Efficient sampling of thermal averages of interacting quantum particle systems with random batches ⊘

Xuda Ye [iD] ; Zhennan Zhou ✉ [iD]

Check for updates

View Online

Export Citation

05 November 2025 21:36:51

Efficient sampling of thermal averages of interacting quantum particle systems with random batches

View Online   Export Citation   CrossMark

Xuda Ye[1,a] (iD) and Zhennan Zhou[2,b] (iD)

AFFILIATIONS

[1] School of Mathematical Sciences, Peking University, Beijing, China
[2] Beijing International Center for Mathematical Research, Peking University, Beijing, China

[a] Electronic mail: abneryepku@pku.edu.cn
[b] Author to whom correspondence should be addressed: zhennan@bicmr.pku.edu.cn

ABSTRACT

An efficient sampling method, the pmmLang + RBM, is proposed to compute the quantum thermal average in the interacting quantum particle system. Benefiting from the random batch method (RBM), the pmmLang + RBM has the potential to reduce the complexity due to interaction forces per time step from $O(NP^2)$ to $O(NP)$, where N is the number of beads and P is the number of particles. Although the RBM introduces a random perturbation of the interaction forces at each time step, the long time effects of the random perturbations along the sampling process only result in a small bias in the empirical measure of the pmmLang + RBM from the target distribution, which also implies a small error in the thermal average calculation. We numerically study the convergence of the pmmLang + RBM and quantitatively investigate the dependence of the error in computing the thermal average on the parameters such as batch size, time step, and so on. We also propose an extension of the pmmLang + RBM, which is based on the splitting Monte Carlo method and is applicable when the interacting potential contains a singular part.

Published under license by AIP Publishing. https://doi.org/10.1063/5.0047437

I. INTRODUCTION

Simulating complex chemical systems with quantum effects has been an appealing subject in computational physics and chemistry. In quantum systems, thermal properties are fully described by the canonical ensemble and a considerable number of methods of calculating thermal averages are based on the path integral representation,[1–4] which reformulates the quantum system as a classical ring polymer system. In the past decades, the path integral Monte Carlo (PIMC)[4–6] and the path integral molecular dynamics (PIMD)[7–9] techniques have been developed and successfully applied to the calculation of quantum properties such as reaction rates,[10–12] correlation functions,[13,14] and quantum tunneling.[15,16]

In this paper, we focus on long-range interacting quantum particle systems. The complexity to calculate the interaction forces of a P-particle system is $O(P^2)$, and efficient computational methods are thus needed to simulate such big systems. Compared to short-range interactions that can be easily treated with cutoff[17] or data structures such as the cell list,[18,19] the long-range nature of the interaction potential makes it more difficult to reduce the complexity due to interaction forces.

So far, a large variety of methods have been proposed to calculate the interaction forces for a particle system with electrostatic interactions. The most representative methods in this class are the Ewald summation[20,21] for problems with periodic boundaries and the fast multipole method (FMM)[22] for open systems, and the complexity of the interaction forces can be reduced to $O(P \log P)$ or even $O(P)$. However, an obvious drawback of these methods is that they rely on the specific expression of the interaction potential, i.e., the Coulomb potential, and can hardly be applied to more general interacting systems.

Besides the methods that aim at calculating the interaction forces directly, there are also methods focusing on modifying the dynamics while maintaining the physical properties of interest. For example, the generalized Verlet algorithm[23] reduces the number

of times of calculating interaction forces by introducing distance classes, and the random batch method (RBM)[24] simplifies the dynamics by random sampling at each iteration.

In this paper, the RBM is employed to reduce the complexity due to interaction forces. The RBM is an efficient sampling method recently introduced for interacting particle systems and has been applied to reduce the computational cost of certain examples in real time dynamics and ensemble average calculation.[25,26] In each time step, the RBM randomly groups all particles into small batches, and the evolution of the whole system is replaced by the evolution in those batches, respectively. Due to the random nature of the grouping in the RBM, the average force felt by each particle in the RBM is statistically the same as the force in the original dynamics. Since interaction forces are only calculated in small batches, the complexity in a time step is reduced from $O(P^2)$ to $O(P)$. Compared to other efficient sampling methods such as the Ewald summation and the FMM, the RBM is easier to implement and more flexible to apply in complex interacting particle systems.

After including quantum effects, the simulation of interacting particle systems meets additional challenges. In the context of the PIMD, suppose there are N beads in the reformulated ring polymer system. The complexity due to interaction forces in this reformulated system per time step is $O(NP^2)$, which is N times of the complexity in the classical case. In addition, the ring polymer system suffers from the stiffness of the ring polymer potential when the number of beads N is large. To resolve the stiffness, one can either precondition the dynamics by modifying the ring polymer mass matrix[27,28] or using the staging coordinates[29,30] or employ other non-preconditioned methods.[14,31–33]

In large interacting quantum systems, the compatibility between the RBM and the technique for resolving the stiffness must be taken into account. We choose to employ the preconditioned mass-modified Langevin dynamics (pmmLang)[28] rather than the staging coordinates to precondition the dynamics since the pmmLang sticks to the use of the physical coordinates.

In this paper, we will show that the combination of the pmmLang and the RBM (denoted by the pmmLang + RBM) is a suitable sampling method in the large interacting quantum particle systems. The pmmLang + RBM costs only the $O(NP)$ complexity in a time step to compute the interaction forces and is able to obtain the accurate thermal average up to a small bias. In addition, the pmmLang + RBM can be naturally combined with the splitting Monte Carlo method[25] to apply to systems with singular interacting potentials.

Now, we present our basic setup of this paper. In the P-particle quantum system in $\mathbb{R}^{3P}$ with long-range interactions, we aim to calculate the thermal average of some observable operator $\hat{A}$. Suppose the potential of the system is $\hat{V} = V(\hat{q})$, with $V(q)$ given by

$$V(q) = \sum_{i=1}^{P} V^{(o)}(q^i) + \sum_{1 \leqslant i < j \leqslant P} V^{(c)}(q^i - q^j), \quad (1)$$

where $q = (q^1, \ldots, q^P) \in \mathbb{R}^{3P}$ is the position and momentum of the P particles, $V^{(o)}(q^i)$ is the external potential of q^i in $\mathbb{R}^3$, and $V^{(c)}$ $(q^i - q^j)$ is the interacting potential between q^i and q^j. Assume each particle has mass m, then the Hamiltonian operator of the system is

$$\hat{H} = \frac{\hat{p}^2}{2m} + V(\hat{q}), \quad q, p \in \mathbb{R}^{3P}, \quad (2)$$

and the thermal average of $\hat{A}$ is given by

$$\langle \hat{A} \rangle = \frac{1}{Z} \mathrm{Tr}[e^{-\beta \hat{H}} \hat{A}], \quad Z = \mathrm{Tr}[e^{-\beta \hat{H}}], \quad (3)$$

where $\beta > 0$ is the inverse temperature.

In the path integral representation, the quantum system (1) is reformulated as a classical ring polymer system with the potential function

$$U_N(\boldsymbol{q}) = \frac{m}{2\beta_N^2} \sum_{k=1}^{N} |q_k - q_{k+1}|^2 + \sum_{k=1}^{N} V(q_k), \quad (4)$$

where $\beta_N = \beta/N$ and $\boldsymbol{q} = (q_1, \ldots, q_N)^\mathrm{T} \in \mathbb{R}^{N \times 3P}$ is the position of the ring polymer. In this way, the thermal average $\langle \hat{A} \rangle$ can be approximated by the ensemble average in the Boltzmann distribution $\pi(\boldsymbol{q}) \propto e^{-\beta_N U_N(\boldsymbol{q})}$. The pmmLang + RBM we propose in this paper, which aims to sample the distribution $\pi(\boldsymbol{q})$, can be implemented in the following steps:

1. Derive the pmmLang for the ring polymer system $U_N(\boldsymbol{q})$ to sample the Boltzmann distribution $\pi(\boldsymbol{q})$.
2. Numerically integrate the sampling path of the pmmLang, where the interaction forces are efficiently computed by the RBM. Time averages of the weight functions are used to approximate the thermal average $\langle \hat{A} \rangle$.
3. (Optional) If the interacting potential $V^{(c)}(q)$ contains a singular part, combine the pmmLang + RBM with the splitting Monte Carlo method.

The pmmLang + RBM is simple and efficient to calculate the thermal average of the interacting quantum particle systems and benefits from both the RBM and the pmmLang. Due to the RBM, the complexity of interaction forces in a time step is reduced to $O(NP)$, and the total complexity in a time step is $O(N \log NP)$, which has satisfactory scaling properties for both N and P. When the interaction potential contains a singular part (e.g., the Lennard-Jones and the Morse potential), the pmmLang + RBM method can be extended by the use of the splitting Monte Carlo method, which lifts the constraint of using extremely small time steps.

This paper is organized as follows: In Sec. II, we introduce the PIMD for interacting particle systems and the difficulties in numerical simulation and then derive the pmmLang. In Sec. III, we introduce the random batch method and discuss of the error analysis of the pmmLang + RBM. In Sec. IV, we introduce the splitting Monte Carlo method and its combination with the pmmLang + RBM and the pmmLang + RBM + split. In Sec. V, we present the numerical results of the pmmLang + RBM and the pmmLang + RBM + split and report the error in the calculation of thermal averages.

II. PIMD FOR INTERACTING PARTICLE SYSTEMS

A. Ring polymer representation and Langevin sampling

In the path integral representation, the thermal average (3) is approximated as

$$\langle \hat{A} \rangle \approx \frac{1}{Z_N} \int d\boldsymbol{q} \times e^{-\beta_N U_N(\boldsymbol{q})} \times W_N(\boldsymbol{q}), \tag{5}$$

where $\beta_N = \beta/N$ and

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix} = \begin{bmatrix} q_1^1 & \cdots & q_1^P \\ \vdots & \ddots & \vdots \\ q_N^1 & \cdots & q_N^P \end{bmatrix} \in \mathbb{R}^{N \times 3P}$$

is the position of the ring polymer. In the coordinate notation q_k^i, the subscript k indicates the bead index in the ring polymer representation and the superscript i indicates the particle index in the physical system. Besides, $U_N(\boldsymbol{q})$ is the total potential of the ring polymer system defined in (4),

$$Z_N = \int d\boldsymbol{q} \times e^{-\beta_N U_N(\boldsymbol{q})} \tag{6}$$

is the partition function, and $W_N(\boldsymbol{q})$ is the weight corresponding to the observable operator $\hat{A}$. The standard theory of the path integral[5] shows that the approximation (5) is exact as the number of beads $N \to \infty$. Define the Boltzmann distribution as

$$\pi(\boldsymbol{q}) = \frac{1}{Z_N} e^{-\beta_N U_N(\boldsymbol{q})}, \quad \boldsymbol{q} \in \mathbb{R}^{N \times 3P}, \tag{7}$$

and then (5) can be equivalently written as

$$\langle \hat{A} \rangle \approx \langle W_N(\boldsymbol{q}) \rangle_\pi := \int W_N(\boldsymbol{q}) \pi(\boldsymbol{q}) d\boldsymbol{q}, \tag{8}$$

i.e., $\langle \hat{A} \rangle$ is approximated as the ensemble average of the weight function $W_N(\boldsymbol{q})$ in the distribution $\pi(\boldsymbol{q})$.

In this paper, the observable operators of interest are the position-dependent operator $\hat{A} = A(\hat{q})$ and the kinetic energy operator $\hat{A} = \hat{p}^2/(2m)$, where $A(q)$ is an analytic function in $\mathbb{R}^{3P}$. For the position-dependent operator, the weight $W_N(\boldsymbol{q})$ is simply

$$W_N(\boldsymbol{q}) = \frac{1}{N} \sum_{k=1}^{N} A(q_k). \tag{9}$$

For the kinetic energy operator, the weight $W_N(\boldsymbol{q})$ is chosen as the virial estimator,[34]

$$W_N(\boldsymbol{q}) = \frac{3P}{2\beta} + \frac{1}{2N} \sum_{k=1}^{N} (q_k - \bar{q})^T \nabla V(q_k), \tag{10}$$

where

$$\bar{q} = \frac{1}{N} \sum_{k=1}^{N} q_k \in \mathbb{R}^{3P} \tag{11}$$

is the center of the ring polymer.

To compute the ensemble average $\langle W_N(\boldsymbol{q}) \rangle_\pi$, we employ the path integral molecular dynamics (PIMD) technique to sample the distribution $\pi(\boldsymbol{q})$. By introducing an auxiliary momentum variable

$\boldsymbol{p} \in \mathbb{R}^{N \times 3P}$, the PIMD couples the Hamiltonian dynamics with a thermostat scheme to preserve the invariant distribution. In practice, the widely used thermostats include the Andersen thermostat,[35] the Nosé–Hoover thermostat,[36] and the Langevin thermostat. In this paper, we focus on the last one to develop an efficient sampling algorithm with random batches and our underlying dynamics is thus the second-order Langevin dynamics.

For heuristic purposes, we present in the following the PIMD with the Langevin thermostat. Introduce the positive definite mass matrix $M \in \mathbb{R}^{N \times N}$ (maybe different from the physical mass m) and extend $U_N(\boldsymbol{q})$ to the Hamiltonian

$$H_N(\boldsymbol{q}, \boldsymbol{p}) = \frac{1}{2} \langle \boldsymbol{p}, M^{-1} \boldsymbol{p} \rangle_F + U_N(\boldsymbol{q}), \tag{12}$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product in $\mathbb{R}^{N \times 3P}$. By adding damping and diffusion terms in the Hamiltonian dynamics of (12), we obtain the second-order Langevin dynamics in the PIMD,

$$d\boldsymbol{q} = M^{-1} \boldsymbol{p} dt,$$

$$d\boldsymbol{p} = -\nabla U_N(\boldsymbol{q}) dt - \gamma \boldsymbol{p} dt + \sqrt{\frac{2\gamma M}{\beta_N}} d\boldsymbol{B}, \tag{13}$$

where $\gamma > 0$ is the friction constant and $\boldsymbol{B}$ is the standard Brownian motion in $\mathbb{R}^{N \times 3P}$.

The invariant distribution of (13) is

$$\pi(\boldsymbol{q}, \boldsymbol{p}) \propto \exp\left(-\beta_N \left(\frac{1}{2} \langle \boldsymbol{p}, M^{-1} \boldsymbol{p} \rangle_F + U_N(\boldsymbol{q})\right)\right), \tag{14}$$

whose marginal distribution in $\boldsymbol{q}$ is exactly $\pi(\boldsymbol{q})$ as in (7). From the ergodicity of the Langevin dynamics, $\langle W_N(\boldsymbol{q}) \rangle_\pi$ can be computed by the infinite time average of $W_N(\boldsymbol{q}(t))$ with the classical trajectory $(\boldsymbol{q}(t), \boldsymbol{p}(t))$ propagated by (13), i.e.,

$$\langle W_N(\boldsymbol{q}) \rangle_\pi = \lim_{T \to \infty} \frac{1}{T} \int_0^T W_N(\boldsymbol{q}(t)) dt. \tag{15}$$

In practice, one employs the discrete time trajectory $\boldsymbol{q}(j\Delta t)$ numerically solved by (13) to approximate the time integral in (15), i.e.,

$$\langle W_N(\boldsymbol{q}) \rangle_\pi = \lim_{J \to \infty} \frac{1}{J} \sum_{j=1}^{J} W_N(\boldsymbol{q}(j\Delta t)), \tag{16}$$

where Δt is the time step.

In the large interacting particle system, the costliest part of the Langevin dynamics (13) is computing the gradient $\nabla U_N(\boldsymbol{q})$, which involves heavy calculation of the interaction forces. As we shall elaborate in the next part, the complexity due to interaction forces is $O(NP^2)$.

B. Simulation bottleneck in Langevin sampling

In this paper, we are mainly concerned with the efficiency of calculating the thermal average (3) for a large quantum interacting particle system, where the numerical challenges mainly originate

05 November 2025 21:36:51

from two aspects of reasons. On the one hand, given a quantum interacting particle system, the number of beads N in the path integral representation needs to be sufficiently large to approximate the thermal average $\langle \hat{A} \rangle$. On the other hand, in order to accurately calculate a wide class of physical quantities, it is often desirable to simulate in large size systems, for example, the liquid water, since the number of particles P should be large enough to incorporate the correct scientific phenomenon.

