Prefetch parallelization and optimization of Monte Carlo in the grand canonical, isothermal-isobaric and Gibbs ensemble

Harold W. Hatch[1, a)]
Chemical Informatics Research Group, Chemical Sciences Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8380, USA

(Dated: 4 May 2026)

Parallelization of Monte Carlo (MC) is required to observe the same growth as molecular dynamics because computer processor clock speeds have plateaued while the number of cores has increased. Although prefetch parallelization can speed up an MC molecular simulation by a factor of 3 using four parallel threads for simultaneous single-particle displacements in the canonical ensemble, other ensembles require multiple trial types that impact efficiency when threads wait for the other threads with more time-consuming trials, such as volume changes or particle insertions and deletions in the isothermal-isobaric, grand canonical and Gibbs ensemble. Load balancing increases efficiency by attempting the same trial in each thread of a parallel batch, but violates detailed balance if done incorrectly. By computing standard deviations as a function of processor time, efficiency is systematically investigated over a variety of ensembles, load balancing algorithms and trial attempt and acceptance probabilities for dense liquids of Lennard-Jones and an extended simple point charge model of water, to reveal numerous efficiency gains, including in serial simulations. Parallel efficiency in these ensembles approached the theoretical maximum by reducing overhead costs with improved algorithms and data structures released in the open-source MC software called FEASST.

I. INTRODUCTION

Molecular simulation algorithms must be parallelized to benefit from the exponential growth in computing power and to enable studies of larger length and time scales.[1] In the 20th century, because central processing unit (CPU) clock rates increased exponentially, waiting for the release of a faster processor before beginning a non-stop, multi-year calculation could be more prudent than immediately beginning on a slower processor.[2] However, in the 21st century, CPU clock rates plateaued and CPUs instead gained more cores, due to fundamental limits[3] such as propagation delay[4] and heat dissipation.[5] Because a serial simulation may take nearly as long as it did two decades ago, rapid scientific progress requires algorithmic improvements, including parallelization.

Despite the effectiveness of Monte Carlo (MC) molecular simulations to study applications in colloidal assembly,[6–10] phase equilibrium,[11–13] coarse-grained protein models,[14–17] second virial coefficients,[18,19] and adsorption,[20–23] molecular dynamics (MD) simulations are more commonly used than MC.[24] Large-scale molecular simulations[25] typically use MD because changes in the positions of the whole system, instead of single-particles, enables scaling to more processors with domain decomposition parallelization.[26] To bridge this gap, parallelizable MC algorithms need to be developed, such as configurational bias,[27–30] waste-recycling,[31] graphical processing unit acceleration[32,33] and prefetching.[34–36] This article is the first study of parallel prefetching MC simulations in the grand canonical (μVT), isothermal-isobaric (NPT) and Gibbs ensemble.[37]

MC prefetching attempts simultaneous trials on different parallel threads and reconstructs the serial simulation by synchronization of the threads, as illustrated in Fig. 1. In comparison to parallel configurational bias,[28,30] prefetching does not require computing the Rosenbluth factor for the old states, and prefetching can be used together with parallel or dual-cut configurational bias.[38] In a previous study of MC prefetching in the canonical (NVT) ensemble,[36] stored random numbers and energies avoided recalculating energies during synchronization. In this article, a more efficient approach synchronizes data structures with object-oriented interfaces, and is amenable to more complex energy calculations that utilize Ewald sums[39] and cell and neighbor lists.[40]

Quantification of efficiency is required to benchmark new algorithms, but is hindered by non-physical MC trials relative to MD. In standard MD, efficiency may be measured by diffusion or steps per CPU time, but MC requires different metrics for trials with variable CPU times and microstate changes.[36,41,42] Although MC trial acceptance is a useful metric for optimization,[43,44] it does not account for trial CPU time or improved sampling (e.g., the generation of less-correlated microstates). Tracking the positions or properties of single particles with CPU time to compute efficiency is not applicable when MC trials include non-physical insertion and deletion of particles and volume (V) and position scaling. In this article, efficiency was defined by the standard deviation of the mean of a statistical quantity as a function of CPU time.[45–47] This efficiency metric accounts for improved sampling of microstates (e.g., to benchmark trials with lower acceptance but greater changes in microstate) and implementation (e.g., overhead and synchronization).

Due to the difficulty of computing efficiency in MC,[2,36,41,42,45,48] few previous studies offer guidelines for optimal choices of parameters, such as attempt probabil-

a)Electronic mail: harold.hatch@nist.gov

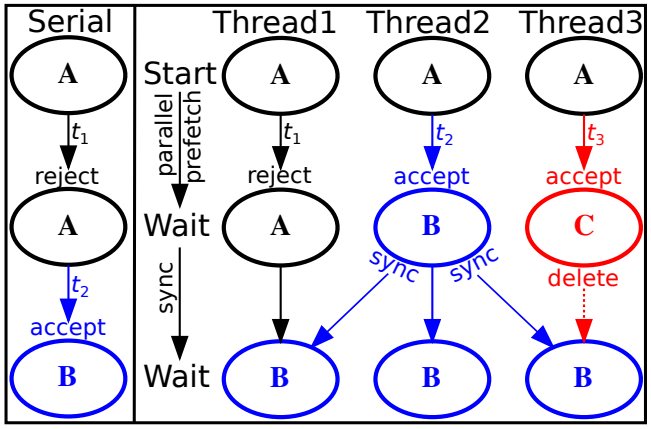

FIG. 1. In the box on the left, an example serial MC simulation begins at microstate A and attempts a rejected trial, t_1, then a second accepted trial into microstate B, t_2, shown in blue. In the box on the right, an identical Markov chain in a parallel MC prefetch batch of three threads ($n_{thread} = 3$) begins by prefetching microstates with three simultaneous and parallel trial attempts. All threads wait for other threads to finish prefetching, and efficiency decreases with increasing variation of CPU time among different trials. Load balancing may reduce this overhead wait time by attempting trials with similar computational cost, but may violate detailed balance in some cases. Finally, synchronization (or sync) communicates the first accepted trial (if any) to other threads, further increasing overhead compared to the serial simulation. Any trial after the first accepted must be deleted, shown in red for microstate C, which is another source of inefficiency.

ity and desired acceptance. In the NVT ensemble, many simulations follow the original guidance of Metropolis *et al.*[49] for 50 % acceptance of particle displacements, which may not be optimal (or not possible at low density with whole-domain displacements[47]). Complexity increases in the NPT, μVT and Gibbs ensemble because different trial types may be attempted with different probabilities. Furthermore, the optimal difference between NVT trial attempt probabilities in the vapor and liquid of the Gibbs ensemble is unknown. For example, two separate processors could simultaneously displace particles in vapor and liquid Gibbs ensemble configurations,[50,51] dedicating roughly half of the CPU time to sampling the vapor. No study, to my knowledge, has quantified efficiency in the Gibbs ensemble to show if the liquid would benefit from more sampling than the vapor.

This article considers relatively small scale systems and parallelization up to 4 threads, which enables the optimization of many adjustable parameters. Unlike domain decomposition, which is typically used in MD, prefetch parallelization scaling, as implemented in this work, does not depend on the size of the system. Prefetching is amenable to cases where the interaction cutoff distance is approximately half the periodic boundary length, as typically encountered in phase behavior calculations. Prefetch parallelization across more than 4 threads is demonstrated in Ref. 36. Furthermore, this article lays the groundwork for parallelization of MC over more threads, because Prefetch parallelization can be combined with many other parallelization methods.

This article systematically investigates efficiency in the NVT, μVT, NPT and Gibbs ensemble for serial and prefetch-parallelized simulations on 2 and 4 threads with the Lennard-Jones (LJ) model and the extended simple point charge (SPC/E)[52] model of water, as described in Section II. In Section III A, the NVT ensemble is revisited to validate and compare with different efficiency metrics used in previous studies. In Section III B, various load balancing algorithms are explored in μVT simulations with different types of trials, and different relative trial attempt probabilities. Optimal NPT volume change acceptance and attempt probability are found in Section III C. In Section III D, Gibbs ensemble simulations optimize the relative size and trial attempt probability of the vapor phase compared to the liquid, including up to 20 % speedup of serial Gibbs simulations. Finally, Section IV summarizes the recommended parameter choices for serial and parallel prefetching MC simulations.

