X^i \leftarrow X^i - \tau_k\left[\frac{\nabla V(X^i)}{w(N-1)} + \frac{1}{p-1}\sum_{j \in \boldsymbol{\xi}_k} \nabla U_1(X^i - x_j)\right] + \sqrt{\frac{2\tau_k}{(N-1)w^2\beta}}z_k,$$

where $\tau_k > 0$ is a step size and $z_k \sim \mathcal{N}(0, I_d)$. Here $\mathcal{N}(\mu, \Sigma)$ represents the multivariate normal distribution with mean μ and covariant matrix Σ.

In practice, the number m is independent of N and a small constant in order to reduce the algorithm complexity. The scheme (3.1) can be further written as, for $k = 1, \ldots, m$,

$$(3.2)$$
$$X^i \leftarrow X^i - \tau_k\left[\frac{\nabla V(X^i)}{w(N-1)} + \frac{1}{N-1}\sum_{j:j\neq i} \nabla U_1(X^i - x_j)\right] + \sqrt{\frac{2\tau_k}{(N-1)w^2\beta}}z_k + \tau_k\zeta_k^i,$$

where

$$(3.3) \qquad \zeta_k^i := -\frac{1}{p-1}\sum_{j \in \boldsymbol{\xi}_k} \nabla U_1(X^i - x_j) + \frac{1}{N-1}\sum_{j:j\neq i} \nabla U_1(X^i - x_j).$$

The following lemma is straightforward, and we omit its proof (see [14, 17] for similar results).

LEMMA 3.1. *Random variable ζ_k^i has zero mean, and the covariance matrix is given by*

$$(3.4) \qquad \text{cov}(\zeta_k^i) = \left(\frac{1}{p-1} - \frac{1}{N-1} \right) \Lambda(x_1, \ldots, x_{i-1}, X^i, x_{i+1}, \ldots, x_N),$$

where Λ is defined by

$$\Lambda(x_1, \ldots, x_N) := \frac{1}{N-2} \sum_{j:j \neq i} [\nabla U_1(x_i - x_j) - G(x_i)] \otimes [\nabla U_1(x_i - x_j) - G(x_i)]$$

with

$$G(x_i) = \frac{1}{N-1} \sum_{j:j \neq i} \nabla U_1(x_i - x_j).$$

The variance of the term $\tau_k \zeta_k^i$ is thus of order τ_k^2, which is small. Hence, this gives an effective approximation for the SDE. In our algorithm, the SGLD is performed on finite interval $[0, T]$ with $T = O(1)$ or $T \ll 1$. We now estimate the error in the transition probability introduced by discretizing the SDE and applying the minibatch idea.

For convenience, we only consider constant step size in this section: $\tau_k \equiv \tau > 0$. To justify the minibatch discretization proposed above, we will show that the transition probabilities are close in the Wasserstein distance. We recall that the Wasserstein-2 distance [36] is given by

$$(3.5) \qquad W_2(\mu, \nu) = \left(\inf_{\gamma \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} |x - y|^2 d\gamma \right)^{1/2},$$

where $\Pi(\mu, \nu)$ means all the joint distributions whose marginal distributions are μ and ν, respectively.

We in fact have the following theorem for the error estimate. In practice, we take $T = m\tau$ with $m = O(1)$ so the error is in fact of order τ.

THEOREM 3.2. *Let $\lambda = 1/w(N-1)$ and $D = 1/\beta w^2(N-1)$. Suppose $\lambda \lesssim O(1)$. Suppose that V and U_1 are smooth, and the derivatives up to third order are bounded. Let $Q(x, \cdot)$ be the transition probability of the SDE (2.2) and $\widetilde{Q}(x, \cdot)$ be the transition probability computed by the discretization with random batch (3.1). Then,*

$$(3.6) \qquad \sup_x W_2(Q(x, \cdot), \widetilde{Q}(x, \cdot)) \leq C \sqrt{(e^{CT} - 1) \left(\frac{1}{p-1} \|\nabla U_1\|_\infty \tau + (1 + D^2)\tau^2 \right)}.$$

Proof. For convenience, in this proof, we will denote the chosen X^i by Y and $f_j(y) := U_1(y - x_j)$. Without loss of generality, we assume that $i = N$ and $j \in \{1, \ldots, N-1\}$. Denote the $L^2(\mathbb{P})$ norm $\| \cdot \| := \sqrt{\mathbb{E}| \cdot |^2}$. To simplify the notation, we also define

$$(3.7) \qquad g_1(y) := -\lambda \nabla V(y) \quad \text{and} \quad g_2(y) := -\frac{1}{N-1} \sum_{j=1}^{N-1} \nabla f_j(y).$$

The desired SDE is then given by

$$(3.8) \qquad dY = g_1(Y)\, dt + g_2(Y)\, dt + \sqrt{2D}\, dW, \quad t \in [0, T].$$

Define $t_k := k\tau$. The numerical algorithm is given by

$$Y^{k+1} = Y^k + g_1(Y^k)\tau - \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y^k)\tau + \sqrt{2D}\Delta W_k, \quad k = 0,\ldots,m-1.$$

To compare their distributions, we apply the standard coupling technique, namely,

$$(3.9) \qquad\qquad \Delta W_k = W(t_{k+1}) - W(t_k),$$

where $W(\cdot)$ is the same Brownian motion used in (3.8).

To estimate the error of the numerical solution in comparison to the solution of the SDE, we consider the following equation for $t \in [t_k, t_{k+1}]$:

$$(3.10) \qquad dY_\tau(t) = g_1(Y_\tau(t_k))\,dt - \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k))\,dt + \sqrt{2D}\,dW.$$

It is clear that

$$Y_\tau(t_k) = Y^k.$$

By Itô's calculus, it is easily found that

$$(3.11) \quad \frac{d}{dt}\mathbb{E}|Y(t) - Y_\tau(t)|^2 = 2\mathbb{E}(Y(t) - Y_\tau(t)) \cdot \left\{ [g_1(Y(t)) - g_1(Y_\tau(t_k))] \right.$$

$$\left. + g_2(Y(t)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k)) \right\}.$$

Using (3.8) and (3.10),

$$Y(t) - Y_\tau(t) = Y(t_k) - Y_\tau(t_k) + \int_{t_k}^t [g_1(Y(s)) - g_1(Y_\tau(t_k))]\,ds$$

$$+ \int_{t_k}^t \left[g_2(Y(s)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k)) \right]\,ds.$$

Using this formula, we can decompose the right-hand side of (3.11) as $I_1 + I_2$, where

$$(3.12) \quad I_1 = 2\mathbb{E}\left[Y(t_k) - Y_\tau(t_k) + \int_{t_k}^t [g_1(Y(s)) - g_1(Y_\tau(t_k))]\,ds \right]$$

$$\cdot \left[(g_1(Y(t)) - g_1(Y_\tau(t_k))) + g_2(Y(t)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k)) \right]$$

$$+ 2\mathbb{E}\int_{t_k}^t \left[g_2(Y(s)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k)) \right]\,ds \cdot \left[g_1(Y(t)) - g_1(Y_\tau(t_k)) \right]$$

and

$$(3.13) \quad I_2 = \mathbb{E}\int_{t_k}^t \left[g_2(Y(s)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k)) \right]$$

$$\cdot \left[g_2(Y(t)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k)) \right]\,ds.$$