When N and P are large, one of the major difficulties in the numerical simulation of (13) is that the computational cost per time step is extremely heavy. To perform a complexity analysis, we write the total $\nabla U_N(\boldsymbol{q})$ as

$$U_N(\boldsymbol{q}) = \frac{1}{2}\langle \boldsymbol{q}, L\boldsymbol{q}\rangle_F + \sum_{k=1}^{N} V(q_k), \qquad (17)$$

where $L \in \mathbb{R}^{N \times N}$ is the second order difference matrix,

$$L = \frac{m}{\beta_N^2}\begin{bmatrix} 2 & -1 & & \cdots & & -1 \\ -1 & 2 & -1 & \cdots & & \\ & -1 & 2 & \cdots & & \\ \vdots & \vdots & \vdots & \ddots & -1 & \\ & & & -1 & 2 & -1 \\ -1 & & & & -1 & 2 \end{bmatrix}. \qquad (18)$$

The gradient $\nabla U_N(\boldsymbol{q})$ is thus given by

$$\nabla U_N(\boldsymbol{q}) = L\boldsymbol{q} + \begin{bmatrix} \nabla V(q_1) \\ \vdots \\ \nabla V(q_N) \end{bmatrix} \in \mathbb{R}^{N \times 3P}, \qquad (19)$$

where each $\nabla V(q_k)$ involves the full interaction forces in system (1). In fact,

$$\nabla V(q_k) = \begin{bmatrix} \dfrac{\partial V(q_k)}{\partial q_k^1} & \cdots & \dfrac{\partial V(q_k)}{\partial q_k^P} \end{bmatrix} \in \mathbb{R}^{3P}, \qquad (20)$$

and the interaction force felt by the ith particle q_k^i is

$$\frac{\partial V(q_k)}{\partial q_k^i} = \sum_{i=1}^{P} \nabla V^{(o)}(q_k^i) + \sum_{j \neq i} \nabla V^{(c)}(q_k^i - q_k^j). \qquad (21)$$

Therefore, the complexity of calculating each $\nabla V(q_k)$ is $O(P^2)$ and the complexity of calculating the full gradient $\nabla U_N(\boldsymbol{q})$ is $O(NP^2)$.

It is worth noting that the $O(P^2)$ computational cost per time step is the common difficulty in the simulation of interacting particle systems, regardless of the thermostat scheme. As long as one uses molecular dynamics approaches, heavy calculation of the interaction forces is inevitable. In the PIMD where an N-bead ring polymer is used, there are N duplicates of the original interacting system, and the computational cost becomes $O(NP^2)$.

Except for the interaction forces $\nabla V^{(c)}(q_k^i - q_k^j)$, calculation of the weight $W_N(\boldsymbol{q})$ can also have $O(NP^2)$ time complexity in a single time step. For the position-dependent operator $\hat{A} = A(\hat{q})$, if the analytic function $A(q)$ is given by

$$A(q) = \frac{1}{P}\sum_{1 \leqslant i < j \leqslant P} V^{(c)}(q^i - q^j), \qquad (22)$$

then the complexity to compute the weight function (9) is $O(NP^2)$. In addition, the virial estimator (10) for the kinetic energy involves the full gradient $\{\nabla V(q_k)\}_{k=1}^{N}$, and thus, the complexity is $O(NP^2)$.

The $O(NP^2)$ complexity in the calculation of the interaction forces and the weight is a huge impediment on efficient sampling, especially for large P. Hence, we seek effective means to accelerate the simulation of the PIMD Langevin dynamics (13) by reducing the computational cost per time step to $O(NP)$. In the classical case, the Ewald summation[20] and FMM[22] have been successfully applied to reduce the complexity of interaction forces, but they are complicated to implement, cumbersome in the ring polymer representation, and only efficacious for specific interacting potentials, for example, the Coulomb potential. Therefore, we aim to develop a method to reduce the $O(NP^2)$ complexity, which is easy-to-use, consistent with the PIMD framework, and applicable for any interaction potential $V^{(c)}(q)$. For this reason, we propose a sampling algorithm motivated by the recently proposed random batch method (RBM).[24]

In the RBM, the group of P particles is randomly divided into small batches, where the division is chosen independently in different time steps. During one time step, the P particles are restricted to interact within their own batches, and in this way, the total complexity of interaction forces is reduced from $O(NP^2)$ to $O(NP)$. A more detailed description of incorporating the RBM in Langevin sampling is presented in Sec. III.

C. Choice of the preconditioning method

Before continuing the discussion on the RBM, we take a short detour to address another ubiquitous numerical issue in simulating the PIMD when N is large. The ring polymer potential $U_N(\boldsymbol{q})$ leads to a stiff term in the Langevin dynamics (13), which prevents the use of large time steps in integrating the sampling path. To get a glimpse of the stiffness, note that the condition number of the matrix $L \in \mathbb{R}^{N \times N}$ defined in (18) is

$$\text{cond}(L) := \frac{\lambda_{\max}(L)}{\lambda_{\min}(L)} = \left(\sin\frac{\pi}{N}\right)^{-1}. \qquad (23)$$

Hence, if the mass matrix is simply chosen as

$$M = mI, \qquad (24)$$

the time step Δt needs to be small as $O(1/N)$ to integrate the highest oscillation mode of (13), which means the Langevin dynamics (13) shows stiffness when N is large.

The numerical stiffness can be resolved by certain preconditioning methods, which is often realized by introducing a proper

change of coordinates or choosing a proper mass matrix. For example, the staging coordinate transformation[29] given by

$$\tilde{q}_1 = q_1, \quad \tilde{q}_k = q_k - \frac{(k-1)q_{k+1} + q_1}{k}, \quad k = 2, \ldots, N, \quad (25)$$

has shown to be a powerful technique in relaxing the use of time steps. However, the staging coordinates are not compatible with the RBM unless one repeatedly uses the staging coordinate transformation and its inverse (see Appendix A for a detailed discussion), which are inconvenient for large systems. In fact, a preconditioning method that can be directly implemented in the physical coordinates is preferable to be combined with the RBM.

For this reason, we adopt the preconditioned mass-modified Langevin dynamics (pmmLang)[28] as the preconditioning method, which we present in the following. In the Langevin dynamics (13), we choose the mass matrix in such a specific form

$$M = L^\alpha := L + \alpha I, \quad (26)$$

where the regularization parameter $\alpha > 0$ is introduced to make sure L^α is positive definite. Let

$$\boldsymbol{v} = M^{-1}\boldsymbol{p} = (L^\alpha)^{-1}\boldsymbol{p} \in \mathbb{R}^{N \times 3P} \quad (27)$$

be the velocity, and define the modified potential

$$U^\alpha(\boldsymbol{q}) = \sum_{k=1}^N V(q_k) - \frac{\alpha}{2}|\boldsymbol{q}|^2, \quad (28)$$

and then (13) becomes

$$d\boldsymbol{q} = \boldsymbol{v}dt, \quad (29)$$

$$d\boldsymbol{v} = -\boldsymbol{q}dt - (L^\alpha)^{-1}\nabla U^\alpha(\boldsymbol{q})dt - \gamma\boldsymbol{v}dt + \sqrt{\frac{2\gamma(L^\alpha)^{-1}}{\beta_N}}d\boldsymbol{B}.$$

The obtained Langevin dynamics (29) is exactly the pmmLang we aim to derive. Compared to the staging coordinates, the pmmLang can be conveniently implemented in the physical coordinates and is naturally compatible with the RBM. A detailed derivation of the pmmLang and the discussion on its implementation is given in Appendix A.

Given the regularity parameter $\alpha > 0$, the invariant distribution of the pmmLang (29) is

$$\pi(\boldsymbol{q}, \boldsymbol{v}) \propto \exp\left(-\beta_N\left(\frac{1}{2}\langle\boldsymbol{v}, L^\alpha\boldsymbol{v}\rangle_F + \frac{1}{2}\langle\boldsymbol{q}, L^\alpha\boldsymbol{q}\rangle_F + U^\alpha(\boldsymbol{q})\right)\right). \quad (30)$$

Since the right-hand sides of (14) and (30) are exactly the same except for the velocity transformation $v = (L^\alpha)^{-1}\boldsymbol{p}$, the pmmLang (29) produces the correct invariant distribution $(L^\alpha)^{-1}\pi(\boldsymbol{q})$ in the marginal of $\boldsymbol{q}$. If we could neglect the $\nabla U^\alpha(\boldsymbol{q})$ term in (29), the Hamiltonian part of the pmmLang is simply

$$d\boldsymbol{q} = \boldsymbol{v}dt, \quad (31)$$
$$d\boldsymbol{v} = -\boldsymbol{q}dt,$$

and thus, there is no stiffness in (29). With further analysis, the pmmLang (29) can be shown to be a successful preconditioning method[28] in the PIMD.

It is worth pointing out that the pmmLang has also been studied in some recent work,[37,38] where the authors obtained a quantitative analysis of the convergence rate with its dependence on the regularization constant α. A qualitative corollary of their result is that α being close to zero or too large, both slowdowns the simulation efficiency of the pmmLang. From our point of view, if α is close to zero, the entries of the matrix $(L^\alpha)^{-1}$ in (29) will be large as $O(1/\alpha)$; hence, the magnitude of $(L^\alpha)^{-1}\nabla U_N(\boldsymbol{q})$ restricts the time step for numerical stability. For mathematical well-posedness, α should be chosen to ensure that the modified potential $U^\alpha(\boldsymbol{q})$ is confined, i.e.,

$$\lim_{|q| \to \infty} U^\alpha(\boldsymbol{q}) = +\infty. \quad (32)$$

While it is not a trivial task to choose the optimal parameter α in the pmmLang, there are other non-preconditioned numerical integrators for the PIMD Langevin dynamics, which are parameter free.[14,33] Nevertheless, how to choose α is not the focus of this work, and we employ the pmmLang as a preconditioning method in the PIMD under certain simplified conditions given below.

We assume that the external potential $V^{(o)}(q)$ in the quantum system (1) is harmonic, i.e., for some constant $\alpha_0 > 0$,

$$V^{(o)}(q) = \frac{\alpha_0}{2}|q|^2, \quad q \in \mathbb{R}^3. \quad (33)$$

For example, the value of α_0 may be specified according to (72), which involves the number of particles P in the interacting system. In this case, we choose the regularization parameter α to coincide with the constant α_0 in (33); thus, the modified potential $U^\alpha(\boldsymbol{q})$ is simply the sum of all interacting potentials,

$$U^\alpha(\boldsymbol{q}) = \sum_{k=1}^N \sum_{1 \leq i < j \leq N} V^{(c)}(q_k^i - q_k^j), \quad (34)$$

and the pmmLang (29) can be equivalently written as

$$d\boldsymbol{q}^i = \boldsymbol{v}^i dt, \quad (35)$$

$$d\boldsymbol{v}^i = -\boldsymbol{q}^i dt - (L^\alpha)^{-1}\sum_{j \neq i}\nabla V^{(c)}(\boldsymbol{q}^i - \boldsymbol{q}^j)dt - \gamma\boldsymbol{v}^i dt$$

$$+ \sqrt{\frac{2\gamma(L^\alpha)^{-1}}{\beta_N}}d\boldsymbol{B}^i \quad (i = 1, \ldots, P).$$

For simplicity, the incorporation of the RBM will be discussed only for (35), under the assumption of (33). Clearly, the RBM can be applied without (33), and there will be additional terms in $U^\alpha(\boldsymbol{q})$ except the interacting potentials in (34). These additional terms contribute only the $O(NP)$ complexity in a single time step and does not change the major difficulties since the $O(NP^2)$ complexity is only due to interaction forces $\nabla V^{(c)}(q_k^i - q_k^j)$.

05 November 2025 21:36:51

III. RANDOM BATCH METHOD IN LANGEVIN SAMPLING

A. RBM for the pmmLang

In the simulation of large interacting particle systems, the high computational cost per time step has always been an impediment to efficient sampling. With the setup in Sec. II, to integrate the sampling trajectory for a single time step, it requires the cost of $O(NP^2)$ complexity to compute all the interaction forces $V^{(c)}(q^i - q^j)$. Since the calculation of the interaction forces is rather expensive when P is large, reducing the complexity due to interaction forces is crucial to attain high efficiency in the PIMD simulation of interacting particle systems.

The recently proposed random batch method (RBM) provides a simple approach to fulfill such a reduction of cost in force evaluation. The RBM avoids calculation of the full interaction by randomly dividing the group of P particles into small-size batches and only allowing interactions within each small batch with adjusted interaction strength. Since for each time step, a new random division is conducted, after sufficiently many times, each particle will not only have an equal probability to interact with any other particles in the group, but also will have plenty of random interactions with the rest of the particles. This is why we expect the RBM to produce the correct statistical properties of the interacting system in the long time simulation. One does not necessarily need to trace the interactions between a typical particle and every other particle to have an accurate prediction of the statistical properties, but rather, the interactions with other particles may vastly cancel each other and result in an average field to the typical particle. For heuristic purpose, we present the RBM for the pmmLang (35) in the following.

During each time step, we first randomly divide the group of P particles into n batches $\mathscr{C}_1, \ldots, \mathscr{C}_q$, where each batch is of size p and $n = P/p$. The batch size p should be far less than P to avoid massive calculation of interaction forces and be at least two to capture the pairwise interactions $\nabla V^{(c)}(q^i - q^j)$. Then, we approximate the full interaction of the P-particle system with the interaction forces within each small batch $\mathscr{C}_l$, and adjust the interaction strength to ensure statistical consistency. To be specific, the interaction force felt by the ith particle $q^i \in \mathbb{R}^{N \times 3P}$ in the original pmmLang (35) is

$$\sum_{j \neq i} \nabla V^{(c)}(q^i - q^j) \in \mathbb{R}^{N \times 3}. \qquad (36)$$

With the use of RBM, each q^i is assigned to a batch $\mathscr{C}_l$ for some $l = 1, \ldots, q$, and hence, the approximation of the interaction force (36) within $\mathscr{C}_l$ is

$$\frac{P-1}{p-1} \sum_{j \in \mathscr{C}_l, j \neq i} \nabla V^{(c)}(q^i - q^j). \qquad (37)$$

Recall that the division $\mathscr{C}_1, \ldots, \mathscr{C}_q$ is randomly generated for each time step; thus, for fixed q^i, the batch $\mathscr{C}_l$ that contains q^i is also random. Therefore, (37) can be interpreted as a $(p-1)$-term sample approximation of a $(P-1)$-term summation (36). In addition, the approximation (37) is unbiased, i.e.,

$$\mathbb{E}\left(\frac{P-1}{p-1} \sum_{j \in \mathscr{C}_l, j \neq i} \nabla V^{(c)}(q^i - q^j)\right) = \sum_{j \neq i} \nabla V^{(c)}(q^i - q^j). \qquad (38)$$

In this way, the random-batch approximated pmmLang within the batch $\mathscr{C}_l$ is given by

$$dq^i = v^i dt, \qquad (39)$$

$$dv^i = -q^i dt - \frac{P-1}{p-1}(L^\alpha)^{-1} \sum_{j \in \mathscr{C}_l, j \neq i} \nabla V^{(c)}(q^i - q^j) dt - \gamma v^i dt$$

$$+ \sqrt{\frac{2\gamma(L^\alpha)^{-1}}{\beta_N}} dB^i \qquad (i \in \mathscr{C}_l),$$

and the RBM for the pmmLang (denoted by the pmmLang + RBM) in a single time step Δt just evolves (39) for all batches $\mathscr{C}_l$, $l = 1, \ldots, n$, as shown in Algorithm 1. In consecutive time steps, the previous divisions are discarded and new sets of divisions are randomly chosen. Hence, a fixed particle $q^i \in \mathbb{R}^{N \times 3}$ can be assigned to different batches $\mathscr{C}_l$ in different time steps and only interacts with the particles in the current batch.

In Algorithm 1, to generate a random division $\mathscr{C}_1, \ldots, \mathscr{C}_q$, one may use the Fisher–Yates shuffle[39] with $O(P)$ complexity to obtain a random permutation of $1, \ldots, P$ and then divide it into n batches in order. Note that there are $\frac{p(p-1)}{2}$ pairs of interactions within each batch $\mathscr{C}_l$, the complexity of Algorithm 35 due to interaction forces is

$$n \times N \times \frac{p(p-1)}{2} = N \times \frac{P(p-1)}{2} = O(NPp). \qquad (40)$$

Hence, the batch size p should be small to attain high efficiency in the pmmLang + RBM. In particular, if p is chosen as small integers such as 2 and 4, the complexity due to interaction forces is only $O(NP)$. In this way, we can employ the pmmLang + RBM (Algorithm 1), which is much more efficient than the original pmmLang (35), to compute the ensemble average $\langle W_N(q)\rangle_\pi$.

Finally, we point out that the RBM has an analogy to the stochastic gradient descent (SGD),[40–43] the stochastic optimization method widely used in machine learning. In machine learning problems, the size of the dataset can be extremely large, and the SGD approximates the full gradient with the average calculated from samples in a randomly chosen small batch. Although the RBM and the SGD share the idea of full gradient approximation, the high dimensional natures of their underlying problems are different. In an optimization problem where the SGD applies, the dimension of the parameter space does not increase with respect to the size of the dataset. However, in an interacting particle system where the RBM applies, the dimension of the coordinate space grows linearly with the number of particles P.

B. Error analysis in PIMD with the RBM

In this section, we aim to investigate the error in calculating the ensemble average $\langle W_N(q)\rangle_\pi$ due to the use of the RBM. By the ergodicity of the pmmLang (35), $\langle W_N(q)\rangle_\pi$ can be accurately computed from the time average,

Algorithm 1. RBM for the pmmLang in a time step Δt.