II. METHODS

All MC simulation software, run scripts and analysis scripts used in this article are available in the open-source FEASST v0.25.15 software,[53] with all 4 ensembles found in the first 4 tutorials of the "prefetch" plugin. Benchmark simulations of a bulk, single-component fluid of single-site LJ or rigid SPC/E[52] water molecules were performed in cubic periodic boundary conditions of length L with N particles and constant temperature, $T = \frac{1}{k_B\beta}$, where k_B is the Boltzmann constant and β is the inverse temperature. For the LJ fluid, the potential was cutoff at a distance of $r_c = 3\sigma$, where σ is the diameter. Long-range corrections were assumed to have a radial distribution function of 1 beyond r_c. LJ simulations were performed at $k_BT/\epsilon = 0.88$, which is near the triple point for a density of $\rho\sigma^3 = 0.85$ and were used in previous studies of efficiency.[36,42] For the SPC/E fluid, $r_c = 10$ Å in the liquid state, or 40 % of the initial L in the vapor configuration of Gibbs simulations, following previous recommendations.[45] SPC/E simulations were performed at a temperature of $T = 300$ K. Simulations were terminated after 5 days, rather than a fixed number of trials, to avoid idle CPU cores, which could speed-up unfinished simulations (using HPC nodes with dual Intel® Xeon® Silver 4216 CPUs with a total of 32 processors per node at 2.10 GHz base frequency, see the disclaimer at the end of Sec. VI). Error bars were obtained by computing the standard deviation of the mean from a sample of 4 independent simulations. Cycles were defined as 10^4 MC trials, regardless of N.

Parallel simulations were performed on 2 or 4 threads. Synchronization from a rejected thread simply updates trial counts and ensemble averages, while synchronization from an accepted thread uses stored energies and

random numbers of the accepted thread. Synchronization is related to copying and imitation referred to in Ref. 36. To synchronize data that are not updated due to skipped energy calculations, such as Ewald k-space vectors,[39] each base class contains a data object (SynchronizeData in FEASST) that derived classes use to store persistent data. A further optimization is that only data associated with particles that changed during the MC trial were synchronized.

NVT simulations. Particles were translated by random uniform generation of positive or negative displacements up to a maximum absolute displacement in each dimension. The desired acceptance, p_{NVT}, was approximately obtained by tuning the maximum absolute displacement[54,55] by 25 % every 10^3 trials during 10^2 cycles of equilibration. For SPC/E, rotational displacements were tuned and performed with equal probability as translations. NVT trials refer to translations for LJ and translations and rotations for SPC/E (e.g., p_{NVT} included rotations for SPC/E). For LJ, $L = 8\sigma$ and $N = 512$, and for SPC/E, $L = 20$ Å and $N = 267$.

μVT simulations. Single particle insertion and deletion were attempted with equal probability subject to a fixed chemical potential, μ,[56] and together were attempted with a probability proportional to $w_{\mu VT}$. Specifically, the probability to attempt a trial of type t,

$$p_t = \frac{w_t}{\sum_i^{trials} w_i} \tag{1}$$

was determined by the trial weight, w_t, where $w_{\mu VT}$ is defined as the weight of particle insertions and deletions. The weight of NVT trials, $w_{NVT} = 1$, was constant for all simulations. For LJ, $L = 8\sigma$, and for SPC/E, $L = 20$ Å and the liquid was initialized with $N = 275$ molecules. For LJ, $\beta\mu = -2.837$ was obtained by reweighting a flat histogram simulation[57] with a density of $\rho\sigma^3 = 0.85$ to match the NVT liquid simulation, as described in the tenth tutorial of the "flat_histogram" plugin of FEASST.[53] For SPC/E, $\beta\mu = -15.265$ was similarly obtained near the equilibrium liquid density at $T = 300$ K. Inserted SPC/E particles were randomly oriented. All μVT simulations were equilibrated for 5×10^3 cycles.

NPT simulations. Volume changes were attempted with a weight of w_{NPT} by random uniform generation of positive or negative displacements in $\ln V$ up to a maximum absolute displacement. The desired acceptance, p_{NPT}, was approximately obtained by tuning the maximum absolute value of the change in $\ln V$ by 25 % during 5×10^3 cycles of equilibration. Because NPT volume changes are performed infrequently, the maximum tunable value was updated every 20 NPT trials during equilibration (as well as for the Gibbs ensemble). Particle coordinates were scaled during volume changes. For LJ, $N = 400$ and for SPC/E, $N = 512$ and intramolecular bonds were not scaled. For LJ, the pressure, $P\sigma^3/\epsilon = 1.227$ was obtained by reweighting flat histogram simulations[57] as described above for a density of $\rho\sigma^3 = 0.85$, and for SPC/E, $p = 0.01$ kJ/mol/Å$^3 \approx 16.6$ MPa was well above the saturation pressure at $T = 300$ K.[58]

Single-component Gibbs simulations. Particles and volume were transferred between two systems with a fixed total number of particles, N, volume, V and temperature.[37] In both LJ and SPC/E, $N = 600$. The amount of volume transfer was chosen uniformly in V. While the NVT trials in the liquid system were performed with a weight of $w_{NVT} = 1$, the NVT trials in the vapor system were performed with a weight of $w_{NVT,vapor}$. The fraction of the number of particles in the vapor, N_{vapor}, is given by $f_{vapor} = N_{vapor}/N$. The initial volumes were based on the expected densities reported in the Standard Reference Simulation Website (SRSW)[58] of approximately $\rho\sigma^3 \approx 0.01237$ and 0.76297 for the LJ fluid vapor and liquid densities, respectively, and $\rho\sigma^3 \approx 0.007373$ and 998.1 kg/m^3, for the SPC/E fluid vapor and liquid densities, respectively. Gibbs simulations were equilibrated with 2×10^3 cycles.

III. RESULTS AND DISCUSSION

The following subsections are divided into various ensembles. First, the NVT ensemble validated the efficiency metrics that optimize the tunable acceptance probability of an NVT trial, p_{NVT}, in agreement with previous studies. The μVT ensemble illustrated parallel efficiency issues when trials take different CPU times, and issues arising from attempting to load balance parallel batches with the same trial type. The μVT ensemble also requires the optimization of the trial attempt weight, $w_{\mu VT}$, relative to an NVT trial, $w_{NVT} = 1$. Next, the tunable acceptance probability of NPT trials, p_{NVT} as well as the trial attempt weight, w_{NPT}, were optimized in the NPT ensemble. Finally, these parameters were used in parallel prefetch Gibbs ensemble simulations to validate the previously optimized[45] fraction of particles in the vapor, and further optimize the probability to attempt NVT trials in the vapor, $w_{NVT,vapor}$ relative to NVT trials in the liquid.

The main quantity reported in this article is the efficiency, z,[47] which is the ratio of the unnormalized z' (with arbitrary units) given by

$$z' = 1/(t\sigma^2) = \exp(-2b), \tag{2}$$

where t is the CPU time, σ is the standard deviation of the mean and b is the ordinate intercept obtained from a log-log linear fit of the equilibrated standard deviation of the mean as a function of CPU time (e.g., z' is the squared inverse of the "difficulty index" as defined by Schultz and Kofke).[45,46] Standard deviations of the mean were computed with the blocking method[59] as a function of CPU time.[47] All z reported in this article are non-dimensional and normalized with respect to the most efficient simulation. The ratio of the efficiency of

simulation 1 relative to simulation 2 is given by,[47]

$$z = \frac{z_1'(\sigma_1 = \sigma)}{z_2'(\sigma_2 = \sigma)} = \frac{t_2(\sigma_2 = \sigma)}{t_1(\sigma_1 = \sigma)} = \exp[2(b_2 - b_1)]. \quad (3)$$

For example, if simulation 2 obtains the same standard deviation as simulation 1 in half the time, then the efficiency of simulation 1 with respect to simulation 2 is 1/2. In this article, the serial simulation with the maximal efficiency was used as the reference simulation 2, and all other simulation efficiencies were reported with respect to the maximally efficient serial simulation for a given ensemble, model and thermodynamic condition (e.g., $z \leq 1$). An efficiency metric based on the potential energy, U, is labeled z_U, while efficiency based on the number of particles or volume is z_N or z_V, respectively. For $z_U = 0.75$ with 4 threads, the same level of energy convergence was reached $4z_U = 3$ times faster than the serial simulation (with four times the cost), whereas $z = 1$ is perfect parallel scaling. The efficiency, z, also compares the efficiency of different parameter values in serial simulations.

A. Canonical (NVT) Ensemble

The efficiency of NVT ensemble LJ and SPC/E simulations based on the convergence of the energy, z_U, is shown in Fig. 2 for serial, 2 and 4 thread simulations over a range of NVT acceptance probabilities, p_{NVT}. To highlight the approximate maximum in z_U, results were least-squared fit to a second order polynomial. The efficiency was normalized to one at the maximum of the polynomial fit to the serial simulation. Due to the statistical variation in the simulation results, some z_U data points are greater than 1 because they are higher than the polynomial fit, which was chosen for normalization. To increase clarity in the subsequent figures, only the fits may be shown, but the simulation data points that were fit can be found in the Supplementary Material.