Since $Y(t)$ and $Y_\tau(t_k)$ are independent of the random batch at t_k (namely, $\boldsymbol{\xi}_k$), one has by Lemma 3.1 that

$$
(3.14) \quad I_1 = 2\mathbb{E}\Bigg\{ \left[Y(t_k) - Y_\tau(t_k) + \int_{t_k}^t (g_1(Y(s)) - g_1(Y_\tau(t_k)))\, ds \right]
$$
$$
\cdot \left[g_1(Y(t)) + g_2(Y(t)) - g_1(Y_\tau(t_k)) - g_2(Y_\tau(t_k)) \right] \Bigg\}
$$
$$
+ 2\mathbb{E}\int_{t_k}^t \left[g_2(Y(s)) - g_2(Y_\tau(t_k)) \right] \cdot \left[g_1(Y(t)) - g_1(Y_\tau(t_k)) \right] ds.
$$

In I_1, we need to estimate two types of terms. The first type is like

$$
J_1 := \mathbb{E}\left[Y(t_k) - Y_\tau(t_k) \right] \cdot \left[g_\ell(Y(t)) - g_\ell(Y_\tau(t_k)) \right],
$$

while the second type is like

$$
J_2 := \mathbb{E}\int_{t_k}^t \left[g_\ell(Y(s)) - g_\ell(Y_\tau(t_k)) \right] \cdot \left[g_m(Y(t)) - g_m(Y_\tau(t_k)) \right] ds,
$$

where $\ell, m = 1$ or 2. Note that

$$
(3.15) \qquad\qquad dY = (g_1 + g_2)(Y(t))\, dt + \sqrt{2D}\, dW(t).
$$

By Itô's formula, one has

$$
(3.16) \qquad g_\ell(Y(t)) = g_\ell(Y(t_k)) + \int_{t_k}^t dY \cdot \nabla g_\ell(Y(s)) + \int_{t_k}^t \Delta g_\ell(Y(s)) D\, ds.
$$

Hence,

$$
J_1 \le C\|Y(t_k) - Y_\tau(t_k)\|^2 + C\|Y(t_k) - Y_\tau(t_k)\|(1 + D)\tau,
$$

where we used the fact that

$$
\mathbb{E}\left[Y(t_k) - Y_\tau(t_k) \right] \cdot \int_{t_k}^t dY \cdot \nabla g_\ell(Y(s)) = \mathbb{E}\left[Y(t_k) - Y_\tau(t_k) \right] \cdot \int_{t_k}^t (g_1 + g_2) \cdot \nabla g_\ell(Y(s))\, ds.
$$

Using similar expansions, the terms like J_2 can be controlled as

$$
J_2 \le C\|Y(t_k) - Y_\tau(t_k)\|^2\tau + C\|Y(t_k) - Y_\tau(t_k)\|(1 + D)\tau^2 + CD\tau^2 + C(1 + D^2)\tau^3.
$$

Overall, we have

(3.17)
$$
I_1 \le C\|Y(t_k) - Y_\tau(t_k)\|^2 + C\|Y(t_k) - Y_\tau(t_k)\|(1 + D)\tau + CD\tau^2 + C(1 + D^2)\tau^3
$$
$$
\le C\|Y(t_k) - Y_\tau(t_k)\|^2 + C(1 + D^2)\tau^2.
$$

We now proceed with the estimate of I_2. Using again (3.16), one obtains

$$I_2 \leq \mathbb{E}\left| -\frac{1}{N-1}\sum_{j=1}^{N-1}\nabla f_j(Y(t_k)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k))\right|^2 \tau$$
$$+ C\left\| -\frac{1}{N-1}\sum_{j=1}^{N-1}\nabla f_j(Y(t_k)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k))\right\|(1+D)\tau^2$$
$$+ CD\tau^2 + C(1+D^2)\tau^3.$$

The first term is nothing but the variance of the random force, and thus Lemma 3.1 gives

$$I_2 \leq 2\left(\frac{1}{p-1} - \frac{1}{N-1}\right)\|\Lambda\|_\infty\tau + CD\tau^2 + C(1+D^2)\tau^3.$$

Note that the factor "2" appears because we applied the inequality

(3.18)
$$\left\| -\frac{1}{N-1}\sum_{j=1}^{N-1}\nabla f_j(Y(t_k)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k))\right\|(1+D)\tau^2$$
$$\leq \left\| -\frac{1}{N-1}\sum_{j=1}^{N-1}\nabla f_j(Y(t_k)) + \frac{1}{p-1}\sum_{j\in\boldsymbol{\xi}_k}\nabla f_k(Y_\tau(t_k))\right\|^2\tau + \frac{1}{4}(1+D)^2\tau^3.$$

Hence, we eventually have

$$\|Y(t_{k+1}) - Y_\tau(t_{k+1})\|^2 = \|Y(t_k) - Y_\tau(t_k)\|^2 + \int_{t_k}^{t_{k+1}}(I_1(s) + I_2(s))\,ds$$
$$\leq (1+C\tau)\|Y(t_k) - Y_\tau(t_k)\|^2 + 2\left(\frac{1}{p-1} - \frac{1}{N-1}\right)\|\Lambda\|_\infty\tau^2 + C(1+D^2)\tau^3.$$

This implies that

(3.19)
$$\|Y(t_k) - Y_\tau(t_k)\|^2 \lesssim (e^{Ck\tau} - 1)\left(\frac{1}{p-1}\|\Lambda\|_\infty\tau + (1+D^2)\tau^2\right)$$
$$\lesssim (e^{Ck\tau} - 1)\left[\frac{1}{p-1}\|\nabla U_1\|_\infty\tau + (1+D^2)\tau^2\right].$$

By the definition of W_2 distance (3.5), the claim follows. $\quad\square$

3.2. The algorithm and some comments. We now combine the ideas in sections 2 and 3.1 into a practical algorithm, called the random-batch Monte Carlo (RBMC) method (detailed in Algorithm 1). In particular, we use the splitting strategy and decompose U into a smooth part with long range and a singular part with short range. For the smooth part, we apply the minibatch idea for the SDE steps similar to SGLD, while we use the singular part for the rejection.

Algorithm 1 (Random-batch Monte Carlo algorithm)

1: Split $U := U_1 + U_2$ such that U_1 is smooth and with long range and U_2 is with short range. Randomly generate N initial particles, set N_s to be the total number of samples, and choose $p > 1$, $m \geq 1$

2: **for** n in $1 : N_s$ **do**

3:      Randomly choose a particle $i \in \{1, \ldots, N\}$ with uniform probability

4:      $X^i \leftarrow x_i$

5:      **for** $k = 1, \ldots, m$ **do**

6:          Choose $\boldsymbol{\xi}_k$, $z_k \sim \mathcal{N}(0, I_d)$, $\tau_k > 0$, and let

$$X^i \leftarrow X^i - \tau_k \left[\frac{\nabla V(X^i)}{w(N-1)} + \frac{1}{p-1} \sum_{j \in \boldsymbol{\xi}_k} \nabla U_1(X^i - x_j) \right] + \sqrt{\frac{2\tau_k}{(N-1)w^2\beta}} z_k$$

7:      **end for**

8:      Let $x_i^* \leftarrow X^i$. Compute the following using the cell list or other data structures:

$$\alpha = \min \left\{ 1, \exp \left[-\beta \sum_{j:j \neq i} w^2(U_2(x_i^* - x_j) - U_2(x_i - x_j)) \right] \right\}$$

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

$$x_i \leftarrow x_i^*$$

10: **end for**

Compared to the traditional Metropolis algorithm [12], our new algorithm (Algorithm 1) has the following advantages:

- The computational cost is $O(1)$ for each iteration.
- There is no rejection in step 2. The efficiency could be higher.