Randomly divide the P particles into n batches $\mathscr{C}_1, \cdots, \mathscr{C}_n$ of size p, where $n = P/p$.

for $l = 1, \cdots, n$ **do**

    Evolve the pmmLang within the batch $\mathscr{C}_l$ in a timestep Δt:

$$d\boldsymbol{q}^i = \boldsymbol{v}^i dt,$$

$$d\boldsymbol{v}^i = -\boldsymbol{q}^i dt - \frac{P-1}{p-1}(L^\alpha)^{-1} \sum_{j \in \mathscr{C}_l, j \neq i} \nabla V^{(c)}(\boldsymbol{q}^i - \boldsymbol{q}^j)dt - \gamma \boldsymbol{v}^i dt + \sqrt{\frac{2\gamma(L^\alpha)^{-1}}{\beta_N}}d\boldsymbol{B}. \qquad (i \in \mathscr{C}_l)$$

end

$$\frac{1}{J}\sum_{j=1}^{J} W_N(\boldsymbol{q}(j\Delta t)), \qquad (41)$$

as the number of samples $J \to \infty$, where $\{\boldsymbol{q}(t)\}_{t \geq 0}$ is the pmmLang trajectory evolved by (35) and Δt is the time step specified in Algorithm 1. Correspondingly, the pmmLang + RBM uses the following time average to estimate $\langle W_N(\boldsymbol{q})\rangle_\pi$:

$$\frac{1}{J}\sum_{j=1}^{J} W_N(\tilde{\boldsymbol{q}}(j\Delta t)), \qquad (42)$$

where $\{\tilde{\boldsymbol{q}}(t)\}_{t \geq 0}$ is the pmmLang + RBM trajectory evolved by Algorithm 1. Therefore, the error analysis in the RBM is actually asking whether the time average (42) is a good estimator of $\langle W_N(\boldsymbol{q})\rangle_\pi$ for sufficiently large J.

In general, exploring the effects of the consecutive uses of random-batch approximations (37) along the sampling process is difficult. To our knowledge, the strong error in a contracting dynamics[24] and the weak error in a short period[44] have been rigorously justified, whereas none of those theoretical results apply to the RBM in the long time sampling process.

If the adoption of the RBM results in a small strong error in the long time simulation, i.e., the pmmLang + RBM trajectory $\tilde{\boldsymbol{q}}(t)$ almost coincides with the pmmLang trajectory $\boldsymbol{q}(t)$ all the time, we can certainly deduce that the RBM also leads to a small error in the ensemble average calculation. However, this is not true. As we shall show in the following, it is likely that the pmmLang + RBM trajectory $\tilde{\boldsymbol{q}}(t)$ drifts significantly apart from $\boldsymbol{q}(t)$ in a short time, but still $\tilde{\boldsymbol{q}}(t)$ provides a faithful and accurate approximation of $\langle W_N(\boldsymbol{q})\rangle_\pi$ in the long time simulation. We shall further rationalize the error induced by the use of the RBM in the PIMD in the following.

1. Strong error analysis

The strong error characterizes the deviation of the numerically computed trajectory from the exact one in the form of the mean squared error.[45] The strong error of the pmmLang + RBM is then defined as

$$e(t) = \sqrt{\mathbb{E}|\tilde{\boldsymbol{q}}(t) - \boldsymbol{q}(t)|^2}, \quad t \geq 0, \qquad (43)$$

where we assume that the pmmLang + RBM trajectory $\tilde{\boldsymbol{q}}(t)$ and the pmmLang trajectory $\boldsymbol{q}(t)$ are driven by the same Brownian motion $\boldsymbol{B}(t)$ and the same initial state. In addition, the expectation is taken over all possible Brownian motions and choices of random batches in the time interval $[0, t]$. When (35) and (39) are integrated exactly, the deviation of $\tilde{\boldsymbol{q}}(t)$ from $\boldsymbol{q}(t)$ is solely due to the use of the random batches.

According to Algorithm 1, the pmmLang + RBM trajectory $\tilde{\boldsymbol{q}}(t)$ coincides with $\boldsymbol{q}(t)$ when the batch size $p = P$ or in the limit $\Delta t \to 0$. In fact, the batch size $p = P$ implies that the interaction forces are calculated accurately, while in the limit $\Delta t \to 0$ each particle is supposed to be driven by the average effects. Nevertheless, such parameters are impractical to apply in the pmmLang + RBM because choosing either $p = P$ or $\Delta t \to 0$ greatly increases the total computation cost and contradicts with our original intention to use the RBM. To be specific, the cost due to interaction forces of the pmmLang + RBM in the time interval $[0, T]$ is

$$O\left(\frac{p}{\Delta t} T P^2\right),$$

which is extremely large when $p = P$ or $\Delta t \to 0$.

When the parameters $p, \Delta t$ are chosen such that the pmmLang + RBM becomes an efficient sampling method, i.e., the batch size p is small and the time step Δt is relatively large, we cannot expect the pmmLang + RBM to produce the accurate pmmLang trajectory $\boldsymbol{q}(t)$. In contrast, the strong error $e(t)$ shortly grows large in a short time period, as we show in the following example.

In the Coulomb interacting system [the potential function is defined in (68) in Sec. V], we show in Fig. 1 how the pmmLang + RBM trajectory $\tilde{\boldsymbol{q}}(t)$ deviates from $\boldsymbol{q}(t)$ and the strong error $e(t)$ grows large along the sampling process. It can be seen from Fig. 1 that the pmmLang + RBM trajectory $\tilde{\boldsymbol{q}}(t)$ drifts apart from $\boldsymbol{q}(t)$ at about $t \approx 5$, and the strong error $e(t)$ grows large at $t \approx 1$, no matter how small the time step Δt is. Therefore, there will always be large deviation in the pmmLang + RBM trajectory $\tilde{\boldsymbol{q}}(t)$ from the pmmLang trajectory $\boldsymbol{q}(t)$, unless one chooses $p = P$ or $\Delta t \to 0$. As we shall show next, requiring a small path-wise error is totally unnecessary in thermal average calculation.

(a)

(b)
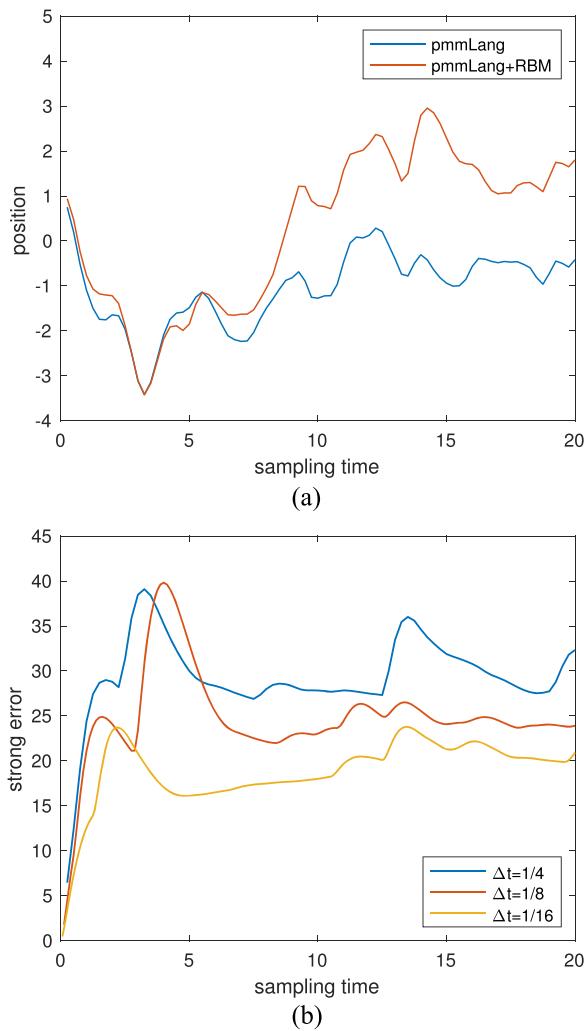
FIG. 1. (a) Typical samples of the pmmLang trajectory $q(t)$ and the pmmLang + RBM trajectory $\tilde{q}(t)$ with the time step $\Delta t = 1/4$. Only the first component of $q \in \mathbb{R}^{N \times 3P}$ is plotted. (b) Strong error $e(t)$ defined in (43) with $\Delta t = 1/4$, $1/8, 1/16$. 10 000 independent trajectories are used to compute the expectations. In the Coulomb interacting system, $m = 1$, $\beta = 4$, $P = 8$, $N = 32$, $T = 20$, and $p = 2$.

2. Weak error analysis

The weak error measures the difference in the approximate and the exact values of the ensemble average.[45] In the PIMD, the weight function of interest is $W_N(q)$, which takes different forms for different observables. To be specific, the weak error of the pmmLang + RBM is defined as

$$e(t) = \mathbb{E}W_N(\tilde{q}(t)) - \mathbb{E}W_N(q(t)), \qquad (44)$$

where the expectation is taken over all possible Brownian motions and choices of random batches, where the observable operator is the kinetic energy, and the corresponding weight function of interest is
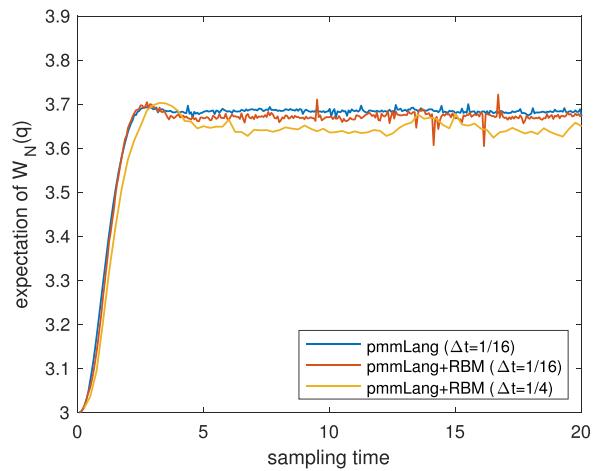

FIG. 2. Expectations $\mathbb{E}W_N(q(t))$ and $\mathbb{E}W_N(\tilde{q}(t))$ computed by the pmmLang and the pmmLang + RBM in the Coulomb interacting system, where the observable operator is the kinetic energy. $m = 1$, $\beta = 4$, $P = 8$, $N = 32$, $\Delta t = 1/4, 1/16$, $T = 20$, and $p = 2$. 10 000 independent trajectories are used to compute the expectations. The blue curve is associated with the pmmLang with $\Delta t = 1/16$, and the red and yellow curves are associated with the pmmLang + RBM with $\Delta t = 1/16$ and $1/4$, respectively. The difference between the curves of the pmmLang and the pmmLang + RBM is exactly the weak error $e(t)$ defined in (44).

the virial estimator (10). To give a first impression of the weak error, we plot in Fig. 2 the expectations $\mathbb{E}W_N(q(t))$ and $\mathbb{E}W_N(\tilde{q}(t))$ with various time steps in the Coulomb interacting system. Unlike the strong error (43), the curve of $\mathbb{E}W_N(\tilde{q}(t))$ associated with the pmm-Lang + RBM is very close to the curve of $\mathbb{E}W_N(q(t))$ associated with the pmmLang, and the weak error (44) remains small all along the sampling process. It can be seen from Fig. 2 that the ring polymer system quickly goes into equilibrium about $t \approx 2.5$ for both the pmmLang and the pmmLang + RBM dynamics; hence, the introduction of the batch force approximation (37) does not influence the convergence mechanism of the original pmmLang (35). In addition, the stochastic error in computing $W_N(q(t))$ due to the use of the force approximation (37) cancels over all possible choices of random batches, which finally results in a small bias on the average $\mathbb{E}W_N(q(t))$, even if the batch size p is small and the time step Δt is relatively large.

However, in the long time simulation, it is impossible to use $\mathbb{E}W_N(q(t))$ or $\mathbb{E}W_N(\tilde{q}(t))$ to estimate $\langle W_N(q) \rangle_\pi$ since the calculation of the expectations in (44) requires a large number of sampling trajectories. In practice, it is more feasible to use the time averages [(41) and (42)] to compute $\langle W_N(q) \rangle_\pi$, which are obtained by numerically integrating one sampling path, respectively.

If the numerical error is neglected, we know that due to the ergodicity of the Langevin dynamics, the time average (41) generated by the pmmLang converges to the correct limit as the number of samples $J \to \infty$, i.e.,

$$\lim_{J \to \infty} \frac{1}{J} \sum_{j=1}^{J} W_N(q(j\Delta t)) = \langle W_N(q) \rangle_\pi. \qquad (45)$$

We can actually view such a convergence from the perspective of the relative entropy. Recall that we aim to sample the Boltzmann

05 November 2025 21:36:51

distribution $\pi(\boldsymbol{q})$ as in (7), and for any distribution $f(\boldsymbol{q})$, we can use the relative entropy[46,47]

$$D(f\|\pi) = \mathbb{E}^f\left(\log\frac{f}{\pi}\right) \qquad (46)$$

to measure how much the distribution f deviates from the target distribution π. The relative entropy is always non-negative, and it is zero only when the two distributions are identical. Note that we can define a family of empirical distributions from the samples generated by the pmmLang, which are given by

$$\mu_J(\boldsymbol{q}) = \frac{1}{J}\sum_{j=1}^{J}\delta(\boldsymbol{q} - \boldsymbol{q}(j\Delta t)). \qquad (47)$$

Thus, the convergence (45) can be interpreted as

$$\lim_{J\to\infty} D(\mu_J\|\pi) = 0. \qquad (48)$$

The diminishing of the relative entropy manifests the time irreversibility and the ergodicity of the sampling trajectory. That is, the thermostat effect of the Langevin dynamics brings in dissipation to the relative entropy with respect to the invariant measure such that the empirical measure $\mu_J(\boldsymbol{q})$ converges to the target Boltzmann distribution $\pi(\boldsymbol{q})$ in the weak sense.

However, when the random batches are used along a sampling path, the empirical distributions are repeated perturbed, mostly likely away from the Boltzmann distribution. In fact, when the random batches are used for each time, although the force can be viewed as an unbiased approximation from the total interaction force, the sampling path deviates from the Langevin dynamics without the use of random batches with a deterministic bias for each specific choice of the random divisions. Let us denote the empirical measure of the pmmLang + RBM by $\tilde{\mu}_J(\boldsymbol{q})$, which is defined by

$$\tilde{\mu}_J(\boldsymbol{q}) = \frac{1}{J}\sum_{j=1}^{J}\delta(\boldsymbol{q} - \tilde{\boldsymbol{q}}(j\Delta t)). \qquad (49)$$

Because the random divisions are chosen independently, we cannot expect $\tilde{\mu}_J(\boldsymbol{q})$ to converge as $J \to \infty$. However, as we shall demonstrate with the following numerical example, the accumulation of the bias for each use of the random batches does not add up to an unbounded error. On the contrary, the perturbations of the interaction forces along the sampling path only result in a small bias in $\tilde{\mu}_J(\boldsymbol{q})$ from the target distribution $\pi(\boldsymbol{q})$.

We present a numerical experiment to observe the difference between $\tilde{\mu}_J(\boldsymbol{q})$ and $\pi(\boldsymbol{q})$ for sufficiently large J. Note that $\pi(\boldsymbol{q})$ is defined over the high-dimensional coordinate space $\mathbb{R}^{N\times 3P}$, it is intractable to directly compute $D(\tilde{\mu}_J\|\pi)$, and hence, we instead numerically simulate the distributions of certain observables associated with $\tilde{\mu}_J(\boldsymbol{q})$ and $\pi(\boldsymbol{q})$. In this example, we choose the observable operator to be the position $q_1^1 \in \mathbb{R}^3$ and the pairwise distance $|q_1^1 - q_1^2| \in \mathbb{R}$. Recall that the superscript denotes the index of particles, while the subscript denotes the index of beads. We plot the relative entropy between $\tilde{\mu}_J(\boldsymbol{q})$ and $\pi(\boldsymbol{q})$ in the observables along the sampling process in Fig. 3. As a comparison, the relative entropy for $\mu_J(\boldsymbol{q})$, the empirical measure of the pmmLang, is also plotted.
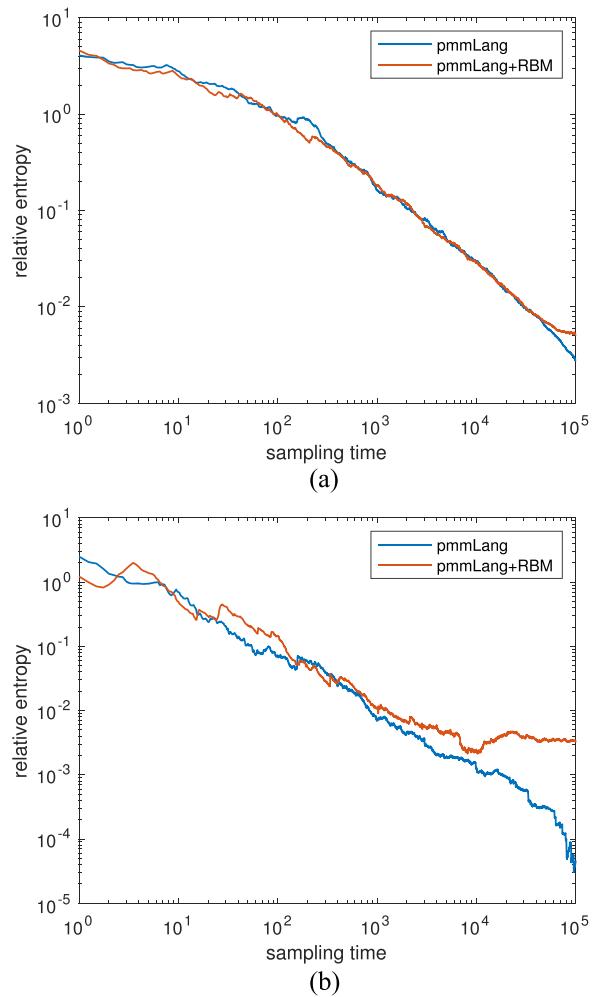

FIG. 3. Relative entropy in the numerical simulation of the Coulomb interacting system. (a) and (b) are associated with the observables in the position $q_1^1 \in \mathbb{R}^3$ and the pairwise distance $|q_1^1 - q_1^2| \in \mathbb{R}$, respectively. $m = 1$, $\beta = 4$, $P = 8$, $N = 32$, $\Delta t = 1/4$, $T = 10^5$, and $p = 2$. The target distribution $\pi(\boldsymbol{q})$ is computed by the pmmLang with a total sampling time of $T = 5 \times 10^5$.