A previous prefetching study[36] found the optimal p_{NVT} for the serial LJ simulation to be approximately $p_{NVT} = 0.25$, with $p_{NVT} = 0.225$ for 2 threads and $p_{NVT} = 0.2$ for 4 threads. The results for the SPC/E fluid in Fig. 2 roughly agree with this trend. The LJ fluid results are shifted to slightly higher values, possibly due to higher relative overhead in simulations of simple models when the energy calculation does not take the overwhelming majority of the CPU time (as opposed to a more complex SPC/E model or $r_c = 5\sigma$ in Ref. 36). In this article, relative overhead is defined as the ratio of the CPU time spent on not computing the energy vs total CPU time.

Another possible reason why z_U is slightly different than the previous diffusion efficiency metric[36] is because diffusion-based metrics may prioritize particle movement over sampling. For example, swapping the positions of two identical particles does not change the energy but will affect the tracking of particle positions. Regardless

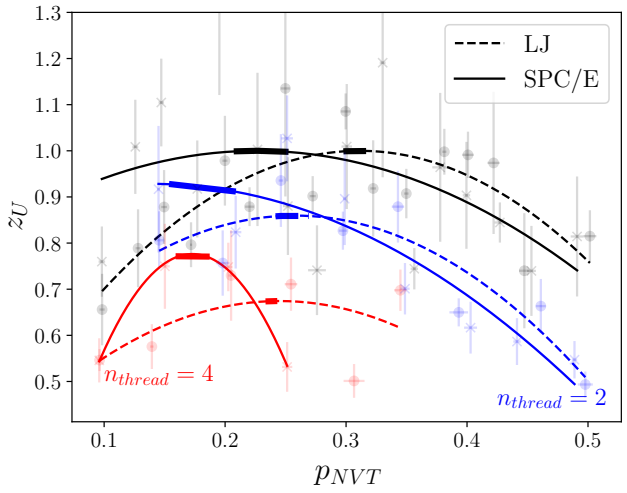

FIG. 2. The efficiency of NVT ensemble energy convergence, z_U, as a function of the translation trial acceptance probability, p_{NVT}, parallelized with 1 (black), 2 (blue) or 4 (red) threads for the SPC/E (solid lines and x's) and LJ (dashed lines and circles) liquids. Error bars in all figures are the standard deviation of the mean obtained from 4 independent simulations. Lines are fits of the data to second-order polynomials. The thick lines show the average location of the maximum in the fit (plus or minus the standard deviation from 4 independent fits).

of the possible reasons for the small differences in p_{NVT}, the differences are within the error bars in Fig. 2 and Ref. 36, which validates the use of z_U and other standard-deviation-based efficiency metrics in the remaining ensembles where diffusion-based metrics are not valid because of the non-physical movement from particle insertions, deletions and volume changes.

The SPC/E fluid obtains higher efficiencies than the LJ fluid (e.g., three times speedup with four cores for $z_U \approx 0.75$). This is likely due to the difference in the relative overhead of SPC/E vs LJ and is consistent with the trends observed previously for decreasing relative overhead with increasing r_c in the LJ fluid.[36] The observed efficiency of SPC/E was enabled by the synchronization data structure implementation described in Section II. The SPC/E fluid also prefers lower p_{NVT} than the LJ fluid, possibly because trial acceptance may be more expensive when Ewald vectors must be updated.

B. Grand Canonical (μVT) Ensemble and Load Balancing

In the μVT ensemble, the particle insertion and deletion trials utilize different CPU times than the NVT trials, because NVT trials compute the energy of the new and the old configuration, while insertion computes only the new configuration energy and deletion computes only the old configuration energy. Even if the old configuration energy is obtained through an optimized method,

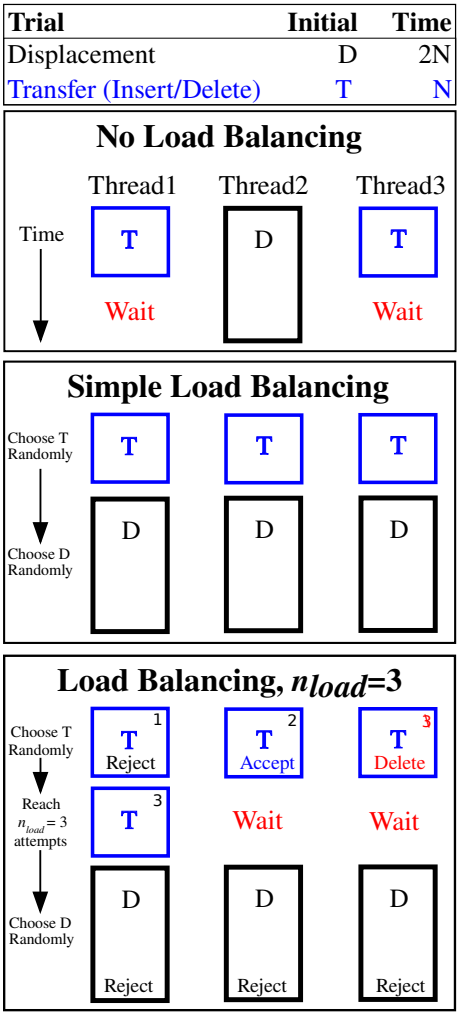

FIG. 3. (No load balancing) A μVT MC simulation where transfer trials (T, insert/delete) take half the time compared to displacement trials (D) loses efficiency when some threads wait for trials of all types to complete. (Simple load balancing) Using the same type of trial in a batch may reduce wait times but complicates detailed balance. (Load balancing, $n_{load} = 3$) Fixing the number of attempted trials of a given type in a batch introduces additional inefficiencies.

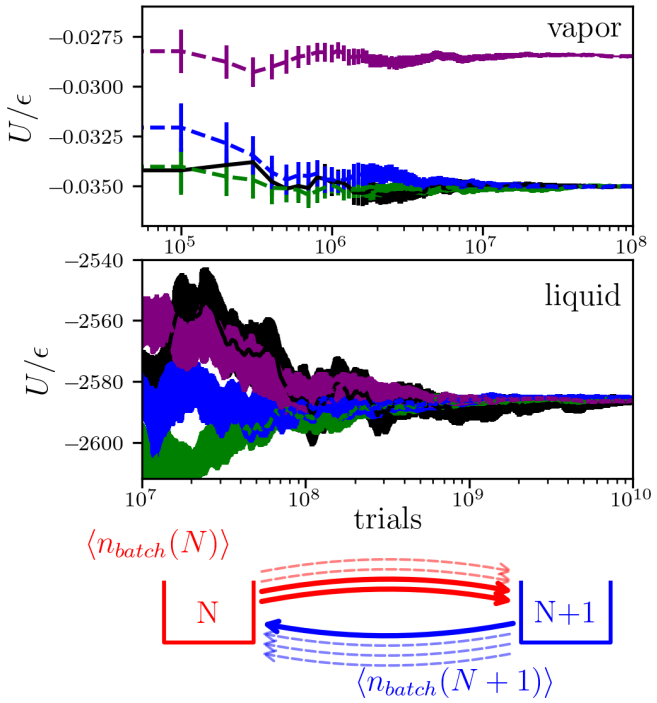

FIG. 4. (top) The average potential energy of an LJ μVT simulation of vapor as a function of the number of MC trials with $\beta\mu = -6$, $L = 8\sigma$ and $k_BT/\epsilon = 0.88$. Serial simulations (black solid line), parallel, $n_{thread} = 4$, simulations without load balancing (green) $n_{thread} = 4$ with simple load balancing (purple) and $n_{thread} = 4$ with load balancing that fixes the number of attempted trials, $n_{load} = 10$ (blue) are shown. Error bars are the block standard deviation of the mean. (middle) The same as above except for a liquid with $\beta\mu = -2.837$. (bottom) The number of attempted trials in one parallel batch, n_{batch} (not including deleted trials) depends on the trial acceptance.

such as stored per-particle energies,[60] the μVT and NPT trials take different CPU times. This leads to inefficiencies in the parallel prefetching method shown in Fig. 3 if the parallel threads are assigned completely random trial types, because the threads with faster trials wait for the threads with slower trials to finish. NPT volume changes may further increase overhead compared to μVT trials due to the expensive energy calculation, as discussed in Section III C.