When we run the implementations, there are two phases for the algorithms. Phase 1 is the burn-in phase, in which the distribution of the Markov chain converges the desired Gibbs distribution. Phase 2 is the sampling phase. We run the algorithms for many iterations and collect these N points for selected iterations as the sampling points. As the iteration goes on, the number of sampling points gets larger and larger.

Below, we have some discussions:

1. For the molecular regime, $w = 1$. Taking $\gamma = 1$ in (1.3) and using the same splitting, one may have the following SDE:

$$(3.20) \qquad dX^i = -\sum_{j:j \neq i} \nabla U_1(X^i - x_j)\, dt + \sqrt{\frac{2}{\beta}}\, dW^i.$$

If one uses minibatch idea, the random force then becomes

$$-\frac{N-1}{p-1} \sum_{j \in \boldsymbol{\xi}_k} \nabla U_1(X^i - x_j).$$

For this type of forcing term, the magnitude can be as large as $O(N)$, and one needs $\tau_k \sim O(1/N)$ for the numerical methods to be stable. This is

not preferred in practice. Taking $\gamma = 1/(N-1)$ corresponds to nothing but rescaling the time with $\tilde{t} = (N-1)t$.

2. If we move N particles, then step 2 can be solved using the RBM in [17] to reduce the cost. However, moving N particles will make the following event have high probability:

$$A = \{\exists(i,j) \text{ such that } |x_i^* - x_j^*| \leq \delta * \text{mean distance}\}.$$

This event will be rejected with high probability in step 3. Hence, moving N particles is not preferred in practice.

4. Numerical examples. In this section, we test our RBMC sampling algorithm on two examples, and compare to the Metropolis-Hastings algorithm (see Algorithm 2), which is a classical MCMC method. The first example is the interacting particle system from the Dyson Brownian motion with the Gibbs measure to be the semicircle law in the $N \to \infty$ limit. The second example is the Lennard-Jones fluids, and we aim to recover the equation of states $P = P(\rho)$, where P is the pressure and ρ is the density. Both examples have singular interaction potentials. The difference is that the first example has a long range while the second example has a short range. Even for short-range potentials, our algorithm can still be applied.

Below, an "iteration" will refer to one complete move of a particle, including the rejection step. In other words, this will correspond to the loop of n in Algorithms 1 and 2. All the numerical experiments are implemented using C++ and performed in a Windows system with Intel Core i7-6700HQ CPU @ 2.60GHz and 8 GB memory.

Algorithm 2 (Metropolis-Hastings algorithm)

1: **for** n in $1 : N_s$ **do**
2:     Randomly choose a particle $i \in \{1, \ldots, N\}$ with uniform probability
3:     Move x_i with some random movement and obtain x_i^*
4:     Compute the following using the cell list or other data structures:

$$\alpha = \min\left\{1, \exp\left[-\beta w(V(x_i^*) - V(x_i)) - \beta w^2 \sum_{j:j\neq i} (U(x_i^* - x_j) - U(x_i - x_j))\right]\right\}$$

5:     Generate $\zeta \sim U[0,1]$. If $\zeta \leq \alpha$, set

$$x_i \leftarrow x_i^*$$

6: **end for**

4.1. Dyson Brownian motion. To test our method, we first focus on an interacting particle system from Dyson Brownian motion arising from random matrix theory [38, 7]. Consider a Hermitian matrix valued Ornstein–Uhlenbeck process

$$(4.1) \qquad\qquad dA = -A\,dt + \frac{1}{\sqrt{N}}dB,$$

where the matrix B is a Hermitian matrix whose diagonal elements are independent standard Brownian motions while the off-diagonal elements in the upper triangular half are of the form $(B_R + iB_I)/\sqrt{2}$, with B_R and B_I being independent standard

Brownian motions. It can be shown [38, 7] that the eigenvalues of A satisfy the following system of SDEs ($1 \leq j \leq N$), called the Dyson Brownian motion:

$$(4.2) \qquad d\lambda_j(t) = -\lambda_j(t)\,dt + \frac{1}{N}\sum_{k:k\neq j}\frac{1}{\lambda_j - \lambda_k}dt + \frac{1}{\sqrt{N}}dW_j,$$

where $\{W_j\}$'s are independent standard Brownian motions and we use W_j instead of B_j to be consistent with the notations throughout this paper. The density of these eigenvalues in the limit $N \to \infty$ is given by

$$(4.3) \qquad \partial_t\rho(x,t) + \partial_x(\rho(u-x)) = 0, \quad u(x,t) = \pi(H\rho)(x,t) = \mathrm{p.v.}\int_{\mathbb{R}}\frac{\rho(y,t)}{x-y}\,dy,$$

where $H(\cdot)$ is the Hilbert transform on $\mathbb{R}$, $\pi = 3.14\cdots$ is the circumference ratio, and p.v. means the integral is evaluated using the Cauchy principal value. The corresponding limiting equation (4.3) has an invariant measure, given by the semicircle law:

$$(4.4) \qquad \rho(x) = \frac{1}{\pi}\sqrt{2 - x^2}.$$

Note that the mean field limit (4.3) is the same for any SDE of the following form:

$$(4.5) \qquad d\lambda_j(t) = -\lambda_j(t)\,dt + \frac{1}{N-1}\sum_{k:k\neq j}\frac{1}{\lambda_j - \lambda_k}dt + \sqrt{\frac{2\mu}{N-1}}dW_j.$$

We changed the prefactor $1/N$ into $1/(N-1)$ for convenience.

For the system (4.5), clearly we have $w = 1/(N-1)$ and the physical temperature in this case is given by $T = \mu/(N-1)^2$, and thus $\beta = (N-1)^2/\mu$. The corresponding Gibbs distribution is

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

The binary potential is given by $U(x_i - x_j) = u(|x_i - x_j|) = -\ln|x_i - x_j|$. We consider $N = 500$ particles sampled from the uniform distribution on $[-5, 5]$ and run Algorithm 1. Let $r = |x_i - x_j|$. We consider the splitting $U = U_1 + U_2$ with $U_i(x) = u_i(|x|)$, $i = 1, 2$, where

$$(4.6) \qquad u_1(r) = \begin{cases} \ln(100) - 100r + 1, & 0 < r < 0.01, \\ -\ln(r), & r \geq 0.01, \end{cases}$$

and

$$(4.7) \qquad u_2(r) = \begin{cases} -\ln(100r) + 100r - 1, & 0 < r < 0.01, \\ 0 & r \geq 0.01. \end{cases}$$

Clearly, $U_1(x_i - x_j) = u_1(r)$ is the interaction extending to $r = 0$ along the tangent line at $r = 0.01$. The short-range potential $u_2(r)$ is nonzero only if $r < 0.01$, so we use the cell list to calculate $U_2(x_i - x_j)$ in the nearby boxes of a particular particle with length 0.01. The reason for choosing 0.01 as the splitting point is that we hope that there are only a few particles in the range of U_2 for a chosen particle.