We observe from Fig. 3 that the empirical measure of the pmmLang + RBM converges to $\pi(\boldsymbol{q})$ as fast as the pmmLang when the sampling time $T < 10^4$. However, as $T > 10^4$, there is a significant slowdown in the convergence of the pmmLang + RBM, especially in Fig. 3(b) where the observable is the pairwise distance $|q_1^1 - q_1^2| \in \mathbb{R}$. Finally, there is a certain bias of the empirical measure $\tilde{\mu}_J(\boldsymbol{q})$ from the target distribution $\pi(\boldsymbol{q})$. The slowdown in Fig. 3(b) is more noticeable than the top one because the RBM modifies the interaction forces and directly impacts the calculation of the pairwise distance. Meanwhile, the RBM has relatively less influence on the marginal distribution of a single particle.

The convergence of $\tilde{\mu}_J(\boldsymbol{q})$ to $\pi(\boldsymbol{q})$ also implies the convergence of the time average (42) to $\langle W_N(\boldsymbol{q})\rangle_\pi$, which can be verified in the following example. We plot the time averages [(41) and (42)] of the

pmmLang and the pmmLang + RBM in Fig. 4, where the observable operator is the kinetic energy. It can be seen from Fig. 4 that although (42) deviates from (41) at the first phase of the sampling process, (42) gradually converges to a fixed limit as time evolves. Furthermore, the bias of (41) from $\langle \hat{A} \rangle$ is small and diminishes if we shrink the time step.

To understand the approximation property of the pmmLang + RBM, we present a heuristic explanation on the decay of the relative entropy $D(\tilde{\mu}_J \| \pi)$. Our key observation lies in the fact that the use of the random batches does not affect the thermostat part of the Langevin dynamics. In addition, thus, the pmmLang + RBM can be viewed as an integrated process, where the thermostat part is constantly driving the empirical distribution approaching the target measure and the use of random batches gives rise to a sequence of random perturbations, which presents the

FIG. 4. pmmLang time average (41) and pmmLang + RBM time average (42) in the Coulomb interacting system, where the observable operator is the kinetic energy. [(a) and (b)] Two phases of the Langevin sampling, respectively. $m = 1$, $\beta = 4$, $P = 8$, $N = 32$, $\Delta t = 1/4, 1/16$, $T = 5000$, and $p = 2$.

convergences of the empirical distribution. The numerical results strong suggest that the thermostat effect is dominating the random perturbations, and as a consequence, although the empirical distribution $\tilde{\mu}_J(q)$ does not converge, it stays in a relatively small vicinity of the target distribution $\pi(q)$ for J sufficiently large. To further analyze the dominance of the thermostat in such an integrated process, we continue the discussion in two distinct phases of sampling.

In the first phase of sampling, the samples generated by the Langevin dynamics are statistically dependent on the choice of the initial state, and within a fairly short time, the sampling path becomes uncorrelated with the initial state. This period roughly corresponds to the upper plot in Fig. 3. During this phase of sampling, we observe that the time average of pmmLang + RBM is actually not very close to that of pmmLang, but the trends of both running averages are similar: After a short time, both of them start to fluctuate within a same small neighborhood. We remark that, in practice, the samples from the first phase are "burned-in" anyway due to their low qualify, but it is crucial that the consecutive random perturbations are suppressed by the thermostatting mechanism of the Langevin dynamics such that the running average of the pmmLang + RBM is able to approach a small vicinity of the true value by the end of the first phase. In probability language, it means that the difference between the probability distribution function of the stochastic process pmmLang + RBM and the target distribution significantly reduces within a short time period, after which the pmmLang + RBM is able to produce the correct samples from the target distribution.

In the second phase of sampling, the Langevin dynamics effectively produces a vast amount of samples of the target distribution such that the empirical measure is converging to the target Boltzmann distribution $\pi(q)$, which is manifested by the decaying of the relative entropy. When the random batches are used, the generated position samples $\tilde{q}(j\Delta t)$ from pmmLang + RBM are not, unfortunately, unbiased samples from $\pi(q)$. However, the thermostatting effect that dissipates the relative entropy with respect to $\pi(q)$ still cause that the difference between the empirical measure $\tilde{\mu}_J(q)$ and the Boltzmann distribution $\pi(q)$ diminishes in time. This argument is further confirmed by the numerical tests, as shown in Fig. 3, where we observe that the relative entropy of $D(\tilde{\mu}_J \| \pi)$ from pmmLang + RBM decays exponentially in time, although the convergence speed may be slightly reduced comparing with the decaying of $D(\mu_J \| \pi)$. This implies that a large number of small biases induced by the use of the RBM in long time sampling only add to a small bias in the empirical measure. Furthermore, we conclude that the use of random batches only leads to small random perturbations to the sample values of the observables, and thus, by the law of large numbers, we expect a small bias in the thermal average calculations.

We summarize the weak error analysis for pmmLang + RBM in the following mathematical statement: There exists a small constant $C_{p,\Delta t}$ depending only on the batch size p and the time step Δt such that for $J \gg 1$,

$$\left| \frac{1}{J} \sum_{j=1}^{J} W_N(\tilde{q}(j\Delta t)) - \langle W_N(q) \rangle_\pi \right| < C_{p,\Delta t}. \tag{50}$$

In other words, the fluctuation–dissipation relationship for the pmmLang + RBM can be established in an approximate sense, when
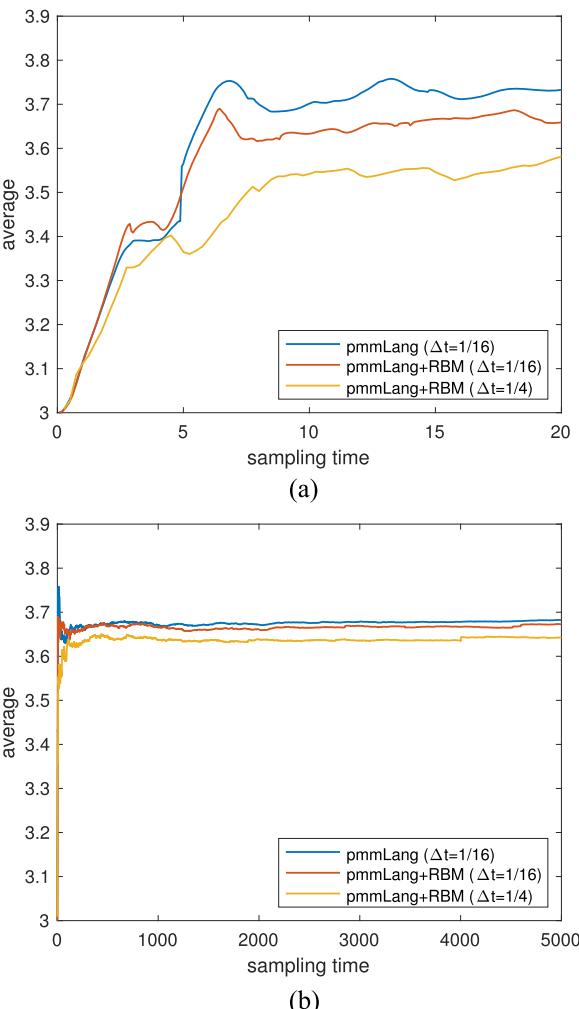
05 November 2025 21:36:51

the RBM mechanism is incorporated. The rigorous justification of this statement is not yet complete and beyond the scope of this paper.

C. Efficient calculation of the weight function

In Sec. II, we have mentioned that the calculation of the weight $W_N(\boldsymbol{q})$ also has the $O(NP^2)$ complexity if the observable operator is the kinetic energy or in the form of a summation over pairwise contributions. Using again the idea of random batches, we can make an unbiased approximation of $W_N(\boldsymbol{q})$ and reduce the complexity of calculating $W_N(\boldsymbol{q})$ from $O(NP^2)$ to $O(NP)$.

For the kinetic energy $\hat{A} = \hat{p}^2/(2m)$, rewrite the virial estimator (10) as

$$W_N(\boldsymbol{q}) = \frac{3P}{2\beta} + \frac{1}{2N}\langle \boldsymbol{q} - \bar{q}, \alpha\boldsymbol{q} + \nabla U^\alpha(\boldsymbol{q})\rangle_F, \quad (51)$$

where the modified potential $U^\alpha(\boldsymbol{q})$ defined in (34) is the sum of all interacting potentials $V^{(c)}(q_k^i - q_k^j)$. We can directly obtain an unbiased approximation of $\nabla U^\alpha(\boldsymbol{q})$ [and thus $W_N(\boldsymbol{q})$] from the random-batch approximation (37). For the position-dependent operator $\hat{A} = A(\hat{q})$ with $A(q)$ given in the form

$$A(q) = \frac{1}{P}\sum_{1 \leqslant i < j \leqslant P} a(q^i - q^j), \quad (52)$$

we randomly pick a batch $\mathscr{C}$ of size p from the group of P particles (a full division is not required now) and approximate $A(q)$ as

$$A(q) \approx \frac{P-1}{p(p-1)}\sum_{i,j \in \mathscr{C}, i < j} a(q^i - q^j), \quad (53)$$

which is unbiased. Hence, the weight $W_N(\boldsymbol{q})$ can be efficiently computed by

$$W_N(\boldsymbol{q}) \approx \frac{P-1}{Np(p-1)}\sum_{k=1}^N \sum_{i,j \in \mathscr{C}, i < j} a(q_k^i - q_k^j). \quad (54)$$

Finally, we emphasize that such approximation of the weight $W_N(\boldsymbol{q})$ does not change the original dynamics.

IV. SPLITTING MONTE CARLO METHOD

In Sec. III, we have shown that the pmmLang + RBM is an efficient approximate integrator of the pmmLang (35). However, when the interaction potential $V^{(c)}(q)$ is singular (e.g., Lennard-Jones or Morse potential), extremely small time steps Δt are needed to integrate the dynamics near the singular kernel, no matter if the RBM is used.

In the simulation of the pmmLang, the singular kernel of $V^{(c)}(q)$ limits the time step when the collision occurs, i.e., there are two particles $\boldsymbol{q}^i, \boldsymbol{q}^j$ close enough to each other. Although the collision is a rare event, the requirement for small time steps largely increases the computational cost. To overcome the singularity of the interaction potential, the pmmLang (35) should be modified to avoid making use of gradients of singular potentials, be compatible with the

random batch method, and do not change the invariant distribution (30) of the pmmLang (35).

In this section, we introduce a modified version of the pmmLang, which is based on the splitting Monte Carlo method.[25] Briefly speaking, we split the potential $U^\alpha(\boldsymbol{q})$ into the smooth part $U_1(\boldsymbol{q})$ and the singular part $U_2(\boldsymbol{q})$. At each time step, a proposal point $\boldsymbol{q}^*$ is generated by evolving the pmmLang driven by $U_1(\boldsymbol{q})$ and is accepted or rejected according to the potential difference $U_2(\boldsymbol{q}^*) - U_2(\boldsymbol{q})$. Therefore, the splitting Monte Carlo method can be seen as a variant of the Metropolis-adjusted Langevin algorithm (MALA).[48]

To illustrate the splitting Monte Carlo method, we split the interaction potential $V^{(c)}(q)$ into

$$V^{(c)}(q) = V_1^{(c)}(q) + V_2^{(c)}(q), \quad q \in \mathbb{R}^3, \quad (55)$$

where $V_2^{(c)}(q)$ is short-ranged and captures the singular part of $V^{(c)}(q)$, and thus, $V_1^{(c)}(q)$ is smooth. Define

$$U_1(\boldsymbol{q}) = \sum_{k=1}^N \sum_{1 \leqslant i < j \leqslant P} V_1^{(c)}(q_k^i - q_k^j), \quad (56)$$

$$U_2(\boldsymbol{q}) = \sum_{k=1}^N \sum_{1 \leqslant i < j \leqslant P} V_2^{(c)}(q_k^i - q_k^j), \quad (57)$$

and then $U^\alpha(\boldsymbol{q})$ is split into the sum of the smooth potential $U_1(\boldsymbol{q})$ and the singular potential $U_2(\boldsymbol{q})$. On the one hand, the pmmLang driven by the smooth $U_1(\boldsymbol{q})$ is

$$d\boldsymbol{q}^i = \boldsymbol{v}^i dt, \quad (58)$$

$$d\boldsymbol{v}^i = -\boldsymbol{q}^i dt - (L^\alpha)^{-1}\sum_{j \neq i}\nabla V_1^{(c)}(\boldsymbol{q}^i - \boldsymbol{q}^j)dt - \gamma\boldsymbol{v}^i dt$$

$$+ \sqrt{\frac{2\gamma(L^\alpha)^{-1}}{\beta_N}}d\boldsymbol{B}^i \quad (i = 1,\ldots,P),$$

whose invariant distribution is

$$\pi_1(\boldsymbol{q}, \boldsymbol{v}) \propto \exp\left(-\beta_N\left(\frac{1}{2}\langle \boldsymbol{v}, L^\alpha \boldsymbol{v}\rangle_F + \frac{1}{2}\langle \boldsymbol{q}, L^\alpha \boldsymbol{q}\rangle_F + U_1(\boldsymbol{q})\right)\right). \quad (59)$$

On the other hand, when the proposal $\boldsymbol{q}^*$ is accepted with probability

$$a(\boldsymbol{q}, \boldsymbol{q}^*) = \min\{1, e^{-\beta_N(U_2(\boldsymbol{q}^*) - U_2(\boldsymbol{q}))}\}, \quad (60)$$

the Metropolis algorithm for the singular potential $U_2(\boldsymbol{q})$ samples its distribution

$$\pi_2(\boldsymbol{q}) \propto e^{-\beta_N U_2(\boldsymbol{q})}. \quad (61)$$

Therefore, the key to assure that the coupled dynamics of the pmmLang (58) driven by $U_1(\boldsymbol{q})$ and the Metropolis algorithm for $U_2(\boldsymbol{q})$ is the obvious relation

$$\pi(\boldsymbol{q}, \boldsymbol{v}) \propto \pi_1(\boldsymbol{q}, \boldsymbol{v})\pi_2(\boldsymbol{q}). \quad (62)$$

Since the pmmLang is the second-order Langevin dynamics, special care should be taken to preserve the detailed balance.

Algorithm 2. Splitting Monte Carlo method for the pmmLang in a time step Δt.

Evolve the pmmLang in a timestep Δt:

$$d\boldsymbol{q}^i = \boldsymbol{v}^i dt,$$

$$d\boldsymbol{v}^i = -\boldsymbol{q}^i dt - (L^\alpha)^{-1} \sum_{j \neq i} \nabla V_1^{(c)}(\boldsymbol{q}^i - \boldsymbol{q}^j) dt - \gamma \boldsymbol{v}^i dt + \sqrt{\frac{2\gamma(L^\alpha)^{-1}}{\beta_N}} d\boldsymbol{B}^i. \quad (i = 1, \cdots, P)$$

Let $(\boldsymbol{q}^*, \boldsymbol{v}^*)$ be the proposal calculate above. Set $(\boldsymbol{q}, \boldsymbol{v}) = (\boldsymbol{q}^*, \boldsymbol{v}^*)$ with probability

$$a(\boldsymbol{q}, \boldsymbol{q}^*) = \min\{1, e^{-\beta_N(U_2(\boldsymbol{q}^*) - U_2(\boldsymbol{q}))}\},$$

otherwise set $(\boldsymbol{q}, \boldsymbol{v}) = (\boldsymbol{q}, -\boldsymbol{v})$.

According to Ref. 49, the detailed balance of the second-order Langevin dynamics is given by

$$\pi(\boldsymbol{q}, \boldsymbol{v}) T_{\Delta t}((\boldsymbol{q}, \boldsymbol{v}), (\boldsymbol{q}', \boldsymbol{v}')) = \pi(\boldsymbol{q}', -\boldsymbol{v}') T_{\Delta t}((\boldsymbol{q}', -\boldsymbol{v}'), (\boldsymbol{q}, -\boldsymbol{v})), \quad (63)$$

where $T_{\Delta t}((\boldsymbol{q}, \boldsymbol{v}), (\boldsymbol{q}', \boldsymbol{v}'))$ is the transition probability density from $(\boldsymbol{q}, \boldsymbol{v})$ to $(\boldsymbol{q}', \boldsymbol{v}')$ in a time step Δt. The velocities $\boldsymbol{v}, \boldsymbol{v}'$ appearing in the right-hand side of (63) are flipped as $-\boldsymbol{v}, -\boldsymbol{v}'$; thus, in the Metropolis algorithm for $U_2(\boldsymbol{q})$, the velocity $\boldsymbol{v}$ should also be flipped if the proposal $\boldsymbol{q}^*$ is rejected. To sum up, the splitting Monte Carlo method for the pmmLang (denoted by the pmmLang + split) is given in Algorithm 2. The correctness of Algorithm 2 is guaranteed by the following theorem.