To overcome the efficiency issue of parallel batches of trials that utilize different CPU times, a simple load balance algorithm fixes each parallel batch of trials with the same type of trial, as illustrated in Fig. 3. In the special case of asymmetric trials such as insertion and deletion, Fig. 4 shows that this simple load balancing algorithm violates detailed balance if the acceptance probabilities for the forward and reverse asymmetric trials are significantly different. At the top of Fig. 4, the average energy of an LJ vapor with simple load balancing converges to the wrong value when the trial acceptances between insertion and deletion are different. This issue was only observable beyond simulation error bars in μVT trials with very high acceptance rates (i.e., vapor), and was not observed in NVT, NPT or liquid μVT simulations (shown in the middle of Fig. 4). This is because the acceptance of the trial factors into the detailed balance equation, and the standard Metropolis acceptance is recovered when the acceptance of the forward and reverse trials is approximately equivalent.

Simple load balancing violates detailed balance if the number of attempted trials depends on the trial acceptance, as illustrated in the bottom of Fig. 4. Consider a trivial two-state simulation with zero or one ideal gas particles in the μVT ensemble where the probability to accept a deletion is 100 % but the probability to accept an insertion is 50 % due to the chosen μ (e.g., $N = 0$ in the bottom of Fig. 4). A simple load-balanced par-

Forward	$\alpha_{o\to n}$
Choose insert	$C\langle n_{batch}^{ins}(N)\rangle$
Choose position in V	$\mathrm{d}\mathbf{r}/V$
Choose orientation	$P_{\omega n}\,\mathrm{d}\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose delete	$C\langle n_{batch}^{del}(N+1)\rangle$
Choose particle of type i	$1/(N+1)$

TABLE I. Parallel prefetching particle insertion transition probabilities when all trials in a parallel batch are of the same type (i.e., simple load balancing), and the probability to choose the type of trial performed in a batch is proportional to the trial attempt weight.

allel deletion batch with $n_{thread} = 4$ performed at the $N = 1$ state will always attempt only one deletion trial per batch because the first deletion trial is guaranteed to be accepted, and the remaining trials in the batch are deleted and treated as if they were never attempted (i.e., $\langle n_{batch}(N+1)\rangle = 1$). Deleted trials are not the same as rejected trials. Deleted trials are treated as if they were never attempted, which is why $n_{batch} \leq n_{thread}$. On the other hand, an insertion batch will attempt an average number of trials per batch[61] of $\langle n_{batch}(N)\rangle = 1*0.5 + 2*0.25 + 3*0.125 + 4*0.125 = 1.875$ in this case, and, in general, $\langle n_{batch}(N)\rangle \neq \langle n_{batch}(N+1)\rangle$.

Metropolis acceptance criteria for simple load-balanced insertions and deletions may be explicitly derived by accounting for the average number of attempted insertion or deletion trials in a batch, $\langle n_{batch}^{ins}(N)\rangle$ and $\langle n_{batch}^{del}(N)\rangle$ at a given number of particles, N, using the checklist approach for deriving MC trials outlined in Ref. 55. The first step in the trial is to choose insertion or deletion. In typical serial simulations, this choice is made with equal probability. For simple load balancing, the acceptance probabilities may vary from states N and $N+1$, and the probability to choose an insertion in the example two-state system is $C\langle n_{batch}^{ins}(N)\rangle$, where C is a proportionality constant, while the probability to choose a deletion in the reverse trial is $C\langle n_{batch}^{del}(N+1)\rangle$. Insertions then choose a position randomly in the system volume with probability $\mathrm{d}\mathbf{r}/V$ and a random orientation with probability $P_{\omega n}\,\mathrm{d}\boldsymbol{\omega}$. In the reverse deletion trial, a random particle is chosen with probability $1/(N+1)$, where N was defined before the insertion. The transition probabilities are summarized in Table I.

The Metropolis acceptance, χ, of the above simple load balancing particle insertion is given by the ratio of the μVT microstate probabilities of the new and old states, times the ratio of the reverse and forward transition probabilities,[55]

$$\begin{aligned}\chi &= \frac{\alpha_{n\to o}}{\alpha_{o\to n}} z\,\mathrm{d}\mathbf{r}\,\mathrm{d}\boldsymbol{\omega} e^{-\beta\Delta U} \\ &= \frac{\langle n_{batch}^{del}(N+1)\rangle V z}{\langle n_{batch}^{ins}(N)\rangle (N+1) P_{\omega n}} e^{-\beta\Delta U},\end{aligned} \tag{4}$$

where ΔU is the change in energy from the old to the new microstate. By analogy, the simple load balancing deletion acceptance is

$$\chi = \frac{\langle n_{batch}^{ins}(N-1)\rangle N P_{\omega n}}{\langle n_{batch}^{del}(N)\rangle V z} e^{-\beta\Delta U}. \tag{5}$$

At low density and high acceptance, such as $\beta\mu = -6$ in the LJ fluid at $\beta\epsilon = 1/0.88$ shown at the top of Fig. 4, Eqs. 4 and 5 can deviate significantly from the serial simulation ($n_{batch} = 1$). Furthermore, the acceptance of inserting or deleting a particle at N or $N+1$ states may deviate when N is small. At high density and low acceptance, $n_{batch}^{del}(N+1)/n_{batch}^{ins}(N)$ approaches 1 and the properties computed from the simple load balancing algorithm with standard Metropolis acceptance criteria are within statistical noise of the serial simulation. In this case, the acceptance of inserting or deleting a particle is similar between N and $N+1$ states when the density is high and acceptances are low. Explicit use of Eq. 4 and 5 for the Metropolis acceptance requires the additional complexity of obtaining converged trial acceptances for a given number of particles in a serial simulation or on-the-fly in parallel simulations (not shown in this article). On the other hand, simple load balancing could use the standard Metropolis acceptance for insertion and deletion, provided post-processing verifies that the acceptance probabilities for insertion and deletion are within the error bars (as performed in this article).

A less efficient approach to load balance without introducing modifications to the Metropolis acceptance criteria is to fix the number of trial attempts for a given trial type, n_{load}, as shown in Fig. 3. The number of trial attempts for a given trial type, n_{load} should be equal to or greater than the number of threads, n_{thread}. When a new batch of trials begins, if the number of trials attempted in the batch, $n_{batch} < n_{load}$, due to a trial acceptance or large n_{load}, then more batches are performed until the sum of the number of (non-deleted) trials in each batch is exactly equal to n_{load}, at which point, a new type of trial is chosen and a new load of batches begins again. This means that some threads may not be utilized in the last batch, which is why this approach is less efficient than simple load balancing that has only one parallel batch per random choice of trial type. The acceptance for a fixed n_{load} does not depend on $\langle n_{batch}\rangle$ because the probability of attempting a type of trial does not depend on the acceptance. The top and middle of Fig. 4 show that fixing n_{load} yields the correct potential energy, even when the acceptance is high. In this article, the same n_{load} was used for each type of trial.

For load balancing, trial types should include both forward and reverse trial (e.g., insertion and deletion together are defined as one type of trial for the purposes of load balancing). Detailed balance is not obeyed for the n_{load} algorithm if one trial type is defined for insertion, and a different trial type is defined for deletion. In that case, a batch may contain only insertion trials, or only deletion trials. If the number of insertion trials is fixed to

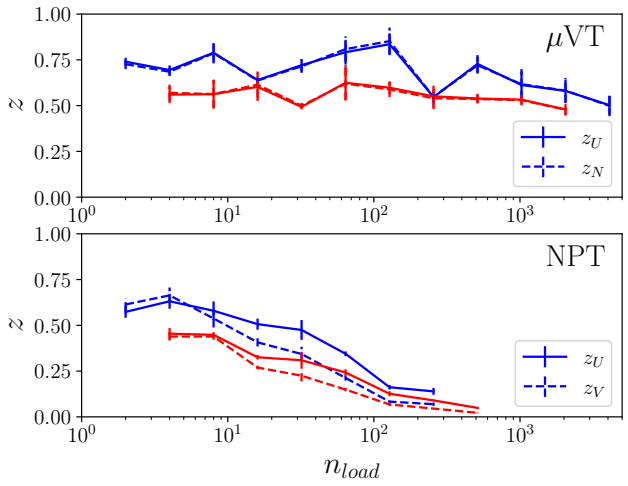

FIG. 5. (Top) The efficiency of 2 threads (blue) and 4 threads (red), relative to serial, for μVT simulations of an LJ liquid at $\beta\mu = -2.837$, $V = (8\sigma)^3$ and $k_BT/\epsilon = 0.88$ with a fixed number of attempted MC trials per trial type, n_{load}. (Bottom) For NPT simulations of an LJ liquid, $N = 400$, $p\sigma^3/\epsilon = 1.227$.