Burn-in phase of the two methods. The first 3e5 iterations in the MH method are discarded, which costs 36.8s. The first 3e6 iterations are discarded in the RBMC method, which costs 14.5s.

	Iteration	Time (s)
MH	3e5	36.8
RBMC	3e6	14.5

We fix the parameter $\mu = 1$. The batch and step sizes $p = 2$ and $m = 9$ are taken so that the SDE step is given by

$$(4.8) \qquad X^i \leftarrow X^i - \left[X^i + \nabla U_1(X^i - x_{\eta_k})\right]\tau + \sqrt{\frac{2\mu}{N-1}}\sqrt{\tau}z_k, \quad k = 0, \ldots, m-1,$$

where η_k is chosen from $\{1, 2, \ldots, N\} \setminus \{i\}$ with uniform probability. The time step is chosen as $\tau = 10^{-4}$ for all the experiments in this example. This small time step is needed largely due to the large magnitude of the gradient for U_1 in the $r < 0.01$ region.

As we have mentioned, there are two phases: the burn-in phase and the sampling phase. Table 1 shows the burn-in phase we pick for the two methods in one experiment. The first 3×10^5 sampling iterations in the MH method and the first 3×10^6 sampling iterations in the RBMC method are discarded, as the distribution is far away from the semicircle law. Though the number of iterations is larger, our proposed method costs only 14.5s compared to 36.8s in MH, implying that our method is more efficient.
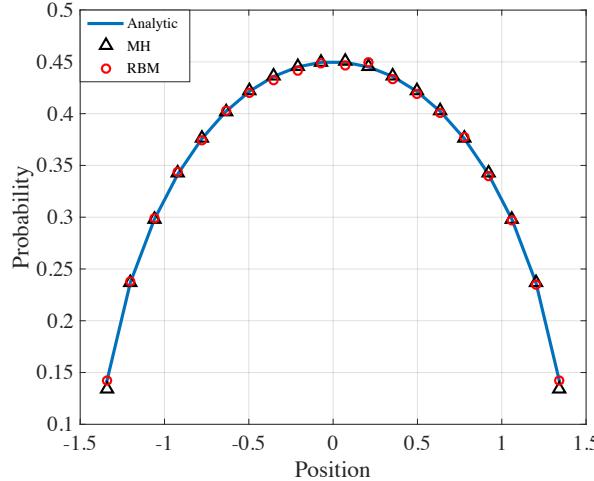

FIG. 1. *Empirical densities obtained by the two methods with $1e7$ sampling iterations ($1e7N$ sample points). The blue (color available online only) curve represents the semicircle law (4.4).*

In Figure 1, we show the performances of the two methods, with the samples collected from the sampling phase. The empirical densities are computed using bin counting. In particular, we divide the interval $[-\sqrt{2}, \sqrt{2}]$ into 20 equal pieces so that the width of each bin is $h = \sqrt{2}/10$. Define $x_i = -\sqrt{2} + ih$. Then, the density in the ith bin $[x_{i-1}, x_i)$ is approximated by $\rho_i \approx N_i/(Nh)$. The samples are collected from all the iterations after the burn-in phase (each iteration contributes $N = 500$ sample points). In particular, let N_i denote the number of particles in the bin $[x_{i-1}, x_i)$. In each iteration k, we do $N_i \leftarrow N_i + N_i^k$, where N_i^k is the number of points in the ith

TABLE 2
Sampling times and errors for two methods.

Iteration	1e5	2e5	5e5	1e6	2e6	5e6	1e7	2e7
MH time	20.5	36.3	88.2	188.3	314.9	750.1	1384.9	2952.7
RBMC time	2.4	2.5	3.8	4.6	13.2	30.7	62.0	126.5
MH error	0.035	0.017	0.0060	0.0038	0.0023	0.0016	0.0015	0.0014
RBMC error	0.012	0.011	0.0062	0.0051	0.0048	0.0031	0.0028	0.0022

bin at the kth iteration. At last, normalize N_i: $N_i = N_i / \sum N_j$. Clearly, Figure 1 shows that both methods capture the semicircle law (4.4).

To compare the performance of the algorithms in more detail, in Table 2, we compare the running times (spent in the sampling phase only) and relative errors of the two methods as the iteration goes on. Here, the iteration means the number of iterations in the sampling phase and the relative error is computed by the L^1 norm

$$E := \sum_i \left| \rho_i - h^{-1} \int_{x_{i-1}}^{x_i} \rho(x)\,dx \right| h \Big/ \int_{-\sqrt{2}}^{\sqrt{2}} \rho(x)\,dx = \sum_i \left| \rho_i - h^{-1} \int_{x_{i-1}}^{x_i} \rho(x)\,dx \right| h.$$

We also plot the running time versus the Monte Carlo errors in Figure 2. The data in time-error plots in our figure are collected at different iterations in a single experiment with fixed N. A more reasonable comparison would be based on the data collected from different N values, and the time should be for the distribution to reach the equilibrium. This will be done in our subsequent work for interactions with long ranges like the Coulomb interaction.

As can be seen in Table 2 and Figure 2, our proposed method is more efficient if the desired accuracy is not very high. The reason is that our proposed method takes only $O(1)$ time per iteration. The overall running time in the RBMC method is roughly $1/20$ of that in the MH method for obtaining the same number of samples. As can be seen in Figure 2, to achieve the same sampling error, the computation time needed in the RBMC method is like $1/10$ of that needed in the MH method.

4.2. Lennard-Jones fluids. In this subsection, we test our algorithm on the Lennard-Jones fluid, aiming to recover the equation of state, which has been studied in [18]. We compare the sampling performance of the standard MH algorithm and our proposed method.

For the Lennard-Jones fluids, we take the external potential $V \equiv 0$, and consider the regime for the molecular dynamics so that $w = 1$. Hence, the Gibbs measure is given by

$$\pi \propto \exp\left[-\beta \sum_{i,j:i<j} U(x_i - x_j) \right].$$

The potential for this model system is given by

$$(4.9) \qquad U(x_i - x_j) = u(r_{ij}) = 4\left[\left(\frac{1}{r_{ij}}\right)^{12} - \left(\frac{1}{r_{ij}}\right)^{6} \right],$$

where again $r_{ij} = |x_i - x_j|$ is the distance between three-dimensional locations x_i and x_j. For convenience, we will also define $\vec{r}_{ij} := x_i - x_j$.

In numerical simulations, the periodic boxes are often used to approximate the fluid. Assume the length of the periodic box is L. A given particle x_i interacts
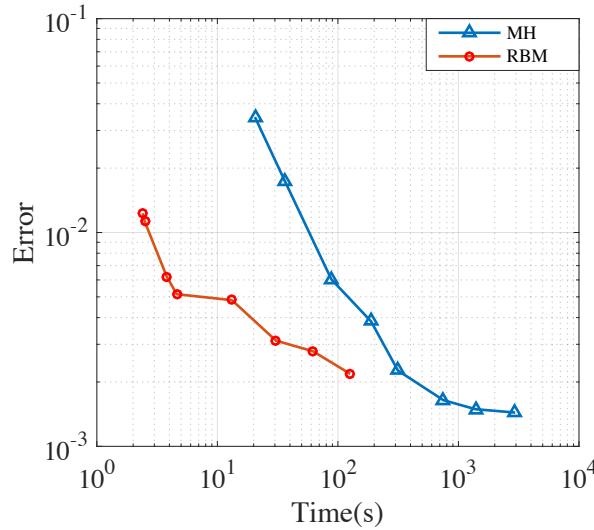
FIG. 2. *The relative error of the two methods decreases as time goes on. In the RBMC method, we only need* 10% *of the time of the MH method to get the error with the same level.*

with all other particles in the system and their periodic images. We pick a cutoff length $r_c = L/2$. Thanks to the fast decay property of the Lennard-Jones potential (4.9), for a fixed particle, one can approximate the summation away from r_c by a continuous integral, and the radial density function (see [8, Chap. 3]) is approximated by $\rho(r) \approx N/L^3$ when $r \geq r_c$.