Theorem IV.1. *Let $T_{\Delta t}((\boldsymbol{q}, \boldsymbol{v}), (\boldsymbol{q}', \boldsymbol{v}'))$ be the transition probability density of the splitting Monte Carlo method (Algorithm 2) in a time step Δt, which satisfies*

$$\int T_{\Delta t}((\boldsymbol{q}, \boldsymbol{v}), (\boldsymbol{q}', \boldsymbol{v}')) d\boldsymbol{q}' d\boldsymbol{v}' = 1, \quad (64)$$

and then the detailed balance holds

$$\pi(\boldsymbol{q}, \boldsymbol{v}) T_{\Delta t}((\boldsymbol{q}, \boldsymbol{v}), (\boldsymbol{q}', \boldsymbol{v}')) = \pi(\boldsymbol{q}', -\boldsymbol{v}') T_{\Delta t}((\boldsymbol{q}', -\boldsymbol{v}'), (\boldsymbol{q}, -\boldsymbol{v})). \quad (65)$$

The detailed balance (65) implies that $\pi(\boldsymbol{q}, \boldsymbol{v})$ is the invariant distribution of the splitting Monte Carlo method (Algorithm 2),

$$\int \pi(\boldsymbol{q}, \boldsymbol{v}) T_{\Delta t}((\boldsymbol{q}, \boldsymbol{v}), (\boldsymbol{q}', \boldsymbol{v}')) d\boldsymbol{q} d\boldsymbol{v} = \pi(\boldsymbol{q}', \boldsymbol{v}'). \quad (66)$$

In the large interacting particle system, the computational cost of Algorithm 2 within a single time step originates from evaluating the interaction forces $\nabla V_1^{(c)}(q_k^i - q_k^j)$ and the potential difference $U_2(\boldsymbol{q}^*) - U_2(\boldsymbol{q})$. Note that $U_2(\boldsymbol{q})$ defined in (56) and (57) is the sum of all short-ranged interacting potentials $V_2^{(c)}(q_k^i - q_k^j)$ and thus can be efficiently calculated by cutoff or using data structures such as the cell list.[18] In this paper, we employ the cutoff method to compute $U_2(\boldsymbol{q})$, i.e., the particle pair (q_k^i, q_k^j) is counted in the summation (56) and (57) only when their distance $|q_k^i - q_k^j|$ is less than the cutoff distance. Therefore, for large interacting particle systems, the

---

Algorithm 3. RBM with splitting Monte Carlo for the pmmLang in a time step Δt.

Randomly divide the P particles into n batches $\mathscr{C}_1, \cdots, \mathscr{C}_n$ of size p, where $n = P/p$.
for $l = 1, \cdots, n$ **do**

    Evolve the pmmLang within the batch $\mathscr{C}_l$ in a timestep Δt:

$$d\boldsymbol{q}^i = \boldsymbol{v}^i dt$$

$$d\boldsymbol{v}^i = -\boldsymbol{q}^i dt - \frac{P-1}{p-1}(L^\alpha)^{-1} \sum_{j \in \mathscr{C}_l, j \neq i} \nabla V_1^{(c)}(\boldsymbol{q}^i - \boldsymbol{q}^j) dt - \gamma \boldsymbol{v}^i dt + \sqrt{\frac{2\gamma(L^\alpha)^{-1}}{\beta_N}} d\boldsymbol{B} \quad (i \in \mathscr{C}_l)$$

end
Let $(\boldsymbol{q}^*, \boldsymbol{v}^*)$ be the proposal calculated above. Set $(\boldsymbol{q}, \boldsymbol{v}) = (\boldsymbol{q}^*, \boldsymbol{v}^*)$ with probability

$$a(\boldsymbol{q}, \boldsymbol{q}^*) = \min\{1, e^{-\beta_N(U_2(\boldsymbol{q}^*) - U_2(\boldsymbol{q}))}\},$$

otherwise set $(\boldsymbol{q}, \boldsymbol{v}) = (\boldsymbol{q}, -\boldsymbol{v})$.

---

majority of the computational cost is still the calculation of the interaction forces $\nabla V_1^{(c)}(q_k^i - q_k^j)$.

Now, we employ the RBM to reduce the complexity due to interaction forces $\nabla V_1^{(c)}(q_k^i - q_k^j)$. At each iteration, we randomly divide the P particles to $n = P/p$ batches, where each batch $\mathscr{C}$ is of size p. The pmmLang driven by the smooth potential $U_1(q)$ within the batch $\mathscr{C}$ is then

$$d q^i = v^i dt,$$

$$d v^i = -q^i dt - \frac{P-1}{p-1}(L^\alpha)^{-1} \sum_{j \in \mathscr{C}, j \neq i} \nabla V_1^{(c)}(q^i - q^j) dt - \gamma v^i dt$$

$$+ \sqrt{\frac{2\gamma(L^\alpha)^{-1}}{\beta_N}} dB \quad (i \in \mathscr{C}).$$

(67)

By coupling the random-batch approximated pmmLang (67) driven by $U_1(q)$ and the Metropolis–Hastings algorithm for $U_2(q)$, we obtain the RBM with splitting Monte Carlo for the pmm-Lang (denoted by the pmmLang + RBM + split, presented in Algorithm 3).

To show how the splitting Monte Carlo method accelerates the sampling efficiency, we compare the performance of the pmmLang and the pmmLang + split in the numerical example below. In the mixed Coulomb–Lennard-Jones system [the potential function and the splitting scheme are defined in (69)–(71) of Sec. V], we plot in Fig. 5 the time averages computed by the pmmLang and the pmm-Lang + split with various time steps. Different inverse temperatures $\beta = 1, 4$ are tested.

We observe from Fig. 5 that the time average of the pmm-Lang is very sensitive to the time step Δt, while in the pmmLang
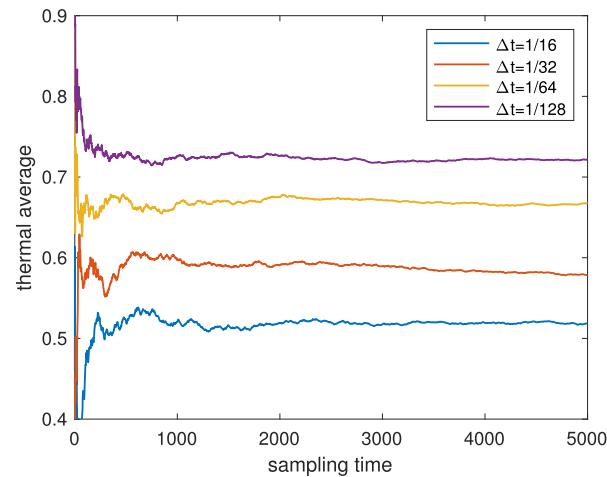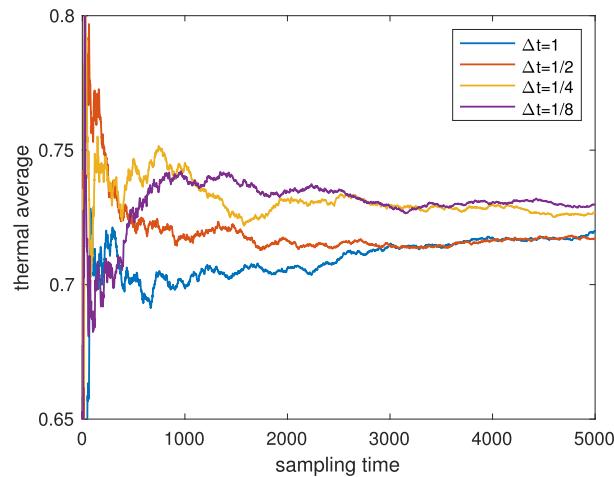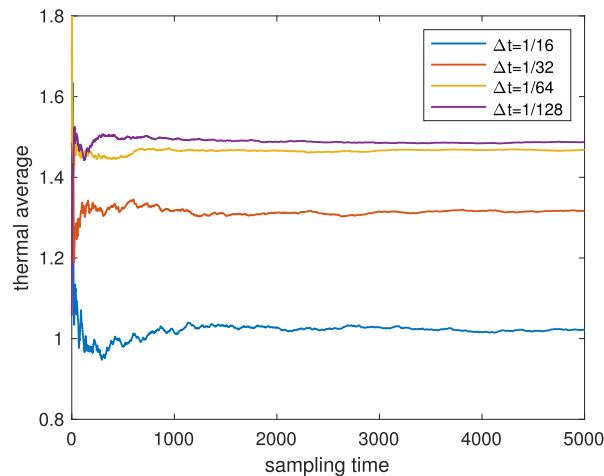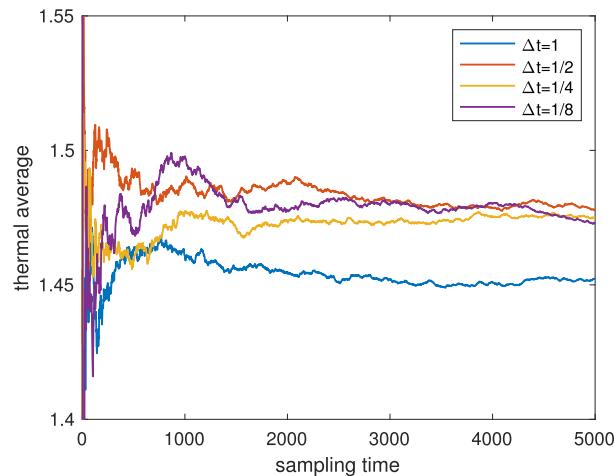

FIG. 5. Time averages computed of the pmmLang and the pmmLang + split in the mixed Coulomb–Lennard-Jones system, where the observable operator is defined in (76). The left [(a) and (c)] and right [(b) and (d)] panels are for the pmmLang and the pmmLang + split, and the top [(a) and (b)] and bottom [(c) and (d)] panels are associated with the inverse temperature $\beta = 1, 4$. $m = 1, P = 8, N = 8, T = 6000, \gamma = 2$, and $p = 2$.

+ split, we can adopt relatively large time steps to obtain the correct thermal average, which greatly improves the efficiency of the simulation.

Despite of the satisfactory performance of the pmmLang + split in this example, the numerical efficiency of this Monte Carlo-type method can be influenced by the rejection rate along the sampling process. When the number of beads N or the number of particles P is large, it is likely that the high-dimensional nature of the dynamics will slow down the simulation since the rejection rate is high. A detailed numerical investigation of the rejection rate with different parameters is presented in Sec. V.

V. NUMERICAL TESTS

A. Examples of the interacting particle system

In Sec. III, we have proposed and analyzed the efficient sampling method for the quantum interacting particle systems, the pmmLang + RBM. To further explore this method, we present in this section more numerical results, where different parameters and interaction potentials are tested. The efficiency and the error in the computing thermal average $\langle \hat{A} \rangle$ are primarily used to quantify the numerical performance of the pmmLang + RBM.

In the quantum system (1), we choose the interacting potential $V^{(c)}(q)$ as either the Coulomb potential

$$V^{(c)}(q) = \frac{\kappa}{r}, \quad q \in \mathbb{R}^3 \tag{68}$$

or the mixed Coulomb–Lennard-Jones potential

$$V^{(c)}(q) = \begin{cases} \frac{1}{6}\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6\right) + 1, & r < \sigma \\ \frac{\sigma}{r}, & r \geq \sigma, \end{cases} \tag{69}$$

where $\kappa = 1$, $\sigma = 0.3$, and $r = |q|$. The kernel of (69) is provided by the Lennard-Jones potential and is much more singular than the Coulomb potential. Hence, we employ the splitting Monte Carlo method introduced in Sec. IV to split the mixed potential (69) into the sum of

$$V_1^{(c)}(q) = \begin{cases} 2 - \frac{r}{\sigma}, & r < \sigma \\ \frac{\sigma}{r}, & r \geq \sigma \end{cases} \tag{70}$$

and

$$V_2^{(c)}(q) = \begin{cases} \frac{1}{6}\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6\right) + 1, & r < \sigma \\ 0, & r \geq \sigma, \end{cases} \tag{71}$$

where $V_1^{(c)}(q)$ is smooth and $V_2^{(c)}$ is short-ranged. The graphs of the Coulomb potential (68) and the mixed potential (69) with its splitting scheme (70) and (71) are shown in Fig. 6.

In the quantum system (1), the external potential is assumed to be harmonic (33). We choose the constant α_0 (and thus the regularization parameter α) as
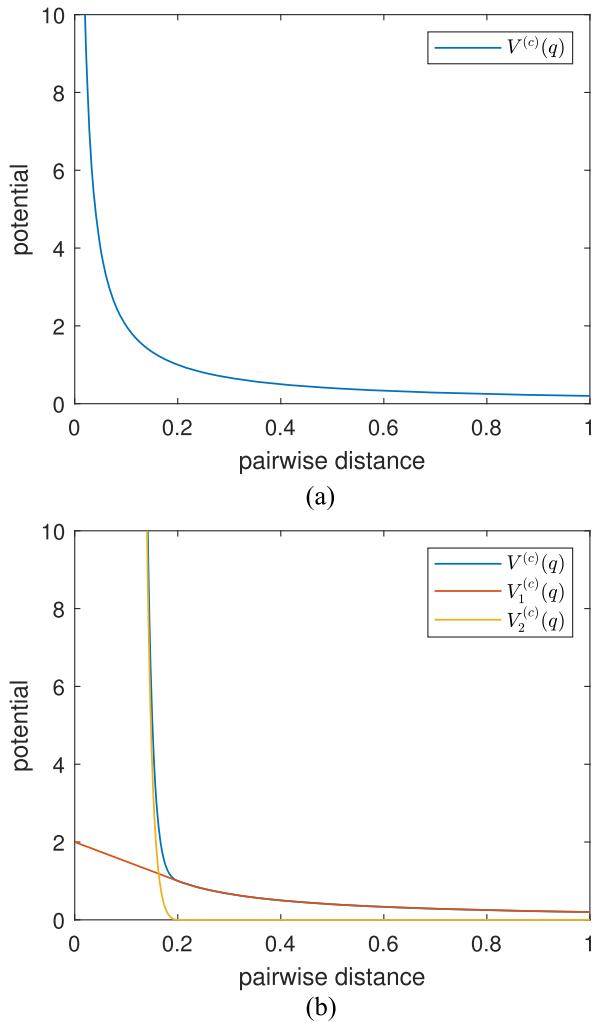

FIG. 6. Interacting potential $V^{(c)}(q)$ with its splitting scheme. (a) Coulomb potential (68). (b) Mixed Coulomb–Lennard-Jones potential (69) and the splitting potentials $V_1^{(c)}(q)$, $V_2^{(c)}(q)$.

$$\alpha_0 = \alpha = P^{-\frac{2}{3}}, \tag{72}$$

where P is the number of particles.

The idea behind the choice (72) is explained as follows: For a single particle placed in the external potential $V^{(o)}(q)$, its Boltzmann distribution $\pi_0(q)$ is

$$\pi_0(q) \propto \exp(-\beta V^{(o)}(q)) = \exp\left(-\frac{\beta \alpha_0}{2}|q|^2\right), \quad q \in \mathbb{R}^3. \tag{73}$$

This particle is then thought to be confined in a finite region in $\mathbb{R}^3$, roughly determined by the inequality

$$\exp\left(-\frac{\beta \alpha_0}{2}|q|^2\right) \geq c \tag{74}$$

for some small constant c. The region (74) is actually a ball with volume $\left(\frac{-2 \log c}{\beta \alpha_0}\right)^{\frac{3}{2}}$, and the choice $\alpha_0 = P^{-\frac{2}{3}}$ assures that the region volume is of order $O(P)$. Thus, if all the P particles are confined in the external potential, they can be distributed within the region (74) with a proper pairwise distance.

The position-dependent observable $\hat{A} = A(\hat{q})$ is chosen in the form of (53). For the Coulomb system (68), choose $a(q) = V^{(c)}(q)$ and then

$$A(q) = \frac{1}{P} \sum_{1 \leqslant i \leqslant j \leqslant P} \frac{\kappa}{|q^i - q^j|}, \quad q \in \mathbb{R}^{3P}, \tag{75}$$

is the average interacting potential of the system. For the mixed system (69), choose $a(q) = e^{-\theta|q|^2}$ with $\theta = 0.1$ and then

$$A(q) = \frac{1}{P} \sum_{1 \leqslant i \leqslant j \leqslant P} e^{-\theta|q^i - q^j|^2}, \quad q \in \mathbb{R}^{3P}. \tag{76}$$

B. Outline of the numerical simulation method

We outline the procedure to compute the thermal average $\langle \hat{A} \rangle$ defined in (42) in this paper. By choosing sufficiently large N, $\langle \hat{A} \rangle$ is approximated by the ensemble average $\langle W_N(\boldsymbol{q}) \rangle_\pi$. Then, the pmm-Lang and the pmmLang + RBM estimate $\langle W_N(\boldsymbol{q}) \rangle_\pi$ by the time averages [(41) and (42)].

We stress that the calculation of the time averages only requires a single sampling path; thus, (41) and (42) are both random variables. That is to say, the values of (41) and (42) depend on the Brownian motion and the choices of random divisions along the sampling process. In the numerical experiments below, only one typical sampling path is used to compute the time averages [(41) and (42)].