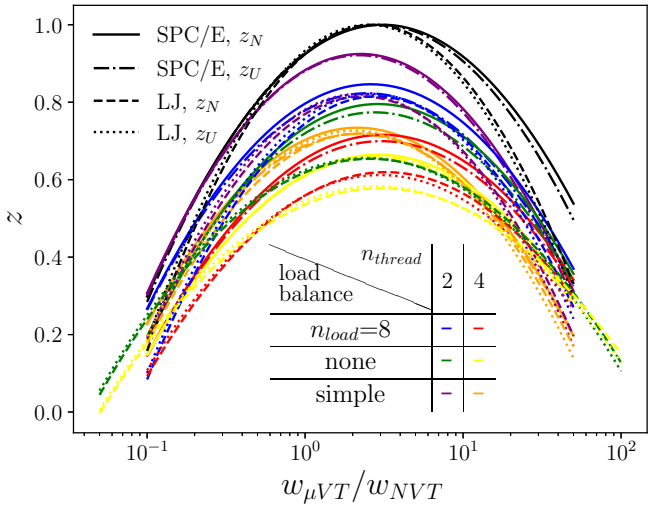

FIG. 6. The efficiency of μVT simulations as a function of the insertion and deletion trial attempt weight, $w_{\mu VT}$ (see Eq. 1), relative to NVT, parallelized with $n_{thread} = 1$, 2 or 4 threads for the U and N convergence efficiency of SPC/E and LJ liquids comparing simple load balancing, $n_{load} = 8$ or no load balancing. Lines are fits as described in Fig. 2.

be n_{load} and multiple insertion trials are accepted, then the first insertion trial cannot be immediately reversed with a deletion trial. Results for load balancing insertion and deletion trials separately demonstrate significant errors in the low density and high acceptance probability case, shown in the alternative Fig. 4 in the Supplemental Material in Section V. However, if the acceptance of the insertion or deletion is low, then defining the insertion and deletion trials separately will not affect the results significantly.

The top half of Fig. 5 shows that z_U and z_N are relatively insensitive to n_{load} in the μVT ensemble. Even small values of n_{load} do not show drastic efficiency decreases within the error bars, and large values of $n_{load} = 10^3$ only slightly drop efficiency. Because of the NPT ensemble results discussed in Sec. **??**, a value of $n_{load} = 8$ was chosen for the value of n_{load} in the remainder of this article.

A smaller n_{load} may reduce efficiency because every other parallel batch could have unutilized threads. In Fig. 3, if the second thread in a batch of 3 threads has an accepted trial, then $n_{batch} = 2$, and if the smallest possible $n_{load} = 3$ was used, then the next batch will only use one of the 3 threads to attempt the required third trial. A larger n_{load} would ensure that at least n_{load}/n_{thread} batches, rounded down to the nearest integer, are performed before threads may go unutilized. However, if $n_{load} \gg n_{thread}$, then many of the same trial types may be performed before any other trial type, and this could reduce sampling efficiency. For example, if the current microstate can not incorporate another particle, insertion trials will never be accepted regardless of the number of attempts. Attempting more insertion trials will not improve sampling, and the microstate needs to be changed by a different kind of trial (e.g., an NVT displacement trial). Similarly, volume contraction attempts will always be rejected if the excluded volumes of two particles touch. Extremely large values of n_{load} are not recommended to avoid sampling many times with the same trial type.

Fig. 6 shows the efficiency of μVT simulations as a function of the relative weight for attempting an insertion or deletion, $w_{\mu VT}/w_{NVT}$ using various load balancing algorithms. The results of Fig. 6 shows that $w_{\mu VT}/w_{NVT} = 2$ is an optimal choice for both the LJ and SPC/E liquids. One possible explanation for an optimal $w_{\mu VT}/w_{NVT} = 2$ is that insertions and deletions cost roughly half the CPU time of NVT displacements. The observed z_U and z_N are within error bars. As in the NVT ensemble results, the SPC/E model again has more efficiency than LJ, likely because of smaller relative overhead.

To compare the efficiency of the various load balancing algorithms in the μVT ensemble shown in Fig. 6, the simple load balancing algorithm (purple and orange in Fig. 6) is more efficient than the $n_{load} = 8$ load balancing algorithm (blue and red). This was expected because fixing the number of trial attempts for a given type of trial leads to idle threads. In addition, both load balancing algorithms are more efficient than not load balancing (green and yellow in Fig. 6).

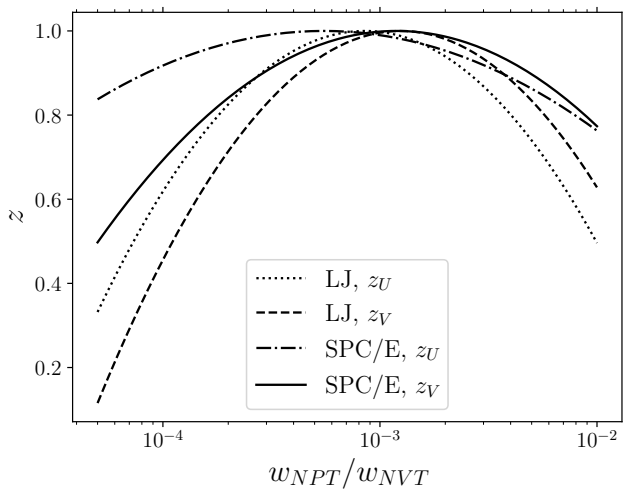

FIG. 7. The efficiency of NPT simulations as a function of the volume change trial weight, w_{NPT} (see Eq. 1), relative to NVT, for the SPC/E and LJ fluids with $p_{NVT} = 0.5$.

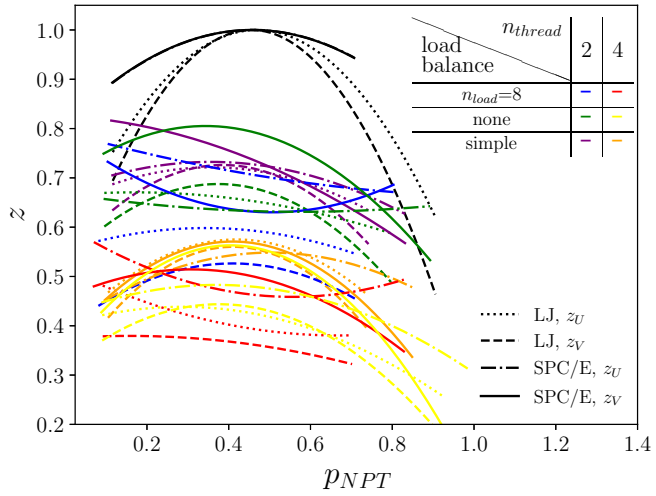

FIG. 8. The U and V convergence efficiency of NPT simulations as a function of the volume change trial acceptance probability, p_{NPT}, for the SPC/E and LJ fluids with various load balancing algorithms and $w_{NPT}/w_{NVT} = 10^{-3}$.

C. Isothermal Isobaric (NPT) Ensemble

Now that the parameters involved in particle displacements, insertions and deletions have been optimized, the NPT ensemble is used to optimize the volume change trial acceptance, p_{NPT} and attempt probability weight, w_{NPT} (also required in the Gibbs ensemble). A $p_{NPT} = 0.5$ was used while optimizing w_{NPT} as shown in Fig. 7. Only the serial simulation is considered in this case because $w_{\mu VT}$ did not depend on n_{thread}. Overall, the optimal value was found to be $w_{NPT}/w_{NVT} \approx 10^{-3} \approx 1/N$, which is similar to one volume change attempt per particle displacement sweep. The LJ and SPC/E fluids yield identical optimal w_{NPT}, but z_V slightly favors larger values of w_{NPT} than z_U. Compared to $w_{\mu VT}/w_{NVT} = 2$, the optimal weight of the trial attempt is approximately inversely proportional to the CPU time required to implement the trial. If each trial type contributes equally to sampling different degrees of freedom (e.g., positions for NVT trials, N for μVT trials and V for NPT trials), then the most optimal choice may be to allocate equal CPU time to each trial type. Allocation of CPU time between trial types that sample the same degrees of freedom is beyond the scope of this article (e.g., different variations of configurational-bias for NVT trials).