With the approximation, the pressure formula is given by

$$(4.10) \quad P = \frac{\rho}{\beta} + \frac{8}{V} \sum_{i=1}^{N} \sum_{j:j>i,\tilde{r}_{ij}<r_c} (2\tilde{r}_{ij}^{-12} - \tilde{r}_{ij}^{-6}) + \frac{16}{3}\pi\rho^2 \left[\frac{2}{3}\left(\frac{1}{r_c}\right)^9 - \left(\frac{1}{r_c}\right)^3 \right],$$

and the acceptance rate in the MH algorithm (Algorithm 2) can be approximated by

$$(4.11) \quad \alpha = \min\left\{ 1, \exp\left[-\beta\left(\sum_{j:j\neq i,\tilde{r}_{ij}^*<r_c} u(\tilde{r}_{ij}^*) - \sum_{j:j\neq i,\tilde{r}_{ij}<r_c} u(\tilde{r}_{ij}) \right) \right] \right\}.$$

Here, we have introduced

$$\tilde{r}_{ij} = |\vec{r}_{ij} + \vec{n}L|$$

for some suitably three-dimensional integer vector $\vec{n}$ so that $|\vec{r}_{ij} + \vec{n}L|$ is minimized. In fact, since $r_c = L/2$, for each j, there is at most one image (including itself) that falls into $B(x_i, r_c)$, so the formula is fine. Clearly, $\tilde{r}_{ij}^*$ is similarly defined just with x_i replaced by x_i^*, the position of the candidate move. Regarding Algorithm 1, for a particle, under the same approximation, the net forces of all the particles/images away from r_c contribute net zero force to the current particle. Equivalently, we think that each of these images contributes zero force to the current particle. Hence, in the SDE step of Algorithm 1, only when the randomly selected particle/image (select $j \in \{1, \ldots, N\} \setminus \{i\}$ randomly and then determine the $\vec{n}$ vector such that $|x_i - x_j + \vec{n}L|$ is minimized) is within $B(X^i, r_c)$ will we compute its contribution and update X^i; otherwise, we do not update X^i but still increase k until it reaches m. For the rejection

TABLE 3
Burn-in phase. The first $5e4$ iterations in the MH method are discarded, which costs $182.6s$. The first $2e5$ iterations are discarded in the RBMC method, which costs $75s$.

	Iteration	Time (s)
MH	5e4	182.6
RBMC	2e5	75

step, since U_2 is short-ranged, we also only consider images within $B(X^i, r_c)$. We refer the readers to Appendix B for the details of these formulas.

Following [8], we take the number of particles $N = 500$ and the length of the periodic boxes will be determined by the desired density correspondingly:

$$(4.12) \qquad L = \left(\frac{N}{\rho}\right)^{1/3}.$$

For the MH method (Algorithm 2), the random displacement for a randomly selected particle obeys the $\mathcal{N}(0, \sigma^2 I_3)$ with $\sigma = 0.05 r_c$. In other words,

$$(4.13) \qquad x_i^* = x_i + \sigma z, \quad z \sim \mathcal{N}(0, I_3).$$

For our proposed algorithm (Algorithm 1), we do the splitting

$$(4.14) \qquad u(r) = u_1(r) + u_2(r),$$

where

$$(4.15) \qquad u_1(r) = \begin{cases} 2^{-1/3}(r - \sqrt[6]{2})^2 - 1, & 0 < r < \sqrt[6]{2}, \\ 4 \times (r^{-12} - r^{-6}), & r \geq \sqrt[6]{2}, \end{cases}$$

and

$$(4.16) \qquad u_2(r) = \begin{cases} 4 \times (r^{-12} - r^{-6}) - 2^{-1/3}(r - \sqrt[6]{2})^2 + 1, & 0 < r < \sqrt[6]{2}, \\ 0, & r \geq \sqrt[6]{2}. \end{cases}$$

In the SDE step, the batch size is chosen as $p = 2$, and we take $m = 9$ so that the iteration in the SDE step is given by

$$(4.17) \qquad X^i \leftarrow X^i - \nabla U_1(X^i - x_{\eta_k})\tau + \sqrt{\frac{2T}{N-1}}\sqrt{\tau}z_k, \quad k = 0, \ldots, m-1,$$

where η_k is chosen from $\{1, 2, \ldots, N\} \setminus \{i\}$ with uniform probability. The time step is chosen as $\tau = 0.01$ for all the experiments in this example.

Table 3 shows the burn-in phase we pick for the two methods in one experiment. Clearly, our method is again more efficient to achieve the thermal equilibriums. In Figure 3, we show the results for these two methods with samples collected from the sampling phase. Clearly, they both have good convergence and ergodicity performance. Figure 3(a) shows the results for $T = 2$, while Figure 3(b) shows the results for the lower temperature $T = 0.9$. The analytical solution is given by the fitting curves in [18].

For $T = 2$, the system stays vapor for all densities, while in the case of $T = 0.9$, there is fluid-vapor coexistence for $\rho < 0.75$ (when $\rho > 0.75$ the system should be in
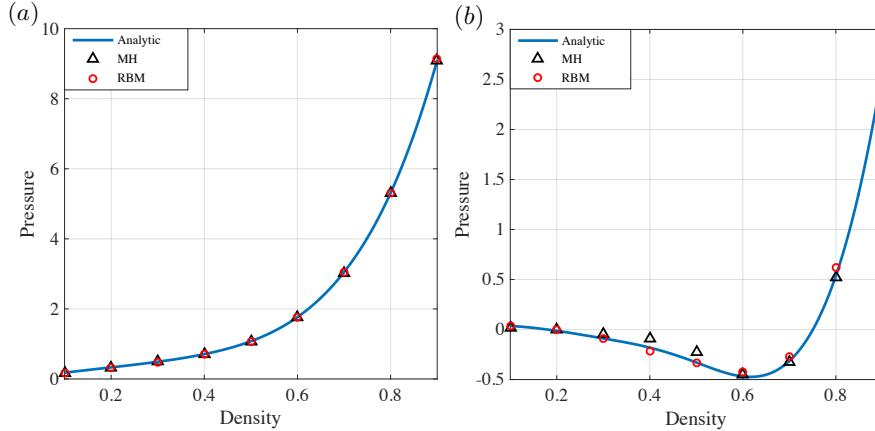
FIG. 3. *Average results of* $2e7$ *sampling iterations for the two methods compared with analytic results.* (a) $T = 2$; (b) $T = 0.9$.