We use the BAOAB scheme[50,51] to integrate the pmmLang dynamics numerically. By combining the velocity Verlet method[51] with the Langevin thermostat, we obtain the BAOAB scheme for the pmmLang (29) in a time step Δt,

$$\boldsymbol{v}^{j*} = \boldsymbol{v}^j - (\boldsymbol{q}^j + (L^\alpha)^{-1} \nabla U^\alpha(\boldsymbol{q}^j)) \frac{\Delta t}{2},$$

$$\boldsymbol{q}^{j+\frac{1}{2}} = \boldsymbol{q}^j + \boldsymbol{v}^{j*} \frac{\Delta t}{2},$$

$$\boldsymbol{v}^{j**} = e^{-\gamma \Delta t} \boldsymbol{v}^{j*} + \sqrt{\frac{1 - e^{-2\gamma \Delta t}}{\beta_N}} \boldsymbol{\eta},$$

$$\boldsymbol{q}^{j+1} = \boldsymbol{q}^{j+\frac{1}{2}} + \boldsymbol{v}^{j**} \frac{\Delta t}{2},$$

$$\boldsymbol{v}^{j+1} = \boldsymbol{v}^{j**} - (\boldsymbol{q}^{j+1} + (L^\alpha)^{-1} \nabla U^\alpha(\boldsymbol{q}^{j+1})) \frac{\Delta t}{2},$$

where $\boldsymbol{\eta} \in \mathbb{R}^{N \times 3P}$ is a random variable and each column of $\boldsymbol{\eta}$ obeys the Gaussian distribution $N(0, (L^\alpha)^{-1})$. Similar BAOAB schemes can be derived for the pmmLang within the batch (39) or the pmmLang in the splitting Monte Carlo method [(58) and (67)].

Since the spectrum of L^α is known, we can sample from the Gaussian distribution $N(0, (L^\alpha)^{-1})$ using the fast Fourier transform (FFT), whose complexity is only $O(N \log N)$ (see Appendix A for a detailed mathematical formulation). Therefore, the computational cost due to algebraic operations is $O(N \log N P)$. In the interacting particle system with N being not relatively large, the complexity

from the interaction forces is still the dominant difficulty in the numerical simulation.

Finally, we point out that the performance of the BAOAB scheme for the pmmLang may depend on N, although the continuous-time pmmLang dynamics has been shown to have a dimension-independent convergence rate.[28] Designing a dimension-independent integrator for the pmmLang is beyond the scope of this article.

C. Tests of the pmmLang + RBM

1. Autocorrelation

First, we aim to investigate the impacts of the RBM on the variance of the estimator, by computing the autocorrelation (abbreviated by AC) of the time averages [(41) and (42)] numerically.

Recall that in Sec. III, we propose the RBM approach to reduce the complexity due to weight function $W_N(\boldsymbol{q})$ if the observable operator is in specific forms. Thus, in practice, there are two ways to implement the pmmLang + RBM: one is to use the RBM in the dynamics (Algorithm 35) but compute $W_N(\boldsymbol{q})$ as it is and the other is to use the RBM in both the dynamics and calculation of the weight function. To analyze the effects of the RBM, we will test both versions of the pmmLang + RBM.

In the Coulomb system, we compute the pmmLang and the pmmLang + RBM time averages [(41) and (42)], where the observable operator of interest is the kinetic energy and both versions of the pmmLang + RBM are considered. The autocorrelation of the time averages is shown in Fig. 7. In Fig. 7 and Table I, the autocorrelation of the pmmLang + RBM decays faster than the pmmLang. Since the random divisions at different time steps are independent, it is likely that the weak correlation of the interaction forces in different time steps leads to a small autocorrelation in the pmmLang + RBM. Still, the randomness in the batch force approximation (37) enlarged the effective variance of the estimator. Compared to
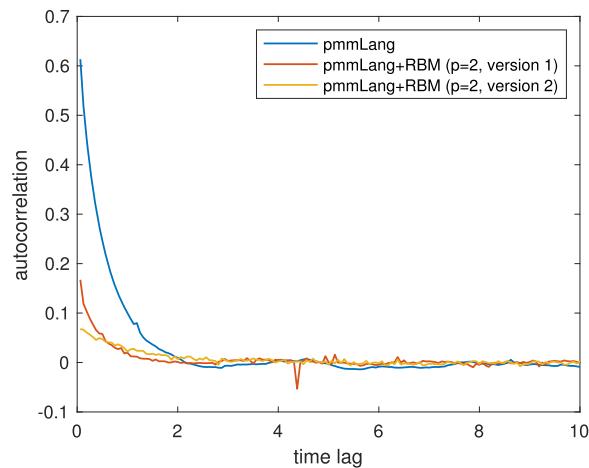

FIG. 7. Autocorrelation of the time averages [(41) and (42)] in the Coulomb interacting system. The blue, red, and yellow curves are the autocorrelations of the pmmLang, the pmmLang + RBM (version 1), and the pmmLang + RBM (version 2), respectively. $m = 1$, $\beta = 4$, $P = 16$, $N = 16$, $\Delta t = 1/16$, $T = 6000$, $\gamma = 2$, and $p = 2$.

TABLE I. Autocorrelation time, effective variance, and mean squared error of the time averages [(41) and (42)]. The first column is for the pmmLang, and the second and the third columns are for the two versions of the pmmLang + RBM, respectively. The reference value is calculated by the pmmLang with $\Delta t = 1/64$.

Methods	pmmLang	w/RBM (ver. 1)	w/RBM (ver. 2)
AC time	1.10	0.38	0.40
Variance	2.18×10^{-5}	5.94×10^{-5}	2.40×10^{-4}
MSE	1.10×10^{-3}	5.37×10^{-3}	5.55×10^{-3}

TABLE II. Effective variance of the time averages [(41) and (42)] in the pmmLang and the pmmLang + RBM when the number of particles $P = 16$.

No. of Beads	pmmLang	w/RBM, $p = 2$	w/RBM, $p = 4$
4	2.50×10^{-5}	9.83×10^{-4}	1.09×10^{-4}
8	2.76×10^{-5}	1.13×10^{-3}	1.85×10^{-4}
16	2.42×10^{-5}	2.87×10^{-4}	8.34×10^{-5}
32	2.18×10^{-5}	2.40×10^{-4}	5.03×10^{-5}
64	2.22×10^{-5}	2.42×10^{-4}	4.82×10^{-5}
128	2.03×10^{-5}	1.47×10^{-4}	4.18×10^{-5}

the pmmLang of version 1, version 2 computes the weight function $W_N(\boldsymbol{q})$ approximately and thus has additional variance in the estimator.

Since the random approximation to the weight function $W_N(\boldsymbol{q})$ is necessary to reduce the computational cost per time step, in the following tests, we always employ the pmmLang + RBM of version 2. That is to say, we use the RBM in both the dynamics and the calculation of the weight function.

2. Convergence with the number of beads

We study the convergence of pmmLang + RBM with the number of beads N. In the PIMD representation, the ensemble

average $\langle W_N(\boldsymbol{q}) \rangle_\pi$ converges to the exact thermal average $\langle \hat{A} \rangle$ as the number of beads $N \to \infty$; hence, it is necessary to check if the pmmLang + RBM time averages (42) also possess this convergence property.

In the Coulomb interacting system, we plot in Fig. 8 the time averages [(41) and (42)] of the pmmLang and the pmmLang + RBM, where the observable operator of interest is the kinetic energy. Different numbers of beads are tested to study the convergence of the pmmLang and the pmmLang + RBM. When the number of particles $P = 16$, we record the effective variance of (41) and (42)
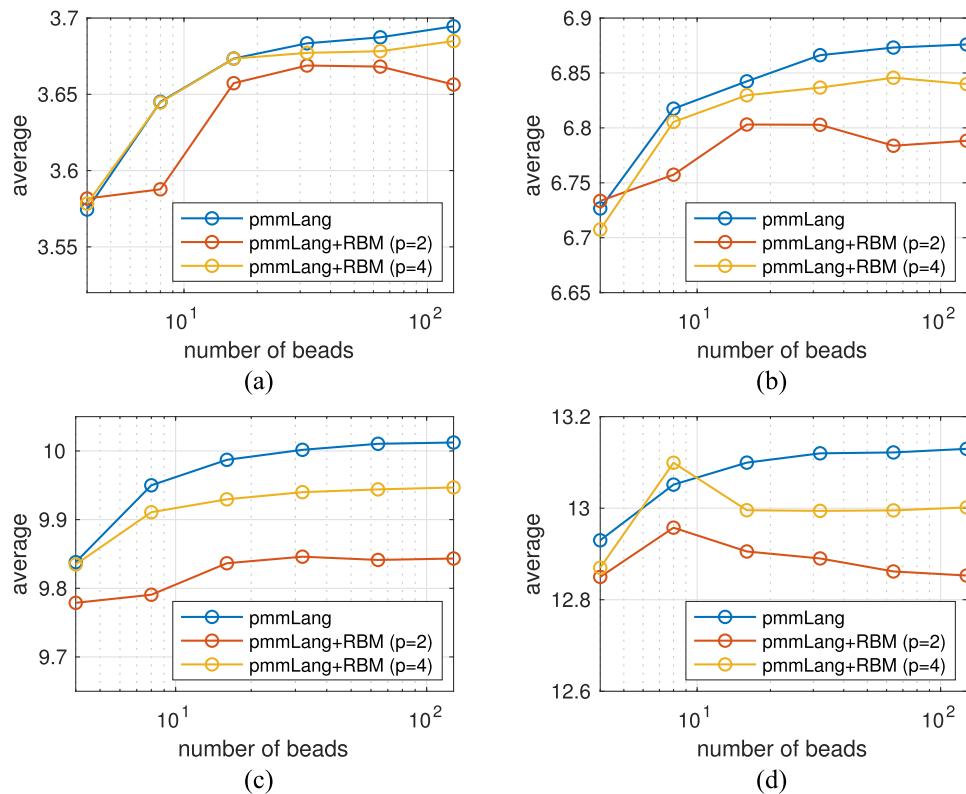

FIG. 8. Time averages [(41) and (42)] of the pmmLang and the pmmLang + RBM in the Coulomb interacting system. (a)–(d) are associated with the number of particles $P = 8, 16, 24, 32$, respectively. The blue curve is for the pmmLang, and the red and yellow curves are for the pmmLang + RBM with batch sizes $p = 2, 4$. $m = 1$, $\beta = 4$, $\Delta t = 1/16$, and $T = 6000$. In the x-axis, the number of beads N varies in $4, 8, \ldots, 128$.
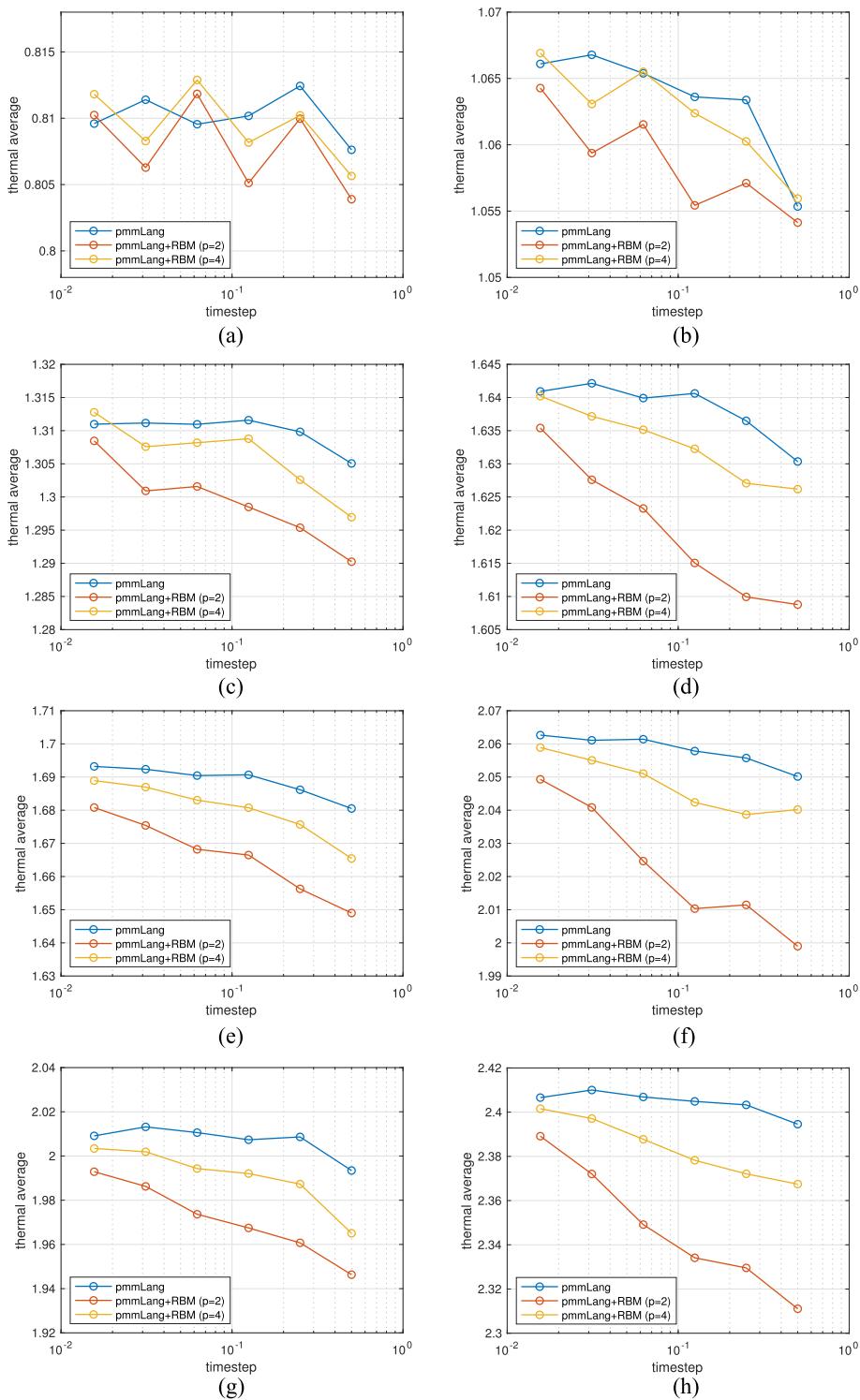
FIG. 9. Time averages [(41) and (42)] of the pmmLang and the pmmLang + RBM in the Coulomb interacting system. The left [(a), (c), (e), and (g)] and right [(b), (d), (f), and (h)] panels represent the inverse temperature $\beta = 1$ and 4. The top to bottom panels [(a) and (b), (c) and (d), (e) and (f), and (g) and (h)] are associated with the number of particles $P = 8, 16, 24, 32$, respectively, and the corresponding spread in the y-axis is 0.02, 0.04, 0.08, 0.12. $m = 1$, $N = 16$, $\Delta t = 1/16$, $T = 10\,000$, $\gamma = 2$, and $p = 2$. In the x-axis, the time step Δt varies in $1/2, 1/4, \ldots, 1/64$.

TABLE III. Relative error of the time averages [(41) and (42)] of the pmmLang and the pmmLang + RBM when $\beta = 4$. The top and bottom tables are for the time step $\Delta t = 1/4, 1/16$, respectively. The reference values are computed by the pmmLang with $\Delta t = 1/64$.

No. of Particles	pmmLang (%)	w/RBM, $p = 2$	w/RBM, $p = 4$
8	0.26	0.84	0.55
16	0.27	1.89	0.84
24	0.33	2.48	1.16
32	0.14	3.20	1.43

No. of Particles	pmmLang (%)	w/RBM, $p = 2$ (%)	w/RBM, $p = 4$ (%)
8	0.07	0.43	0.06
16	0.06	1.07	0.35
24	0.06	1.84	0.56
32	0.01	2.39	0.78

TABLE IV. Cost of CPU time (second) in one single time step of the pmmLang and the pmmLang + RBM.

No. of Particles	pmmLang	w/RBM, $p = 2$	w/RBM, $p = 4$
8	7.6×10^{-3}	1.3×10^{-3}	2.7×10^{-3}
16	2.2×10^{-2}	2.4×10^{-3}	5.3×10^{-3}
24	5.1×10^{-2}	3.3×10^{-3}	7.5×10^{-3}
32	1.0×10^{-1}	4.5×10^{-3}	9.6×10^{-3}

in Table II. In Fig. 8, we observe that pmmLang + RBM time averages (42) converge as the number of beads N enlarges. Besides, Table II shows that the variance of the estimator is not sensitive to the number of beads N; thus, it is safe increase N in the pmmLang + RBM to obtain a more accurate approximation of the thermal average.

TABLE V. Rejection rates of the pmmLang + split and the pmmLang + RBM + split with numbers of particles P along the sampling process. The tables from top to bottom correspond to the time step $\Delta t = 1/8, 1/16, 1/32$, respectively. The number of beads $N = 16$, the inverse temperature $\beta = 4$, and the mass $m = 1$. The rejection rates are computed with the total sampling time $T = 500$.