Before finding the optimal p_{NPT}, the value of $n_{load} = 8$ was chosen based on the results shown in the bottom half of Fig. 5. Although the μVT ensemble efficiency is relatively insensitive to n_{load}, the efficiency in the NPT ensemble decreases significantly with increasing n_{load}. One possible explanation is that a batch of volume change trials that start from the same microstate differ only in the scale of the change, and are highly correlated, as opposed to single-particle trials that randomly choose particles or new positions. If volume scaling trials are more correlated with one another, then a batch of volume change trials may not be as effective at sampling microstates compared to a batch of single-particle trials.

Fig. 8 shows that $p_{NPT} \approx 0.5$ is optimal for the serial simulation and is roughly twice that of p_{NVT}. The LJ and SPC/E fluids agree on this optimum, as well as z_U and z_V. In the parallel case, p_{NPT} shifts to lower values of ≈ 0.45 for 2 threads and ≈ 0.4 for 4 threads, which again correspond roughly to twice that of p_{NVT}. The differences between the optimal p_{NVT} and p_{NPT} may be related to differences in implementation because volume changes require computing the energy of the entire configuration, whereas volume changes do not require recomputing the old configuration energy.

As opposed to the μVT ensemble, the NPT ensemble shows the highest parallel efficiency without load balancing (green and yellow in Fig. 8). Thus, the parallel efficiency in the NPT ensemble does not reach as high as in the μVT ensemble. This was an unexpected result because the NPT volume change trials take significantly more CPU time to compute, and load balancing was designed to reduce the time that the threads wait. Fortunately, the optimal NPT trial weight, $w_{NPT}/w_{NVT} \approx 10^{-3}$ reduces the number of batches with NPT trials, such that waiting only happens infrequently, and infrequent waiting is apparently better than batches of correlated volume changes. A possibly promising parallelization strategy would be to utilize domain decomposition parallelization for the energy calculation of a single NPT trial instead of prefetching, which is beyond the scope of this article. Optimized volume changes for single-site LJ using the virial were also not implemented in this article.

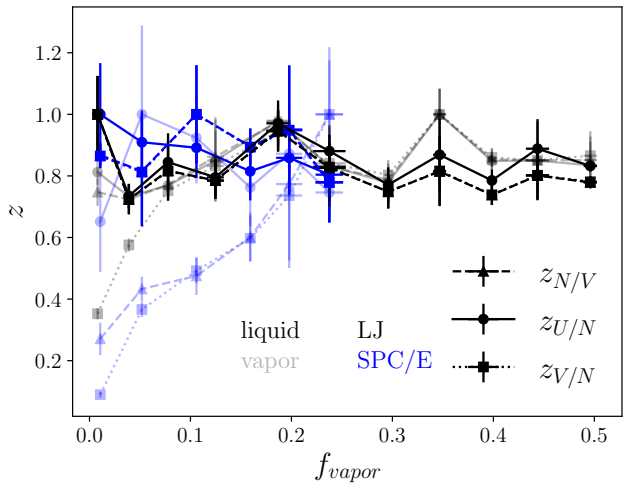

FIG. 9. The efficiency of serial Gibbs simulations as a function of the fraction of particles in the vapor, f_{vapor}, of LJ (black) and SPC/E (blue) in the liquid (opaque) and vapor (translucent). Error bars are standard deviations of the mean from 4 independent simulations.

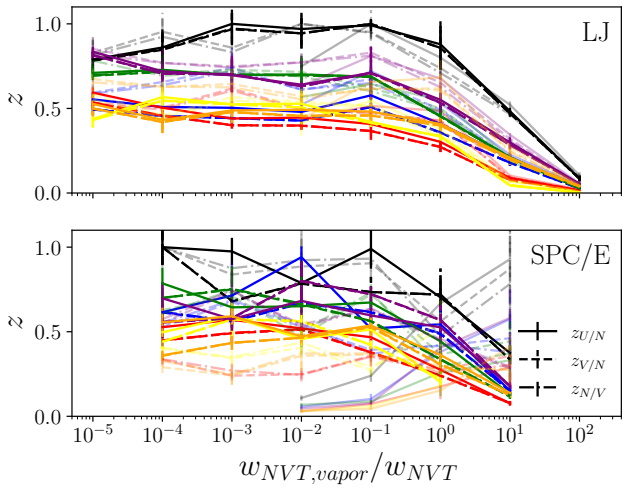

FIG. 10. (Top) The efficiency of Gibbs ensemble simulations of an LJ fluid at $\beta\epsilon = 1/0.88$ as a function of the vapor displacement trial weight, $w_{NVT,vapor}/w_{NVT}$. The number of threads and load balancing algorithms are described in Fig. 6. The translucent lines are for the vapor phase, $z_{U/N}$ (solid), $z_{V/N}$ (dashed), $z_{N/V}$ (dash dot) and $f_{vapor} \approx 0.2$. Error bars are standard deviations of the mean from four independent simulations. (Bottom) For the SPC/E fluid at $T = 300$ K.

D. Gibbs Ensemble

In this section, the Gibbs ensemble utilizes the optimized parameters associated with NVT, NPT and μVT trials. In the case of single component Gibbs with two configurations, the remaining parameters are the fraction of the total number of particles in the vapor configuration, f_{vapor}, and the weight to attempt NVT trials in the vapor, $w_{NVT,vapor}$. For serial simulations, an optimal value of $f_{vapor} = 0.2$ was reported previously,[45] and is further validated in Fig. 9. The efficiency drops at very low numbers of particles in the vapor, then reaches a plateau around $f_{vapor} = 0.2$. In many Gibbs ensemble simulations, the size of the liquid configuration is minimally constrained by r_c which is typically greater than one or two solvation shells. The size of a liquid configuration typically dictates the majority of the computational cost of the entire Gibbs ensemble simulation, although Ewald summation in a large vapor can also become costly. Therefore, the smallest possible f_{vapor} that does not compromise the efficiency and sampling of the simulation is typically the optimal choice, $f_{vapor} = 0.2$, as reported previously.[45] An optimal $w_{\mu VT}/w_{NVT} = 2$ is also assumed because the vapor energy calculation is typically less than that of the liquid, so the optimal choice is likely to follow that of the liquid.

The author is not aware of any previous simulation study with a different particle displacement attempt probability in the vapor, $w_{NVT,vapor}$ than in the liquid, w_{NVT}. That is probably because the vapor trials are typically less expensive to compute than the liquid trials, and therefore little effort is spent to ensure the vapor is well equilibrated. However, in parallel prefetch simulations, NVT vapor displacements have a relatively high acceptance, and therefore reduce the parallel efficiency.[36,61] A lower $w_{NVT,vapor}$ could further increase parallel efficiency.

The top half of Fig. 10 shows that $w_{NVT,vapor}/w_{NVT} \approx 0.1$ is optimal for vapor-liquid equilibrium Gibbs ensemble simulation of the LJ fluid at $\beta\epsilon = 1/0.88$. No load balancing (green and yellow in Fig. 10) is about as efficient as simple load balancing (purple and orange), while $n_{load} = 8$ load balancing (blue and red) is the least efficient. When $w_{NVT,vapor} > 1$, efficiency decreases with increasing $w_{NVT,vapor}$, which seems counter intuitive because the vapor should be improving, but the sampling of the vapor appears to be coupled to that of the liquid, such that poor liquid sampling may also reduce vapor sampling.

Similar trends are also observed for the SPC/E fluid at $T = 300$ K, shown in the bottom half of Fig. 10, except that the vapor phase (translucent lines) is more difficult to sample than the LJ vapor, possibly due to the number of Ewald vectors required for large volumes at low density, and therefore the vapor benefits from larger values of $w_{NVT,vapor}$. Particle transfer acceptance probabilities were $\approx 1.1 \times 10^{-4}$ with $\approx 5.5 \times 10^{4}$ accepted transfers during the production phase. In the SPC/E case, balancing the efficiency of sampling in the vapor versus sampling in the liquid is challenging. Because the liquid possesses a more expensive energy calculation, the error bars of the liquid are often higher than the vapor. Accordingly, $w_{NVT,vapor}/w_{NVT} \approx 0.1$ appears to be a reasonable choice.

For the specific energy of the SPC/E vapor, data for

results below $w_{NVT,vapor}/w_{NVT} = 0.01$ were omitted for $z_{U/N}$ in the bottom half of Fig. 10 because the sampling in the vapor was poor enough that the ensemble average values were statistically different, as shown in the second and third figures in the supplementary material. The efficiency metric, z may only compare simulations that yield statistically equivalent ensemble averages. Otherwise, a change in the magnitude of the ensemble average may spuriously appear as a change in the efficiency. For example, if the magnitude of the average decreases due to poor sampling, and the standard deviation of the average also decreases, this can spuriously appear as an increase in efficiency.