the liquid state). According to the figure, our proposed algorithms behave similarly to the MH algorithm. They can capture the equation of state correctly. For $T = 2$, the result is quite good and the eventual relative error is like 0.5% and 0.2% relatively, as has been mentioned above. For $T = 0.9$, though the simulation result is not extremely great, the curves have been correctly captured. The relative errors are about 4%– 5%. The reason for the relatively poorer performance when the temperature is low ($T = 0.9$) is that the acceptance rates for both methods become smaller. When $\rho = 0.4, 0.5$, the MH method seems to be poorer. In fact, acceptance rates in the MH method are small for these ranges of densities. When $\rho > 0.7$ in the RBMC method, the new sampling points generated by the SDE step are more likely to be rejected by the short-range potential.
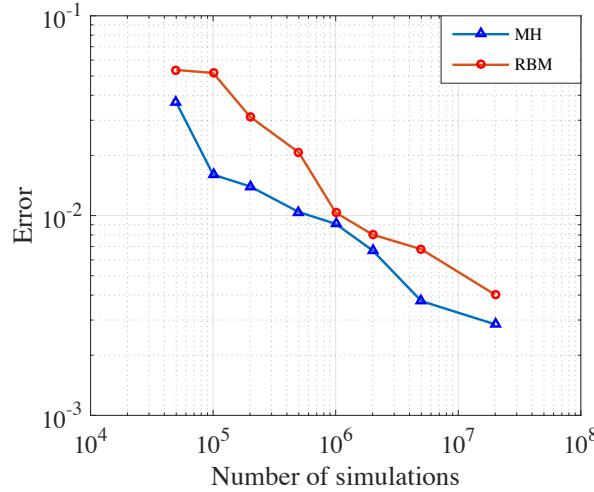

FIG. 4. $T = 2$. *Relative errors for the two methods decrease as the iteration goes on (and thus the number of sample points increases).*

In Figure 4, we show how the errors change as the iteration goes on in the sampling

Running times and errors for the two methods in the sampling phase. Compared to the MH method, the RBMC method needs twice the number of iterations to give comparable sampling errors, but the sampling time is much less, such as 2e6 iterations in the MH method and 5e6 iterations in the RBMC method.

Iteration	5e4	1e5	2e5	5e5	1e6	2e6	5e6	2e7
MH time	188.8	369.0	740.9	1861.0	3767.3	7528.6	18656.4	73557.4
RBMC time	40.5	82.5	137.3	283.9	533.2	1059.6	2463.0	9509.0
MH error	0.037	0.016	0.014	0.010	0.0090	0.0067	0.0037	0.0029
RBMC error	0.054	0.052	0.031	0.021	0.010	0.0080	0.0068	0.0040

phase for $T = 2$, where the relative error is calculated using the ℓ^2 norm as

$$(4.18) \qquad E = \sqrt{\sum_{i=1}^{9} \frac{1}{9}(P_i - P(\rho_i))^2} \Big/ \sqrt{\sum_{i=1}^{9} \frac{1}{9} P(\rho_i)^2}.$$

Here, P_i represents the numerical solution while $P(\cdot)$ is the analytical solution given by the fitting curves in [18]. The samples are also taken from the iterations after the burn-in: in each iteration, we compute the pressure using (4.10) and use all the pressures in the history up to the current iteration to calculate the empirical mean, which is used as the numerical pressure, even though some of them are rejected. The rejected points are necessary for satisfying a detailed balance condition. According to the picture, the MH method has smaller error than the RBMC method in the same number of simulations. In the long run, the relative error of the RBMC method is about 0.5%, but in the MH method it is about 0.2%. The error in the two methods decreases as $1/\sqrt{N}$, which satisfies the convergence rate in the Monte Carlo method.
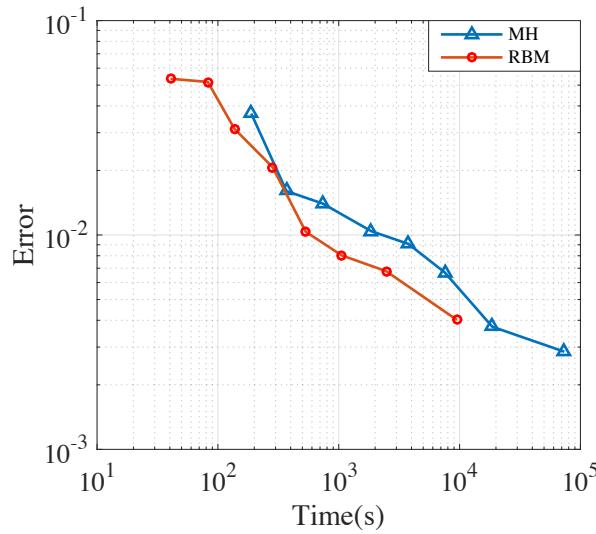

FIG. 5. *Relative error of densities in the L^1 norm. For the RBMC method, we only need 1/3 the time of the MH method to get the accuracy in the same level.*

To compare the efficiency of the two methods, we again present the relative errors and running times in Table 4 and plot the errors versus the running times in Figure 5. The time here is the time in the sampling phase only while the relative errors

are computed using the data collected from the sampling phase only. Here are some observations: to achieve the same sampling error, our proposed method needs roughly twice the number of iterations compared to the MH method but takes about 1/3 the time of the MH method. Since our method is an approximation method with a systematical error, when the number of iterations gets large (more sample points), the sampling error will not decrease to zero; hence our method is more efficient in the early stage.

It seems that the time saving in this example is not as significant as in the Dyson Brownian motion example. The reason is that in our implementation of the cell list, we check all of the 27 neighbor cells, which is not quite efficient, so our proposed method does not save time as significant as we desire. We expect that when N is larger, our algorithm can be more efficient. As mentioned above in section 4.1, a more reasonable comparison would be based on the data collected from different N values and the time should be computed for the distribution to reach the equilibrium. Since our aim in this paper is to propose an algorithm and we have seen it already owns some advantages in time efficiency, we choose not to do this careful comparison. The careful and detailed comparison of our algorithm to the MH algorithm will be performed in our subsequent work for the interactions with long ranges like the Coulomb interactions, which is more interesting. (After all, the Lennard-Jones potential still has a short range of interaction.) Besides, during our experiments, we observed that the acceptance rates of the RBMC method are like 75%–95% while the acceptance rates in the MH method are like 20%–50%, which also indicates that our proposed method is more efficient.

It is remarked that when $N = 500$, the speedup of our algorithm for the Lennard-Jones potential is comparable to the speedup made by MTS-MC [13]. Since our algorithm is $O(1)$ per iteration, we expect that the CPU time will increase linearly in N and thus the speedup could be larger for larger N. MTS-MC has roughly the same behavior, but we have theoretic analysis for our algorithm ($O(1)$ per move). One more thing, as we have pointed out, is that the data structure for the cell list is not optimized in our implementation, and the speedup can be increased if it is improved.

5. Conclusions. In summary, an RBMC method is developed for sampling from Gibbs distributions of interacting particle systems with singular kernels. The coupling of the random-batch and the splitting approaches in the RBMC leads us to a highly efficient sampling method. The splitting of the interaction kernels allows the use of the random batch to obtain a candidate sample with $O(1)$ operations and the acceptance-rejection strategy which avoids the long-term relaxation in the case of a particle moving to the repulsive region of a neighboring particle and ensures the detailed balance condition. Since the calculation of the acceptance ratio is only related to the interaction between a particle and its neighboring particles, the cost remains $O(1)$. The theoretical error estimate is present for the accuracy of the RBMC method.