No. of Particles	pmmLang + split (%)	w/RBM, $p = 2$ (%)	w/RBM, $p = 4$ (%)
8	9.73	9.40	9.53
16	15.37	17.92	15.25
24	19.43	23.60	21.75
32	26.40	29.93	27.72

No. of Particles	pmmLang + split (%)	w/RBM, $p = 2$ (%)	w/RBM, $p = 4$ (%)
8	6.05	5.70	4.79
16	10.85	11.54	10.27
24	13.05	18.71	14.22
32	17.50	23.54	20.59

No. of Particles	pmmLang + split (%)	w/RBM, $p = 2$ (%)	w/RBM, $p = 4$ (%)
8	2.95	2.58	3.60
16	6.00	5.33	5.45
24	8.26	9.31	8.93
32	10.50	13.69	11.69

TABLE VI. Rejection rates of the pmmLang and the pmmLang + RBM + split with different numbers of beads N along the sampling process. The tables from top to bottom correspond to the time step $\Delta t = 1/8, 1/16$, respectively. The number of particles $P = 16$, the inverse temperature $\beta = 4$, and the mass $m = 1$. The rejection rates are computed with the total sampling time $T = 500$.

No. of Beads	pmmLang + split (%)	w/RBM, $p = 2$ (%)	w/RBM, $p = 4$ (%)
16	15.54	18.29	18.21
32	26.19	30.83	28.63
64	47.38	47.00	47.75
128	74.62	75.62	71.25

| No. of Beads	pmmLang + split (%)	w/RBM, $p = 2$ (%)	w/RBM, $p = 4$ (%)
16	9.94	10.87	11.10
32	19.10	16.13	18.94
64	34.31	37.81	38.84
128	66.44	58.28	67.52
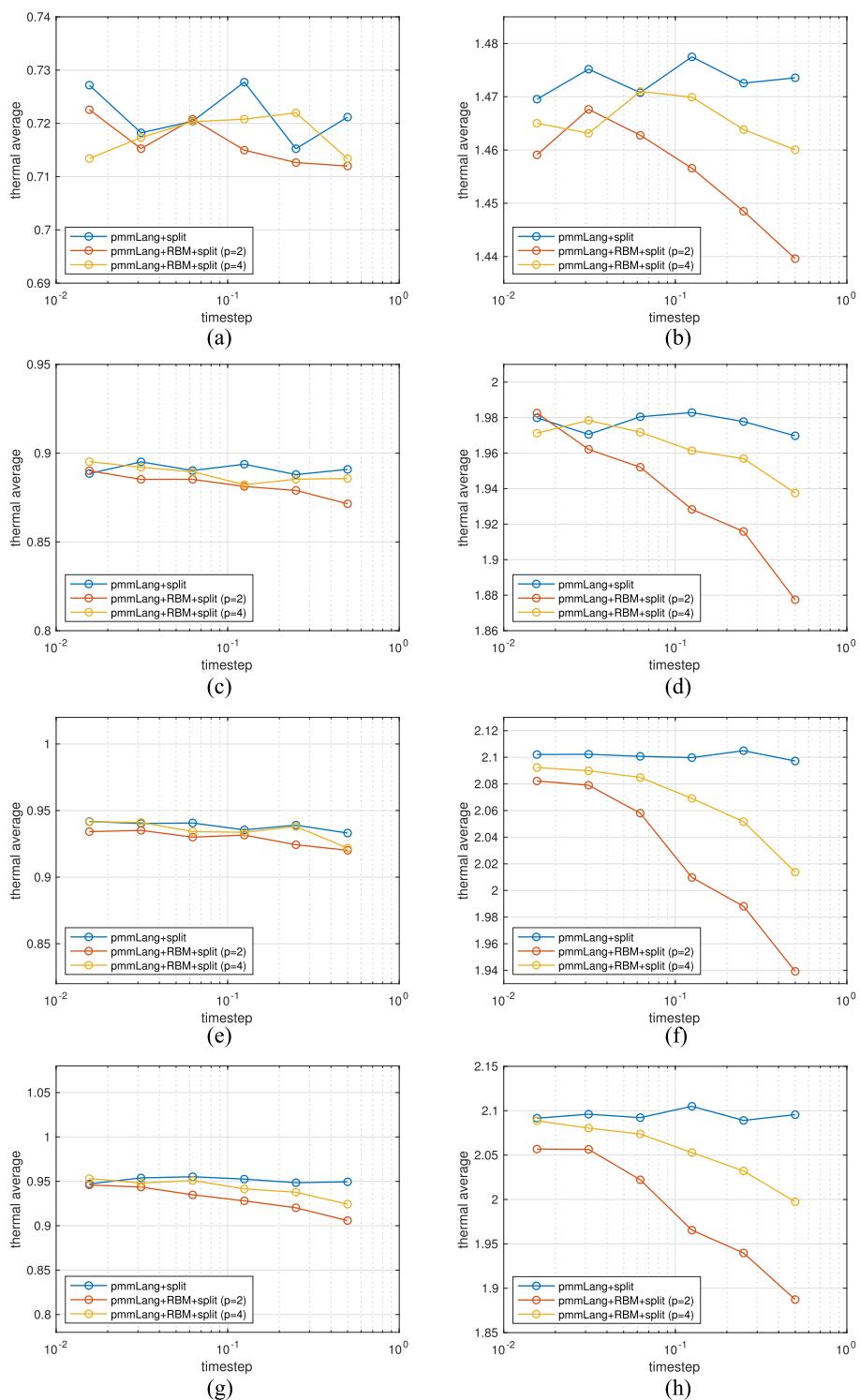
FIG. 10. Time averages of the pmmLang + split and the pmmLang + RBM + split in the mixed Coulomb–Lennard-Jones system. The left [figures (a), (c), (e), and (g)] and right [figures (b), (d), (f), and (h)] panels represent the inverse temperature $\beta = 1$ and 4. The top to bottom panels [(a) and (b), (c) and (d), (e) and (f), and (g) and (h)] are associated with the number of particles $P = 8, 16, 24, 32$, respectively, and the corresponding spread in the y-axis is 0.05, 0.15, 0.2, 0.3. $m = 1$, $N = 16$, $\Delta t = 1/16$, $T = 10\,000$, $\gamma = 2$, and $p = 2$. In the x-axis, the time step Δt varies in $1/2, 1/4, \ldots, 1/64$.

We also note that, even if N is sufficiently large, there is a bias of the pmmLang + RBM time average (42) from $\langle \hat{A} \rangle$. Moreover, the bias becomes significant when the number of particles P grows large or the batch size p is small as 2. This bias though can be reduced with fixed batch size p by decreasing the time step Δt, which are to be confirmed in the tests of Subsection V C 3. Overall, the pmm-Lang + RBM time average (42) is still an accurate approximation of $\langle \hat{A} \rangle$. When $N = 128$, the relative error of (42) from $\langle \hat{A} \rangle$ is no more than 2.5%. This bias though can be reduced with fixed batch size p by decreasing the time step Δt, which are to be confirmed in the follow-up tests. For simplicity, in the following, we will take the number of beads $N = 16$ in both the pmmLang and the pmmLang + RBM.

3. Error in the calculation of the thermal average

In this part, we test the error of time average (42) of the pmm-Lang + RBM with respect to varying time steps Δt. In the Coulomb interacting system, we employ the pmmLang and the pmmLang + RBM to compute the time averages [(41) and (42)], where the observable of interest is the position-dependent one with $A(q)$ given in (75). In Fig. 9, we plot the time averages [(41) and (42)] with different time steps. Different numbers of particles are used to test the sampling methods. The relative error of the time averages for $\beta = 4$ is shown in Table III. In Fig. 9, we observe that the bias of the pmmLang + RBM time average (42) from $\langle \hat{A} \rangle$ diminishes as the time step Δt approaches 0. In addition, the bias associated with the batch size $p = 4$ is much less than $p = 2$. The numerical results above confirm our arguments on the constant $C_{p, \Delta t}$ in the weak error analysis (50).

Finally, we report the time complexity of the pmmLang and the pmmLang + RBM in Table IV. When the number of particles P is large, the pmmLang + RBM is much more efficient than the pmmLang, while the relative error always still keeps small.

D. Tests of the pmmLang + RBM + split

1. Rejection rates

We focus on the mixed Coulomb–Lennard-Jones system, where the pairwise interacting potential $V^{(c)}(q)$ is very singular. As we have demonstrated in Sec. IV, we shall use the splitting Monte Carlo method to avoid using too small time steps. In this case, the rejection rate is an important index to assess the numerical efficiency of the method. Below, we compute the rejection rates of the pmmLang + split and the pmmLang + RBM + split with different parameters such as N, P, p, and Δt.

In Tables V and VI, we fix the number of beads $N = 16$ and the number of particles $P = 16$, respectively, and compute the rejection rate of the pmmLang + split and the pmmLang + RBM + split. The parameters used are inverse temperature, $\beta = 4$, mass, $m = 1$, and different batch sizes, p, and time steps, Δt.

Under the proper scaling $\alpha = P^{-\frac{2}{3}}$ specified in (72), Table V shows that the rejection rate gently grows with the number of particles P and can be significantly reduced by shrinking the time step. For example, when $N = 16$ and $P = 32$, the time step $\Delta t = 1/16$ makes sure the rejection rate not larger than 25%. Additionally, the application of the RBM does not change the rejection rate too much.

However, when we fix the number of particles P, Table VI shows that the rejection rate rapidly deteriorates with the number of beads N, even in the pmmLang + split without the use of random batches. This is due to growing N makes a particle $\boldsymbol{q}^i \in \mathbb{R}^{N \times 3}$ more easily to collide with other particles. How to properly design the splitting Monte Carlo method to overcome the effects of the singular potential within the PIMD framework remains a topic to study in the future.

2. Error in the calculation of the thermal average

Next, we test the error of the time average computed by the pmmLang + RBM + split with respect to varying time steps Δt. In the mixed Coulomb–Lennard-Jones system, we employ the pmm-Lang + split and the pmmLang + RBM + split to compute the time averages, where the observable of interest is the position-dependent one with $A(q)$ given in (76). In Fig. 10, we plot the time averages computed by the pmLang + split and the pmmLang + RBM + split with different time steps. Different numbers of particles are used to test the sampling methods. The relative error of the time averages for $\beta = 4$ is in Table VII. It can be seen from Fig. 10 and Table VII

TABLE VII. Relative error of the time averages of the pmmLang + split and the pmmLang + RBM + split when $\beta = 4$. The top and bottom tables are for the time step $\Delta t = 1/4, 1/16$, respectively. The reference values are computed by the pmmLang + split with $\Delta t = 1/64$.

No. of Particles	pmmLang + split (%)	w/RBM, $p = 2$ (%)	w/RBM, $p = 4$ (%)
8	0.20	1.43	0.39
16	0.11	3.23	1.16
24	0.14	5.42	2.40
32	0.12	7.27	2.84
No. of Particles	pmmLang + split (%)	w/RBM, $p = 2$ (%)	w/RBM, $p = 4$ (%)
8	0.08	0.46	0.10
16	0.03	1.40	0.41
24	0.06	2.10	0.82
32	0.03	3.32	0.85

TABLE VIII. Cost of CPU time (second) in one single time step of the pmmLang + split and the pmmLang + RBM + split.

No. of Particles	pmmLang + split	w/RBM, $p = 2$	w/RBM, $p = 4$
8	5.7×10^{-3}	1.6×10^{-3}	3.2×10^{-3}
16	2.5×10^{-2}	2.9×10^{-3}	6.1×10^{-3}
24	5.8×10^{-2}	4.0×10^{-3}	8.6×10^{-3}
32	1.1×10^{-1}	5.9×10^{-3}	1.1×10^{-2}

that the error estimation (50) also holds for the pmmLang + RBM + split. That is to say, the bias of the time average computed by the pmmLang + RBM + split diminishes as the time step $\Delta t \to 0$ or the batch size p increases.

Finally, we report the time complexity of the pmmLang and the pmmLang + RBM in Table VIII. Comparing the results in Tables IV and VIII, we observe that the pmmLang + RBM + split is slightly slower than the pmmLang + RBM but is still much more efficient than the pmmLang + split. Therefore, the additional calculation of the potential difference $U_2(\boldsymbol{q}^*) - U_2(\boldsymbol{q})$ in the splitting Monte Carlo method does not increase the total computational cost too much.

VI. CONCLUSION

We have proposed the pmmLang + RBM, an efficient sampling method of the quantum interacting particle system in the PIMD framework. The pmmLang + RBM properly combines the preconditioned mass-modified Langevin dynamics (pmmLang) and the random batch method (RBM) to resolve the stiffness of the ring polymer and reduce the complexity due to interaction forces. In the pmmLang + RBM, the computational cost due to interaction forces in a time step is reduced from $O(NP^2)$ to $O(NP)$, where N is the number of beads in the ring polymer and P is the number of particles.

In the extensive numerical tests, the pmmLang + RBM shows fine performance in the calculation of the thermal average. The pmmLang + RBM shares a similar convergence mechanism with the original pmmLang and results in a small bias from the target distribution even for large N and P. Nevertheless, a rigorous error estimation of the pmmLang + RBM remains to be studied further.

Under the circumstances of singular interacting potentials (e.g., the Lennard-Jones potential), we introduce the pmmLang + split and the pmmLang + RBM + split to avoid using extremely small time steps. When the pmmLang + split has a low rejection rate, the pmmLang + RBM + split greatly reduces the computational cost per time step, keeps small error in computing the thermal average, and does not increase the rejection rate too much. It will be valuable to explore how to control the rejection rate when N or P is large.

APPENDIX A: DISCUSSION ON THE PRECONDITIONING METHODS

1. Design of the preconditioning methods

In this section, we briefly introduce the theory of the preconditioning methods in the PIMD, which aim to resolve the stiffness in the ring polymer. As we have shown in Sec. II, the stiffness originates from the stiffness matrix $L \in \mathbb{R}^{N \times N}$, which is defined by

$$
L = \frac{m}{\beta_N^2}
\begin{bmatrix}
2 & -1 & & \cdots & & & -1 \\
-1 & 2 & -1 & \cdots & & & \\
 & -1 & 2 & \cdots & & & \\
\vdots & \vdots & \vdots & \ddots & -1 & & \\
 & & & & -1 & 2 & -1 \\
-1 & & & & & -1 & 2
\end{bmatrix}.
$$

Now, let us focus on the Hamiltonian dynamics of the ring polymer system,

$$
\begin{aligned}
\mathrm{d}\boldsymbol{q} &= -M^{-1}\boldsymbol{p}\,\mathrm{d}t, \\
\mathrm{d}\boldsymbol{p} &= -L\boldsymbol{q}\,\mathrm{d}t,
\end{aligned}
\tag{A1}
$$

where $M \in \mathbb{R}^{N \times N}$ is the positive definite mass matrix.

The first step of preconditioning is to use a coordinate transformation $\boldsymbol{q} = D\tilde{\boldsymbol{q}}$ to decompose (A1) into different modes. To be specific, using the transformation

$$
\tilde{\boldsymbol{q}} = D^{-1}\boldsymbol{q}, \quad \tilde{\boldsymbol{p}} = D^{\mathrm{T}}\boldsymbol{p},
\tag{A2}
$$

we can rewrite (A1) as

$$
\begin{aligned}
\mathrm{d}\tilde{\boldsymbol{q}} &= -(D^{\mathrm{T}}MD)^{-1}\tilde{\boldsymbol{p}}\,\mathrm{d}t, \\
\mathrm{d}\tilde{\boldsymbol{p}} &= -(D^{\mathrm{T}}LD)\tilde{\boldsymbol{q}}\,\mathrm{d}t,
\end{aligned}
\tag{A3}
$$

where we require $D^{\mathrm{T}}MD$ and $D^{\mathrm{T}}LD$ to be both diagonal. In this, a preconditioning method in the PIMD is designed via the following steps:

1. Find a transformation matrix $D \in \mathbb{R}^{N \times N}$ such that

$$
D^{\mathrm{T}}LD = \mathrm{diag}\{\lambda_1, \ldots, \lambda_N\}.
\tag{A4}
$$

2. For some suitable constants $\mu_1, \ldots, \mu_N > 0$, choose the mass matrix

$$
M = D^{-\mathrm{T}}\mathrm{diag}\{\mu_1, \ldots, \mu_N\}D^{-1}.
\tag{A5}
$$

The transformed Hamiltonian dynamics (A3) is then decomposed into different modes, where the frequency of the kth mode is

$$
\omega_k = \sqrt{\frac{\lambda_k}{\mu_k}}, \quad k = 1, \ldots, N,
\tag{A6}
$$

05 November 2025 21:36:51

and the stiffness is resolved if the frequencies ω_k are uniform for $k = 1, \ldots, N$.

In the following, we introduce two specific preconditioning methods, the staging coordinates and the preconditioned mass-modified Langevin dynamics (pmmLang).

2. Staging coordinates

The staging coordinate transformation is given by

$$\tilde{q}_1 = q_1, \quad \tilde{q}_k = q_k - \frac{(k-1)q_{k+1} + q_1}{k}, \quad k = 2, \ldots, N, \quad (A7)$$

which directly yields the transformation matrix D. Using this transformation, the ring polymer potential becomes

$$\sum_{k=1}^{N} |q_k - q_{k+1}|^2 = \sum_{k=1}^{N} m_k |\tilde{q}_k|^2, \quad (A8)$$

where $m_1 = 0$ and $m_k = \frac{k}{k-1}$ for $k = 2, \ldots, N$. Therefore, $L \in \mathbb{R}^{N \times N}$ is diagonalized as in (A4), where

$$\lambda_1 = 0, \quad \lambda_k = \frac{k}{k-1} \cdot \frac{m}{\beta_N^2}, \quad k = 2, \ldots, N. \quad (A9)$$

In the staging coordinate method, the constants $\mu_1, \ldots, \mu_N$ are chosen to be

$$\mu_1 = m, \quad \mu_k = \frac{k}{k-1} m, \quad k = 1, \ldots, N. \quad (A10)$$

Thus, the mass matrix $M \in \mathbb{R}^{N \times N}$ is given by

$$M = \beta_N^2 L + m D^{-T} E_{11} D, \quad (A11)$$

where $E_{11} \in \mathbb{R}^{N \times N}$ is the matrix with only $(1,1)$ entry equal to 1. If one attempts to use the staging coordinates in the physical coordinates q, p, then it will be a tough task to deal with the algebra of M in (A11).