IV. CONCLUSIONS

Prefetch parallelization was shown to be effective for Monte Carlo simulations in the canonical, grand canonical, isothermal-isobaric and Gibbs ensembles with an increase in speed up to a factor of 3 using 4 parallel threads. Systematic comparisons of the efficiency of serial and parallel simulations of the LJ and SPC/E models over a variety of trial acceptance probabilities, trial attempt weights and load balancing algorithms revealed the following rough guidelines. Single-particle translations and rotations should have maximum tunable parameters optimized for acceptances of 0.25 in serial,[42] 0.225 with 2 threads and 0.2 with 4 threads, while using 50 % acceptance reduces sampling by about half. Volume changes should have maximum tunable parameters optimized for acceptances of 0.5 in serial, 0.45 with 2 threads and 0.4 with 4 threads (i.e., twice that of displacements). Volume change trial attempt rates, relative to displacements, should be approximately inversely proportional to the number of particles (e.g., approximately one volume change per displacement of each particle to spend roughly equal CPU time on each type of trial). Similarly, insertions and deletions should be attempted twice as often as displacements, possibly because these trials take half the CPU time as displacements. If insertions, deletions or volume changes are attempted ten times as often as or less often than the above recommendations, then sampling is reduced by approximately a factor of 2. Approximately 20 % of the particles should be in the vapor phase of a Gibbs ensemble simulation.[45] Gibbs ensemble particle displacements in the vapor may be attempted approximately ten times less than those in the liquid, which results in up to 20 % additional efficiency for serial and parallel simulations.

In general, efficiencies computed from fits to standard deviations as a function of CPU time was shown to be an effective tool for optimizing simulation parameters with non-physical moves with the following caveat. The efficiency comparison must have a chosen quantity whose average is statistically equivalent to the reference simulation.

The optimal choice of load balancing algorithm depended on the ensemble. In the μVT ensemble, simple load balancing was most efficient choice. Although simple load balancing requires care to ensure detailed balance, either by directly utilizing Eq. 4 and 5, or verifying that the acceptances of insertion and deletion are roughly equal (typical with low acceptance). Prefetching is not an efficient choice when acceptances are high, which is also when simple load balancing presents issues. Surprisingly, in the NPT ensemble the most efficient choice was to not load balance at all, likely due to the correlation between volume change trials that start from the same microstate. In the Gibbs ensemble, the efficiency of simple and no load balancing were roughly comparable. In most cases, the fixed n_{load} load balancing algorithm was typically the least efficient choice, but this algorithm ensures detailed balance (as well as not load balancing).

A number of considerations remain for future investigations. The cause of the relatively poor performance of load balancing in the NPT and Gibbs ensemble compared to the μVT ensemble is uncertain, and alternative parallelization strategies for volume changes, such as domain decomposition, could be more effective. The previous recommendation of at least 0.1 accepted volume or particle transfer per cycle[45] was not systematically investigated because this guideline depends on the acceptance of the trial (e.g., temperature, density, etc), but this previous recommendation is not inconsistent with those made in this article.

V. SUPPLEMENTAL MATERIAL

The Supplemental Material contains (1) the file "sm.pdf" with additional figures and (2) the folder "fig_data" which includes figure data in comma-separated value format.

VI. ACKNOWLEDGEMENT

This article was funded by the National Institute of Standards and Technology and is not subject to U.S. Copyright. Certain commercial firms and trade names are identified to specify the usage procedures adequately for reproducibility. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that related products are necessarily the best available for the purpose. The author did not use artificial intelligence tools for the text, images or software created for this article. The author has no conflicts to disclose.

VII. REFERENCES

[1] P. Czarnul, *Parallel Programming for Modern High Performance Computing Systems* (Chapman and Hall/CRC, New York, 2018).

[2]D. Frenkel, "Simulations: The dark side," The European Physical Journal Plus **128**, 1–21 (2013).

[3]I. L. Markov, "Limits on fundamental limits to computation," Nature **512**, 147–154 (2014).

[4]J. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. Souri, K. Banerjee, K. Saraswat, A. Rahman, R. Reif, and J. Meindl, "Interconnect limits on gigascale integration (GSI) in the 21st century," Proceedings of the IEEE **89**, 305–324 (2001).

[5]R. S. Whitney, "Most Efficient Quantum Thermoelectric at Finite Power Output," Phys. Rev. Lett. **112**, 130601 (2014).

[6]A. Haji-Akbari, M. Engel, A. S. Keys, X. Zheng, R. G. Petschek, P. Palffy-Muhoray, and S. C. Glotzer, "Disordered, quasicrystalline and crystalline phases of densely packed tetrahedra," Nature **462**, 773–777 (2009).

[7]H. W. Hatch, W. P. Krekelberg, S. D. Hudson, and V. K. Shen, "Depletion-driven crystallization of cubic colloids sedimented on a surface," J. Chem. Phys. **144**, 194902 (2016).

[8]H. W. Hatch, S.-Y. Yang, J. Mittal, and V. K. Shen, "Self-assembly of trimer colloids: effect of shape and interaction range," Soft Matter **12**, 4170–4179 (2016).

[9]R. P. Murphy, H. W. Hatch, N. A. Mahynski, V. K. Shen, and N. J. Wagner, "Dynamic arrest of adhesive hard rod dispersions," Soft Matter **16**, 1279–1286 (2020).

[10]C.-C. G. Yeh, H. W. Hatch, A. N. Sreenivasan, B. Bharti, V. K. Shen, Z. M. Sherman, and T. M. Truskett, "Colloidal Monolayers with Short-Range Attractions and Dipolar Repulsions," J. Phys. Chem. B **129**, 6428–6438 (2025).

[11]K. S. Rane, S. Murali, and J. R. Errington, "Monte Carlo Simulation Methods for Computing Liquid–Vapor Saturation Properties of Model Systems," J. Chem. Theory Comput. **9**, 2552–2566 (2013).

[12]J. Xu, H. W. Hatch, V. K. Shen, and Z. Jin, "Phase Equilibria of CO2 and n-Alkanes in Bulk and Confined Space Using Parallelized Wang–Landau Transition-Matrix Monte Carlo Simulations," Energy & Fuels **39**, 7305–7313 (2025).

[13]A. Chremos, W. P. Krekelberg, H. W. Hatch, D. W. Siderius, N. A. Mahynski, and V. K. Shen, "Development of SAFT-Based Coarse-Grained Models of Carbon Dioxide and Nitrogen," J. Phys. Chem. B **129**, 3443–3453 (2025).

[14]C. Calero-Rubio, A. Saluja, and C. J. Roberts, "Coarse-Grained Antibody Models for "Weak" Protein–Protein Interactions from Low to High Concentrations," J. Phys. Chem. B **120**, 6592–6605 (2016).

[15]C. Calero-Rubio, A. Saluja, E. Sahin, and C. J. Roberts, "Predicting High-Concentration Interactions of Monoclonal Antibody Solutions: Comparison of Theoretical Approaches for Strongly Attractive Versus Repulsive Conditions," J. Phys. Chem. B **123**, 5709–5720 (2019).

[16]M. A. Blanco, H. W. Hatch, J. E. Curtis, and V. K. Shen, "Evaluating the Effects of Hinge Flexibility on the Solution Structure of Antibodies at Concentrated Conditions," J. Pharm. Sci. **108**, 1663–1674 (2019).

[17]H. W. Hatch, C. Bergonzo, M. A. Blanco, G. Yuan, S. Grudinin, M. Lund, J. E. Curtis, A. V. Grishaev, Y. Liu, and V. K. Shen, "Anisotropic coarse-grain Monte Carlo simulations of lysozyme, lactoferrin, and NISTmAb by precomputing atomistic models," J. Chem. Phys. **161**, 094113 (2024).

[18]A. J. Schultz and D. A. Kofke, "Virial equation of state as a new frontier for computational chemistry," J. Chem. Phys. **157**, 190901 (2022).

[19]Y. Liu, H. W. Hatch, G. Yuan, V. K. Shen, A. V. Grishaev, J. Panchal, and M. Blanco, "Extracting Orientation and Distance-Dependent Interaction Potentials between Proteins in Solutions Using Small-Angle X-ray/Neutron Scattering," J. Phys. Chem. Letters **15**, 12401–12407 (2024).

[20]H. Zhang and R. Q. Snurr, "Computational Study of Water Adsorption in the Hydrophobic Metal–Organic Framework ZIF-8: Adsorption Mechanism and Acceleration of the Simulations," J. Phys. Chem. C **121**, 24000–24010 (2017).