Numerical examples of the Dyson Brownian motion and the Lennard-Jones fluids demonstrate that the RBMC method is promising in simulating particle systems. It is remarked that the Lennard-Jones potential itself is of short range in the sense that the $1/r^6$ kernel decays rapidly and the cutoff can be used. Nevertheless, our numerical results have already shown that the RBMC method is faster than the traditional MH sampling using the cutoff scheme. The RBMC method will be certainly more advantageous for particle systems with longer-range interaction, such as the Coulomb potential, which is essential in many areas of applications. This extension will be studied in the future.

Appendix A. Proof of Lemma 2.1. The first claim regarding invariant measure is trivial. Let us only verify the detailed balance formally. Consider the

general SDE

$$dX = b(X)\,dt + \sigma(X)dW,$$

where σ is a $d \times m$ matrix while W is an m-dimensional standard Wiener process.

The law of X (or the density of the law of X), $p(x,t)$, satisfies the Fokker–Planck equation [33]

$$(A.1) \qquad \partial_t p = -\nabla \cdot (bp) + \frac{1}{2} \sum_{i,j=1}^{d} \partial_{ij}(\Lambda_{ij}p) =: \mathcal{L}^* p,$$

where $\Lambda = \sigma\sigma^T$ and the operator

$$\mathcal{L}^* = -\nabla \cdot (b\cdot) + \frac{1}{2} \sum_{i,j=1}^{d} \partial_{ij}(\Lambda_{ij}\cdot)$$

is the dual operator of the generator of the SDE [30, Theorem 7.3.3] given by

$$\mathcal{L} := -b \cdot \nabla + \frac{1}{2}\Lambda : \nabla^2.$$

Let $p(\cdot, t; x_i)$ be the Green's function which is the solution of (A.1) with initial condition $p(x, 0; x_i) = \delta(x - x_i)$ and gives the transition density starting from x_i:

$$p(\cdot, t; x_i) = e^{t\mathcal{L}^*}\delta(\cdot - x_i).$$

The detailed balance condition is therefore the distributional identity

$$\pi_1(x_i)\mathcal{L}_z^*\delta(z - x_i) = \pi_1(z)\mathcal{L}_{x_i}^*\delta(x_i - z).$$

We pick a test function φ. It can be shown easily that

$$\langle \pi_1(x_i)\mathcal{L}^*\delta(\cdot - x_i), \varphi(\cdot) \rangle = \pi_1(x_i)\mathcal{L}\varphi(x_i)$$

and that

$$\langle \pi_1(z)\mathcal{L}_{x_i}^*\delta(x_i - \cdot), \varphi(\cdot) \rangle = \mathcal{L}_{x_i}^*(\varphi(x_i)\pi_1(x_i)).$$

Hence, to verify the detailed balance condition, one needs

$$(A.2) \qquad \pi_1(x_i)\mathcal{L}\varphi(x_i) = \mathcal{L}_{x_i}^*(\varphi(x_i)\pi_1(x_i)),$$

which is reduced to

$$-\pi_1 b + \frac{1}{2}\nabla \cdot \left(\sigma\sigma^T \pi_1\right) = 0.$$

This holds for b being a gradient field and σ being square with $\sigma \propto I$.

Appendix B. Pressure and energy formulas for the Lennard-Jones fluids. In computing the total energy or computing the rejection probability for the MH algorithms, one needs to compute the summation $\sum_{i,j} U(x_i - x_j)$. For this purpose, we may make use of the rapid decay for the interaction kernel so that we can pick a truncation distance r_c and approximate the summation for $r_{ij} \geq r_c$ by a continuous integral. More precisely,

$$(B.1) \qquad \frac{1}{2}\sum_{i,j:i\neq j} U(x_i - x_j) \approx \frac{1}{2}\sum_{i\neq j:r_{ij}<r_c} U(x_i - x_j) + \frac{N}{2}\int_{r_c}^{\infty} 4\pi r^2 \rho(r)u(r)\mathrm{d}r.$$

In fact, this formula is obtained by

$$\frac{1}{2}\sum_{i,j:i\neq j}U(x_i - x_j) = \frac{1}{2}\sum_{i\neq j:r_{ij}<r_c}U(x_i - x_j) + \frac{1}{2}\sum_{i\neq j:r_{ij}\geq r_c}U(x_i - x_j)$$

$$= \frac{1}{2}\sum_{i\neq j:r_{ij}<r_c}U(x_i - x_j) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j:j\neq i,r_{ij}\geq r_c}u(r_{ij}).$$

Due to symmetry, when the thermal fluctuation is small, $\sum_{j:j\neq i,r_{ij}\geq r_c}u(r_{ij})$ should be independent of i and can be approximated by the following integral using radial density $\rho(r)$:

$$\sum_{j:j\neq i,r_{ij}\geq r_c}u(r_{ij}) \approx \int_{r_c}^{\infty}u(r)4\pi r^2\rho(r)\,dr.$$

With the approximation of the radial density $\rho(r) \approx \rho = N/V$, $r > r_c$, the average tail contribution (per particle) u^{tail} of the interaction energy is given by

(B.2) $$u^{tail} = \frac{1}{2}\int_{r_c}^{\infty}4\pi r^2\rho u(r)\mathrm{d}r = \frac{8}{3}\pi\rho\left[\frac{1}{3}\left(\frac{1}{r_c}\right)^9 - \left(\frac{1}{r_c}\right)^3\right].$$

This tail contribution is independent of the particles so that the acceptance probability in the MH algorithm (Algorithm 2) can be approximated by

(B.3) $$\alpha = \min\left\{1, \exp\left[-\beta\left(\sum_{j:j\neq i,r_{ij}^*\leq r_c}U(x_i^* - x_j) - \sum_{j:j\neq i,r_{ij}\leq r_c}U(x_i - x_j)\right)\right]\right\}.$$

Apparently, r_{ij}^* means $|x_i^* - x_j|$.

The pressure is calculated by [8, sect. 3.4]

(B.4) $$P = \frac{\rho}{\beta} + \frac{vir}{V},$$

where the virial is defined by

(B.5) $$vir = \frac{1}{3}\sum_i\sum_{j>i}f(r_{ij})\cdot\vec{r}_{ij}.$$

The function $f(r_{ij})$ in (B.5) is the intermolecular force, i.e., the negative gradient of the Lennard-Jones potential, so that

$$vir = 8\sum_{i<j}(2r_{ij}^{-12} - r_{ij}^{-6}) \approx 8\sum_{i<j:r_{ij}\leq r_c}(2r_{ij}^{-12} - r_{ij}^{-6}) + 4N\int_{r_c}^{\infty}(2r^{-12} - r^{-6})4\pi r^2\rho\,dr.$$

Note that we split all summation items into $r_{ij} \geq r_c$ and $r_{ij} < r_c$. Because $f(r_{ij})\cdot\vec{r}_{ij}$ decreases rapidly when $r_{ij} \geq r_c$ and the particles far away can be approximated by an average radial density $\rho = N/V$, the integral approximation for the summation makes sense. Consequently, the pressure can be approximated by

(B.6) $$P = \frac{\rho}{\beta} + \frac{8}{V}\sum_i\sum_{j:j>i,|r_{ij}|<r_c}(2r_{ij}^{-12} - r_{ij}^{-6}) + \frac{16}{3}\pi\rho^2\left[\frac{2}{3}\left(\frac{1}{r_c}\right)^9 - \left(\frac{1}{r_c}\right)^3\right].$$

With the cutoff, for our proposed algorithm (Algorithm 1), the cutoff affects the SDE step in the sense that the particles away from r_c contribute net zero force to the current particle. As mentioned in the main text, only when the randomly selected particle is within $B(X^i, r_c)$ will we compute its force and update X^i; otherwise, we do not update X^i but still increase k until it reaches m.