3. Preconditioned mass-modified Langevin dynamics

The pmmLang is based on the spectral decomposition of L, and the corresponding transformation matrix D is orthogonal. For simplicity, assume the number of beads N is even. Then, $L \in \mathbb{R}^{N \times N}$ is orthogonally diagonalized as in (A4), where

$$\lambda_1 = 0 : D_{j,1} = \frac{1}{\sqrt{N}},$$

$$\lambda_N = \frac{4m}{\beta_N^2} : D_{j,N} = \frac{(-1)^j}{\sqrt{N}},$$

$$\lambda_{2k} = \frac{4m}{\beta_N^2} \sin^2 \frac{\pi k}{N} : D_{j,2k} = \sqrt{\frac{2}{N}} \cos \frac{2\pi kj}{N},$$

$$\lambda_{2k+1} = \frac{4m}{\beta_N^2} \sin^2 \frac{\pi k}{N} : D_{j,2k+1} = \sqrt{\frac{2}{N}} \sin \frac{2\pi kj}{N}$$

$$\left(j = 1, \ldots, N; k = 1, \ldots, \frac{N}{2} - 1 \right).$$

The constants $\mu_1, \ldots, \mu_N$ are simply chosen to be

$$\mu_k = \lambda_k + \alpha, \quad k = 1, \ldots, N, \quad (A12)$$

where the regularization parameter $\alpha > 0$ is a fixed constant and the corresponding mass matrix is $M = L^\alpha := L + \alpha I$.

The advantage of the pmmLang is that it is simple and can be directly applied in the spatial coordinates q. On the contrary, the mass matrix of the staging coordinates is complicated in the spatial coordinates q and is inconvenient to combine with the RBM; hence, we choose the pmmLang as the preconditioning method in the interacting particle system.

Finally, we discuss the complexity due to algebraic operations in the pmmLang. In the BAOAB scheme in a time step, one needs to solve the linear system

$$(L^\alpha)^{-1} \nabla U^\alpha(q) \quad (A13)$$

and compute the matrix-vector multiplication

$$(L^\alpha)^{-\frac{1}{2}} \xi, \quad \xi \sim N(0,1)^{N \times 3P} \quad (A14)$$

to obtain the Gaussian random variable $\eta \sim N(0, (L^\alpha)^{-1})$. Since L^α is tridiagonal, the complexity of the linear system is $O(NP)$. With the use of the spectrum of L given above, the matrix-vector multiplication $(L^\alpha)^{-\frac{1}{2}} \xi$ can be calculated by the fast Fourier transform, and the complexity is $O(N \log NP)$. In conclusion, the complexity due to algebraic operations in the pmmLang in a time step is $O(N \log NP)$.

APPENDIX B: SPLITTING MONTE CARLO METHOD

In this section, we establish the detailed balance for the splitting Monte Carlo method and prove that the corresponding second-order Langevin dynamics preserves the desired Boltzmann distribution. To simplify our arguments, consider the target Boltzmann distribution

$$\pi(q, v) = \exp\left(-\beta\left(\frac{1}{2}\langle v, Mv \rangle + U(q)\right)\right), \quad (B1)$$

where $q, v \in \mathbb{R}^d$ is the position and velocity, $M \in \mathbb{R}^{d \times d}$ is the positive definite mass matrix, and $\beta > 0$ is the inverse temperature.

The Langevin dynamics that preserves the distribution $\pi(q, v)$ is given by

$$dq = v dt,$$
$$dv = -M^{-1} \nabla U(q) dt - \gamma v dt + \sqrt{\frac{2\gamma M^{-1}}{\beta}} dB, \quad (B2)$$

where B is the standard Brownian motion in $\mathbb{R}^d$ and $\gamma > 0$ is the friction constant. The detailed balance for (B2) reads as follows:[49]

Theorem B.1. *Let $T((q, v), (q', v'))$ be the transition probability density of the Langevin dynamics (B2) in time t, and then the detailed balance holds*

$$\pi(q, v) T((q, v), (q', v')) = \pi(q', -v') T((q', -v'), (q, -v)), \quad (B3)$$

which implies that $\pi(q,v)$ is the invariant distribution of (B2),

$$\int_{\mathbb{R}^{2d}} \pi(q,v)T((q,v),(q',v'))\mathrm{d}q\mathrm{d}v = \pi(q',v'). \qquad (B4)$$

When the potential function $U(q)$ is singular, we split it into

$$U(q) = U_1(q) + U_2(q), \qquad (B5)$$

where $U_1(q)$ is smooth and $U_2(q)$ is short-ranged and singular. Then, we obtain the splitting Monte Carlo method (Algorithm 4), which preserves the Boltzmann distribution $\pi(q,v)$ and avoids the gradient of the singular part of $U_1(q)$. The detailed balance of this method is given in the following theorem.

Theorem B.2. *Let $T((q,v),(q',v'))$ be the transition probability density of Algorithm 4, and then the detailed balance holds*

$$\pi(q,v)T((q,v),(q',v')) = \pi(q',-v')T((q',-v'),(q,-v)), \quad (B6)$$

which implies that $\pi(q,v)$ is the invariant distribution of the splitting Monte Carlo method (Algorithm 4),

$$\int_{\mathbb{R}^{2d}} \pi(q,v)T((q,v),(q',v'))\mathrm{d}q\mathrm{d}v = \pi(q',v'). \qquad (B7)$$

Proof. Define the distributions

$$\pi_1(q,v) = \exp\left(-\beta\left(\frac{1}{2}\langle v, Mv\rangle + U_1(q)\right)\right), \qquad (B8)$$

$$\pi_2(q) = \exp(-\beta U_2(q)), \qquad (B9)$$

and then the target distribution $\pi(q,v) = \pi_1(q,v)\pi_2(q)$. Define the acceptance probability

$$a(q,q^*) = \min\{1, e^{-\beta(U_2(q^*)-U_2(q))}\}, \qquad (B10)$$

and then $\pi_2(q)$ satisfies

$$\pi_2(q)a(q,q') = \pi_2(q')a(q',q) \qquad (B11)$$

with the acceptance probability $a(\cdot,\cdot)$ defined above. Let $T_1((q,v),(q',v'))$ be the transition probability density of the Langevin dynamics in a time step Δt,

$$\mathrm{d}q = v\mathrm{d}t,$$
$$\mathrm{d}v = -M^{-1}\nabla U_1(q)\mathrm{d}t - \gamma v\mathrm{d}t + \sqrt{\frac{2\gamma}{\beta}}M^{-\frac{1}{2}}\mathrm{d}B, \qquad (B12)$$

and then from the detailed balance (63), we obtain

$$\pi_1((q,v),(q',v'))T_1((q,v),(q',v'))$$
$$= \pi_1(q',-v')T_1((q',-v'),(q,-v)). \qquad (B13)$$

Note that the transition probability density of Algorithm 4 is

$$T((q,v),(q',v')) = T_1((q,v),(q',v'))a(q,q')$$
$$+ \delta(q'-q)\delta(v'+v)(1-A(q,v)), \qquad (B14)$$

where $A(q,v)$ is the average acceptance probability at (q,v),

$$A(q,v) = \int_{\mathbb{R}^{2d}} T_1((q,v),(q',v')). \qquad (B15)$$

To prove the detailed balance (B6), we just need to verify

$$\pi(q,v)T_1((q,v),(q',v'))a(q,q')$$
$$= \pi(q',-v')T_1((q',-v'),(q,-v))a(q',q) \qquad (B16)$$

and

$$\pi(q,v)\delta(q'-q)\delta(v'+v)(1-A(q,v))$$
$$= \pi(q',-v')\delta(q'-q)\delta(v'+v)(1-A(q',-v')). \qquad (B17)$$

In fact, (B16) is the product of (B11) and (B14), and (B17) holds for $q = q', v = -v'$. Hence, Theorem B.2 is proved.

Theorem B.2 directly applies to the pmmLang (35) with the splitting scheme $U^\alpha(\boldsymbol{q}) = U_1(\boldsymbol{q}) + U_2(\boldsymbol{q})$, yielding theorem IV.1 within the PIMD framework.

Algorithm 4. Splitting Monte Carlo method for the Langevin dynamics in a time step Δt.

Evolve the Langevin dynamics in a timestep Δt:

$$\mathrm{d}q = v\mathrm{d}t$$
$$\mathrm{d}v = -M^{-1}\nabla U_1(q)\mathrm{d}t - \gamma v\mathrm{d}t + \sqrt{\frac{2\gamma}{\beta}}M^{-\frac{1}{2}}\mathrm{d}B$$

Let (q^*,v^*) be the proposal calculated above. Set $(q,v) = (q^*,v^*)$ with probability

$$a(q,q^*) = \min\{1, e^{-\beta(U_2(q^*)-U_2(q))}\}$$

otherwise set $(q,v) = (q,-v)$.

DATA AVAILABILITY

The codes and data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] R. P. Feynman, A. R. Hibbs, and D. F. Styer, *Quantum Mechanics and Path Integrals* (Courier Corporation, 2010).

[2] W. H. Miller, "Path integral representation of the reaction rate constant in quantum mechanical transition state theory," J. Chem. Phys. **63**, 1166–1172 (1975).

[3] G. A. Voth, "Feynman path integral formulation of quantum mechanical transition-state theory," J. Phys. Chem. **97**, 8365–8377 (1993).

[4] B. J. Berne and D. Thirumalai, "On the simulation of quantum systems: Path integral methods," Annu. Rev. Phys. Chem. **37**, 401–424 (1986).

[5] J. A. Barker, "A quantum-statistical Monte Carlo method; path integrals with boundary conditions," J. Chem. Phys. **70**, 2914–2918 (1979).

[6] D. Chandler and P. G. Wolynes, "Exploiting the isomorphism between quantum theory and classical statistical mechanics of polyatomic fluids," J. Chem. Phys. **74**, 4078–4095 (1981).

[7] M. Parrinello and A. Rahman, "Study of an *F* center in molten KCl," J. Chem. Phys. **80**, 860–867 (1984).

[8] M. Ceriotti, M. Parrinello, T. E. Markland, and D. E. Manolopoulos, "Efficient stochastic thermostatting of path integral molecular dynamics," J. Chem. Phys. **133**, 124104 (2010).

[9] M. E. Tuckerman, D. Marx, M. L. Klein, and M. Parrinello, "Efficient and general algorithms for path integral Car–Parrinello molecular dynamics," J. Chem. Phys. **104**, 5579–5588 (1996).

[10] I. R. Craig and D. E. Manolopoulos, "Chemical reaction rates from ring polymer molecular dynamics," J. Chem. Phys. **122**, 084106 (2005).

[11] I. R. Craig and D. E. Manolopoulos, "A refined ring polymer molecular dynamics theory of chemical reaction rates," J. Chem. Phys. **123**, 034102 (2005).

[12] X. Tao, P. Shushkov, and T. F. Miller III, "Microcanonical rates from ring-polymer molecular dynamics: Direct-shooting, stationary-phase, and maximum-entropy approaches," J. Chem. Phys. **152**, 124117 (2020).

[13] I. R. Craig and D. E. Manolopoulos, "Quantum statistics and classical mechanics: Real time correlation functions from ring polymer molecular dynamics," J. Chem. Phys. **121**, 3368–3373 (2004).

[14] R. Korol, N. Bou-Rabee, and T. F. Miller III, "Cayley modification for strongly stable path-integral and ring-polymer molecular dynamics," J. Chem. Phys. **151**, 124103 (2019).

[15] E. Mátyus, D. J. Wales, and S. C. Althorpe, "Quantum tunneling splittings from path-integral molecular dynamics," J. Chem. Phys. **144**, 114108 (2016).

[16] C. L. Vaillant, D. J. Wales, and S. C. Althorpe, "Tunneling splittings from path-integral molecular dynamics using a Langevin thermostat," J. Chem. Phys. **148**, 234102 (2018).

[17] S. Toxvaerd and J. C. Dyre, "Communication: Shifted forces in molecular dynamics," J. Chem. Phys. **134**, 081102 (2011).

[18] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, 2017).

[19] W. Mattson and B. M. Rice, "Near-neighbor calculations using a modified cell-linked list method," Comput. Phys. Commun. **119**, 135–148 (1999).

[20] P. P. Ewald, "Ewald summation," Ann. Phys. **369**, 253 (1921).

[21] T. Darden, D. York, and L. Pedersen, "Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems," J. Chem. Phys. **98**, 10089–10092 (1993).

[22] B. K. Alpert and V. Rokhlin, "A fast algorithm for the evaluation of Legendre expansions," SIAM J. Sci. Stat. Comput. **12**, 158–179 (1991).

[23] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, "Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions," Mol. Simul. **6**, 121–142 (1991).

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

[25] 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**, A1486–A1509 (2020).

[26] L. Li, J.-G. Liu, and Y. Tang, "A direct simulation approach for the Poisson–Boltzmann equation using the random batch method," arXiv:2004.05614 (2020).

[27] A. Beskos, F. J. Pinski, J. M. Sanz-Serna, and A. M. Stuart, "Hybrid Monte Carlo on Hilbert spaces," Stochastic Processes Appl. **121**, 2201–2230 (2011).

[28] J. Lu and Z. Zhou, "Continuum limit and preconditioned Langevin sampling of the path integral molecular dynamics," arXiv:1811.10995 (2018).

[29] M. E. Tuckerman, B. J. Berne, G. J. Martyna, and M. L. Klein, "Efficient molecular dynamics and hybrid Monte Carlo algorithms for path integrals," J. Chem. Phys. **99**, 2796–2808 (1993).

[30] J. Liu, D. Li, and X. Liu, "A simple and accurate algorithm for path integral molecular dynamics with the Langevin thermostat," J. Chem. Phys. **145**, 024103 (2016).

[31] R. Korol, J. L. Rosa-Raíces, N. Bou-Rabee, and T. F. Miller III, "Dimension-free path-integral molecular dynamics without preconditioning," J. Chem. Phys. **152**, 104102 (2020).

[32] T. E. Markland and D. E. Manolopoulos, "An efficient ring polymer contraction scheme for imaginary time path integral simulations," J. Chem. Phys. **129**, 024105 (2008).

[33] J. L. Rosa-Raíces, J. Sun, N. Bou-Rabee, and T. F. Miller III, "A generalized class of strongly stable and dimension-free T-RPMD integrators," J. Chem. Phys. **154**, 024106 (2021).

[34] M. F. Herman, E. J. Bruskin, and B. J. Berne, "On path integral Monte Carlo simulations," J. Chem. Phys. **76**, 5150–5155 (1982).

[35] H. C. Andersen, "Molecular dynamics simulations at constant pressure and/or temperature," J. Chem. Phys. **72**, 2384–2393 (1980).

[36] G. J. Martyna, M. L. Klein, and M. Tuckerman, "Nosé–Hoover chains: The canonical ensemble via continuous dynamics," J. Chem. Phys. **97**, 2635–2643 (1992).

[37] N. Bou-Rabee and J. M. Sanz-Serna, "Geometric integrators and the Hamiltonian Monte Carlo method," Acta Numer. **27**, 113–206 (2018).

[38] N. Bou-Rabee and A. Eberle, "Two-scale coupling for preconditioned Hamiltonian Monte Carlo in infinite dimensions," Stochastics Partial Differ. Equations: Anal. Comput. **9**, 207–242 (2021).

[39] R. A. Fisher and F. Yates, *Statistical Tables: For Biological, Agricultural and Medical Research* (Oliver & Boyd, 1938).

[40] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of the COMPSTAT'2010* (Springer, 2010), pp. 177–186.

[41] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv:1609.04747 (2016).

[42] L. Bottou, "Stochastic gradient descent tricks," *Neural Networks: Tricks of the Trade* (Springer, 2012), pp. 421–436.

[43] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2014), pp. 661–670.

[44] S. Jin, L. Li, and J.-G. Liu, "Convergence of random batch method for interacting particles with disparate species and weights," arXiv:2003.11257 (2020).

[45] E. Weinan, T. Li, and E. Vanden-Eijnden, *Applied Stochastic Analysis* (American Mathematical Society, 2019), Vol. 199.

[46] A. Wehrl, "General properties of entropy," Rev. Mod. Phys. **50**, 221 (1978).

[47] S. Kullback, *Information Theory and Statistics* (Courier Corporation, 1997).

[48] J. Besag, "Comments on 'representations of knowledge in complex systems' by U. Grenander and M. I. Miller," J. R. Stat. Soc.: Ser. B **56**, 591–592 (1994).

[49] G. Bussi and M. Parrinello, "Accurate sampling using Langevin dynamics," Phys. Rev. E **75**, 056707 (2007).

[50] B. Leimkuhler and C. Matthews, "Robust and efficient configurational molecular sampling via Langevin dynamics," J. Chem. Phys. **138**, 174102 (2013).

[51] B. Leimkuhler and C. Matthews, *Molecular Dynamics* (Springer, 2016).