[21]D. W. Siderius, H. W. Hatch, and V. K. Shen, "Temperature Extrapolation of Henry's Law Constants and the Isosteric Heat of Adsorption," J. Phys. Chem. B **126**, 7999–8009 (2022).

[22]D. W. Siderius, H. W. Hatch, J. R. Errington, and V. K. Shen, "Comments on "Monte Carlo simulations for water adsorption in porous materials: Best practices and new insights"," AIChE J. **68**, e17686 (2022).

[23]D. W. Siderius, H. W. Hatch, and V. K. Shen, "Flat-Histogram Monte Carlo Simulation of Water Adsorption in Metal–Organic Frameworks," J. Phys. Chem. B **128**, 4830–4845 (2024).

[24]L. Talirz, L. M. Ghiringhelli, and B. Smit, "Trends in Atomistic Simulation Software Usage [Article v1.0]," Living J. Comput. Mol. Sci. **3**, 1483–1483 (2021).

[25]J. Jung, W. Nishima, M. Daniels, G. Bascom, C. Kobayashi, A. Adedoyin, M. Wall, A. Lappala, D. Phillips, W. Fischer, C.-S. Tung, T. Schlick, Y. Sugita, and K. Y. Sanbonmatsu, "Scaling molecular dynamics beyond 100,000 processor cores for large-scale biophysical simulations," J. Comput. Chem. **40**, 1919–1930 (2019).

[26]A.-P. Hynninen and M. F. Crowley, "New faster CHARMM molecular dynamics engine," J. Comput. Chem. **35** (2014), 10.1002/jcc.23501.

[27]J. I. Siepmann and D. Frenkel, "Configurational bias Monte Carlo: a new sampling scheme for flexible chains," Mol. Phys. **75**, 59–70 (1992).

[28]K. Esselink, L. D. J. C. Loyens, and B. Smit, "Parallel Monte Carlo simulations," Phys. Rev. E **51**, 1560–1568 (1995).

[29]L. Loyens, B. Smit, and K. Esselink, "Parallel Gibbs-ensemble simulations," Mol. Phys. **86**, 171–183 (1995).

[30]T. J. H. Vlugt, "Efficiency of Parallel CBMC Simulations," Mol. Simul. **23**, 63–78 (1999).

[31]D. Frenkel, "Speed-up of Monte Carlo simulations by sampling of rejected states," Proc. Natl. Acad. Sci. USA **101**, 17571–17575 (2004).

[32]J. Mick, E. Hailat, V. Russo, K. Rushaidat, L. Schwiebert, and J. Potoff, "GPU-accelerated Gibbs ensemble Monte Carlo simulations of Lennard-Jonesium," Comput. Phys. Commun. **184**, 2662–2669 (2013).

[33]Y. Nejahi, M. S. Barhaghi, G. Schwing, L. Schwiebert, and J. Potoff, "Update 2.70 to "GOMC: GPU Optimized Monte Carlo for the simulation of phase equilibria and physical properties of complex fluids"," SoftwareX **13**, 100627 (2021).

[34]A. E. Brockwell, "Parallel Markov chain Monte Carlo Simulation by Pre-Fetching," J. Comput. Graph. Statist. **15**, 246–261 (2006).

[35]B. Calderhead, "A general construction for parallelizing Metropolis-Hastings algorithms," Proc. Natl. Acad. Sci. USA **111**, 17408–17413 (2014).

[36]H. W. Hatch, "Parallel Prefetching for Canonical Ensemble Monte Carlo Simulations," J. Phys. Chem. A **124**, 7191–7198 (2020).

[37]A. Z. Panagiotopoulos, "Direct determination of phase coexistence properties of fluids by Monte Carlo simulation in a new ensemble," Mol. Phys. **61**, 813–826 (1987).

[38]T. J. H. Vlugt, M. G. Martin, B. Smit, J. I. Siepmann, and R. Krishna, "Improving the efficiency of the configurational-bias Monte Carlo algorithm," Mol. Phys. **94**, 727–733 (1998).

[39]P. P. Ewald, "Die Berechnung optischer und elektrostatischer Gitterpotentiale," Annalen der Physik **369**, 253–287 (1921).

[40]M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, USA, 1989).

[41]J. Kolafa, "On optimization of Monte Carlo simulations," Mol. Phys. **63**, 559–579 (1987).

[42]R. D. Mountain and D. Thirumalai, "Quantative measure of efficiency of Monte Carlo simulations," Physica A **210**, 453–460 (1994).

[43]A. Torres-Knoop, N. C. Burtch, A. Poursaeidesfahani, S. P. Balaji, R. Kools, F. X. Smit, K. S. Walton, T. J. Vlugt, and D. Dubbeldam, "Optimization of Particle Transfers in the Gibbs Ensemble for Systems with Strong and Directional Interactions Using CBMC, CFCMC, and CB/CFCMC," J. Phys. Chem. C

120, 9148–9159 (2016).
[44]P. Bai and J. I. Siepmann, "Assessment and Optimization of Configurational-Bias Monte Carlo Particle Swap Strategies for Simulations of Water in the Gibbs Ensemble," J. Chem. Theory Comput. **13**, 431–440 (2017).
[45]A. D. Cortés Morales, I. G. Economou, C. J. Peters, and J. Ilja Siepmann, "Influence of simulation protocols on the efficiency of Gibbs ensemble Monte Carlo simulations," Mol. Simul. **39**, 1135–1142 (2013).
[46]A. J. Schultz and D. A. Kofke, "Quantifying Computational Effort Required for Stochastic Averages," J. Chem. Theory Comput. **10**, 5229–5234 (2014).
[47]H. W. Hatch, D. W. Siderius, J. R. Errington, and V. K. Shen, "Efficiency Comparison of Single- and Multiple-Macrostate Grand Canonical Ensemble Transition-Matrix Monte Carlo Simulations," J. Phys. Chem. B **127**, 3041–3051 (2023).
[48]A. Gelman, W. R. Gilks, and G. O. Roberts, "Weak convergence and optimal scaling of random walk Metropolis algorithms," The Annals of Applied Probability **7**, 110–120 (1997).
[49]N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," J. Chem. Phys. **21**, 1087–1092 (1953).
[50]H. Chopra, "Parallel algorithms for the Gibbs ensemble Monte Carlo simulations," University of Tennessee Knoxville Masters Theses (1996).
[51]B. Stenqvist, A. Thuresson, A. Kurut, R. Vácha, and M. Lund, "Faunus – a flexible framework for Monte Carlo simulation," Mol. Simul. **39**, 1233–1239 (2013).
[52]H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma, "The missing term in effective pair potentials," J. Phys. Chem. **91**, 6269–6271 (1987).
[53]H. W. Hatch, D. W. Siderius, and V. K. Shen, "Monte Carlo molecular simulations with FEASST version 0.25.1," J. Chem. Phys. **161**, 092501 (2024).
[54]D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Academic Press, 2002).
[55]H. W. Hatch, D. S. Corti, D. A. Kofke, and V. K. Shen, "Best Practices for Developing Monte Carlo Methodologies in Molecular Simulations [Article v1.0]," Living J. Comput. Mol. Sci. **6**, 3289–3289 (2025).
[56]D. J. Adams, "Grand canonical ensemble Monte Carlo for a Lennard-Jones fluid," Mol. Phys. **29**, 307–311 (1975).
[57]J. R. Errington, "Direct calculation of liquid-vapor phase equilibria from transition matrix Monte Carlo simulation," J. Chem. Phys. **118**, 9915–9925 (2003).
[58]V. K. Shen, D. W. Siderius, W. P. Krekelberg, and H. W. Hatch, *NIST Standard Reference Simulation Website*, NIST Standard Reference Database Number 173 (National Institute of Standards and Technology, 2017).
[59]H. Flyvbjerg and H. G. Petersen, "Error Estimates on Averages of Correlated Data," J. Chem. Phys. **91**, 461–466 (1989).
[60]J. K. Shah, E. Marin-Rimoldi, R. G. Mullen, B. P. Keene, S. Khan, A. S. Paluch, N. Rai, L. L. Romanielo, T. W. Rosch, B. Yoo, and E. J. Maginn, "Cassandra: An open source Monte Carlo package for molecular simulation," J. Comput. Chem. **38**, 1727–1739 (2017).
[61]J. M. R. Byrd, S. A. Jarvis, and A. H. Bhalerao, "Reducing the run-time of MCMC programs by multithreading on SMP architectures," in *2008 IEEE International Symposium on Parallel and Distributed Processing* (IEEE, Miami, FL, USA, 2008) pp. 1–8.