Consider now the periodic box that approximates the fluid, with the length of the box being L. A given particle x_i interacts with all other particles in the system and their periodic images. In this case, the total potential energy is

$$(B.7) \qquad U_{tot} = \frac{1}{2} \sum_{i,j,\vec{n}}' u(|\vec{r}_{ij} + \vec{n}L|),$$

where $\vec{n}$ is a three-dimensional integer vector and the notation $\sum_{i,j,\vec{n}}'$ means that the case $\vec{n} = 0$ when $i = j$ are excluded. We pick the truncation length $r_c = L/2$ as commonly used in literature [8, Chap. 3]. The total energy is then approximated as

$$(B.8) \qquad U_{tot} \approx \frac{1}{2} \sum_{|\vec{r}_{ij}+\vec{n}L|<r_c}' u(|\vec{r}_{ij} + \vec{n}L|) + \frac{8N}{3}\pi\rho \left[\frac{1}{3}\left(\frac{1}{r_c}\right)^9 - \left(\frac{1}{r_c}\right)^3 \right].$$

The pressure and the acceptance rate (B.3) in the MH algorithm are similarly modified for periodic boxes. The modification for our proposed RBMC algorithm (Algorithm 1) is also similar. We do not repeat here.

REFERENCES

[1] M. P. ALLEN AND D. J. TILDESLEY, *Computer Simulation of Liquids*, Oxford University Press, 1987.
[2] J. E. BESAG, *Comments on "Representations of knowledge in complex systems" by U. Grenander and M.I. Miller*, J. Roy. Statist. Soc. Ser. B, 56 (1994), pp. 591–592.
[3] L. BOTTOU, *Online learning and stochastic approximations*, in On-Line Learning in Neural Networks, Cambridge University Press, 1998, pp. 9–42.
[4] S. BUBECK, *Convex optimization: Algorithms and complexity*, Found. Trends Mach. Learn., 8 (2015), pp. 231–357.
[5] B. DAI, N. HE, H. DAI, AND L. SONG, *Provable Bayesian inference via particle mirror descent*, in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, 2016, pp. 985–994.
[6] A. S. DALALYAN, *Theoretical guarantees for approximate sampling from smooth and log-concave densities*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 79 (2017), pp. 651–676.
[7] L. ERDOS AND H.-T. YAU, *Dynamical Approach to Random Matrix Theory*, Courant Lect. Notes Math. 28, Courant Institute of Mathematical Sciences/American Mathematical Society, 2017.
[8] D. FRENKEL AND B. SMIT, *Understanding Molecular Simulation: From Algorithms to Applications*, Vol. 1, Elsevier, 2001.
[9] D. GAMERMAN AND H. F. LOPES, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, Chapman and Hall/CRC, 2006.
[10] A. GEORGES, G. KOTLIAR, W. KRAUTH, AND M. J. ROZENBERG, *Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions*, Rev. Modern Phys., 68 (1996), pp. 13–125.
[11] W. R. GILKS, S. RICHARDSON, AND D. SPIEGELHALTER, *Markov Chain Monte Carlo in Practice*, Chapman and Hall/CRC, 1995.
[12] W. K. HASTINGS, *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*, Oxford University Press, 1970.
[13] B. HETENYI, K. BERNACKI, AND B. J. BERNE, *Multiple "time step" Monte Carlo*, J. Chem. Phys., 117 (2002), pp. 8203–8207.

[14] W. Hu, C. J. Li, L. Li, and J.-G. Liu, *On the diffusion approximation of nonconvex stochastic gradient descent*, Ann. Math. Sci. Appl., 4 (2019), pp. 3–32.

[15] J. N. Israelachvili, *Intermolecular and Surface Forces*, 3rd ed., Academic Press, 2010.

[16] P.-E. Jabin and Z. Wang, *Mean field limit for stochastic particle systems*, in Active Particles, Vol. 1, Springer, 2017, pp. 379–402.

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

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

[19] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Stoch. Model. Appl. Probab. 23, Springer Science & Business Media, 2013.

[20] J.-M. Lasry and P.-L. Lions, *Mean field games*, Jpn. J. Math., 2 (2007), pp. 229–260.

[21] L. Li, Y. Li, J.-G. Liu, Z. Liu, and J. Lu, *A stochastic version of Stein Variational Gradient Descent for efficient sampling*, Commun. Appl. Math. Comput. Sci., to appear.

[22] L. Li, J.-G. Liu, and P. Yu, *On the mean field limit for Brownian particles with Coulomb interaction in* 3*D*, J. Math. Phys., 60 (2019), 111501.

[23] E. M. Lifshitz and L. P. Pitaevskii, *Statistical Physics: Theory of the Condensed State*, Vol. 9, Elsevier, 2013.

[24] Q. Liu and D. Wang, *Stein variational gradient descent: A general purpose Bayesian inference algorithm*, in Advances In Neural Information Processing Systems, MIT Press, 2016, pp. 2378–2386.

[25] Y.-A. Ma, T. Chen, and E. Fox, *A complete recipe for stochastic gradient MCMC*, in Advances in Neural Information Processing Systems, MIT Press, 2015, pp. 2917–2925.

[26] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas, *A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion*, SIAM J. Sci. Comput., 34 (2012), pp. A1460–A1487, https://doi.org/10.1137/110845598.

[27] M. G. Martin, B. Chen, and J. I. Siepmann, *A novel Monte Carlo algorithm for polarizable force fields: Application to a fluctuating charge model for water*, J. Chem. Phys., 108 (1998), pp. 3383–3385.

[28] 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 (1953), pp. 1087–1092.

[29] G. N. Milstein and M. V. Tretyakov, *Stochastic Numerics for Mathematical Physics*, Springer Science & Business Media, 2013.

[30] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, 6th ed., Springer, 2003.

[31] M. Raginsky, A. Rakhlin, and M. Telgarsky, *Non-convex Learning via Stochastic Gradient Langevin Dynamics: A Nonasymptotic Analysis*, preprint, https://arxiv.org/abs/1702.03849, 2017.

[32] D. J. Rezende and S. Mohamed, *Variational Inference with Normalizing Flows*, preprint, https://arxiv.org/abs/1505.05770, 2015.

[33] H. Risken, *The Fokker-Planck Equation*, Springer, 1996.

[34] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statistics, 22 (1951), pp. 400–407.

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

[36] F. Santambrogio, *Optimal Transport for Applied Mathematicians*, Springer, 2015.

[37] H. E. Stanley, *Phase Transitions and Critical Phenomena*, Clarendon Press, 1971.

[38] T. Tao, *Topics in Random Matrix Theory*, Grad. Stud. Math. 132, American Mathematical Society, 2012.

[39] 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), 2011, pp. 681–688.