The Markov Chain Monte Carlo (MCMC) method is widely used in various fields as a powerful numerical integration technique for systems with many degrees of freedom. In MCMC methods, probabilistic state transitions can be considered as a random walk in state space, and random walks allow for sampling from complex distributions. However, paradoxically, it is necessary to carefully suppress the randomness of the random walk to improve computational efficiency. By breaking detailed balance, we can create a probability flow in the state space and perform more efficient sampling along this flow. Motivated by this idea, practical and efficient nonreversible MCMC methods have been developed over the past ten years. In particular, the lifting technique, which introduces probability flows in an extended state space, has been applied to various systems and has proven more efficient than conventional reversible updates. We review and discuss several practical approaches to implementing nonreversible MCMC methods, including the shift method in the cumulative distribution and the directed-worm algorithm.

II. BALANCE CONDITION

One of the fundamental algorithms in the MCMC method is the Metropolis algorithm proposed by Metropolis et al. in 1953 (see Ref. [6]). Many MCMC algorithms, including the Metropolis and the heat bath algorithms (Gibbs sampler), satisfy detailed balance, namely reversibility:

$$\pi_i P_{ij} = \pi_j P_{ji} \quad \forall i, j,$$

where $\pi_i$ is the weight of state $i$ (the measure of the target distribution) and $P_{ij}$ is the transition probability from state $i$ to state $j$. These reversible MCMC methods simulate physical dynamics that seem plausible, as real physical systems also satisfy detailed balance in equilibrium. However, if the goal is numerical integration using running averages, Monte Carlo dynamics are not constrained by physical requirements. Detailed balance is not necessary, and global balance, which ensures that the target distribution is stationary under time evolution,
where \( \pi \) is the variance of the physical quantity, and the integrated autocorrelation time is defined as

\[
\tau_{\text{int},\mathcal{O}} = \frac{1}{2} + \sum_{t=1}^{\infty} A_{\mathcal{O}}(t).
\]

From Eqs. (5) and (6), it is clear that the effective number of samples decreases from \( M \) to \( M/2\tau_{\text{int},\mathcal{O}} \) due to autocorrelation. When the autocorrelation function is a single exponential and the correlation times are large, the integrated autocorrelation time is almost identical to the exponential autocorrelation time.

On the other hand, at the critical point, critical slowing down causes the autocorrelation time to increase proportionally to \( L^z \), where \( L \) is the system size, and \( z \) is the dynamic critical exponent. In local reversible update methods, such as single-particle or single-site updates using the Metropolis algorithm, \( z \) typically becomes around 2. Although critical slowing down is a physical phenomenon, the long autocorrelation time becomes a barrier to phase transition analysis. Reducing the dynamic critical exponent is crucial for analyzing critical phenomena.

These autocorrelation times and the resulting dynamic critical exponents depend on the observables to measure. However, the exponential autocorrelation time for most observables is given by \( \tau_{\text{exp},\mathcal{O}} = -1/\ln |\lambda_2| \), where \( \lambda_2 \) is the second largest eigenvalue of the transition matrix, independent of the observable. Furthermore, the second largest eigenvalue is associated with the mixing time, in which any initial distribution nearly converges to the target \( \pi \) within a small tolerance in the total variation distance. The mixing time is typically \( \tau_{\text{mix}} \approx -1/\ln |\lambda_2| \), giving an estimate of an appropriate period of thermalization, or burn-in, steps. In practice, the integrated autocorrelation time is relatively easy to estimate, and we can approximately estimate the exponential autocorrelation time and an appropriate period of thermalization steps.

### IV. SHIFT IN CUMULATIVE DISTRIBUTION

For discrete state space, several approaches to introducing probability flow have been proposed, and their effectiveness has been tested in representative models, such as the Ising and Potts models. In particular, the shift in the cumulative distribution naturally introduces a single parameter, the amount of the shift, and provides a handy approach to controlling the rejection rate and probability flow.\[15\] Here, we review the shift approach and discuss an extension to continuous variables.

Let us consider an update of a state that can take one of \( n \) states. For example, in the single-site update of the six-state Potts model, \( n = 6 \). We first calculate the

\[
\sum_{j} \pi_j P_{ij} = \sum_{j} \pi_j P_{ji} \forall i.
\]

This condition means the total probability flow into and out of each state balances. Detailed balance is a sufficient but not necessary condition for this global balance. MCMC methods that satisfy global but not detailed balance are called nonreversible MCMC methods. Physically, this corresponds to calculating long-time averages using nonequilibrium steady states.

Interestingly, it has been proven that breaking the reversibility in a reversible method always speeds up distribution convergence.\[11\]\[13\] The problem of how to optimally introduce probability flow remains a hot topic in probability theory and applied mathematics. In the following, we discuss probability optimization and lifting, relatively simple methods to construct efficient nonreversible MCMC methods.

### III. COMPUTATIONAL EFFICIENCY OF THE MCMC METHOD

In the MCMC method, a generated state depends on the previous states, leading to correlations (autocorrelations) between the states at different times. Generally, reducing autocorrelation improves the computational efficiency of the MCMC method. The autocorrelation function, defined as

\[
A_{\mathcal{O}}(t) = \frac{\langle \mathcal{O}_i \mathcal{O}_{i+t} \rangle - \langle \mathcal{O}_i \rangle^2}{\langle \mathcal{O}_i^2 \rangle - \langle \mathcal{O}_i \rangle^2},
\]

describes the correlation of physical quantities at time \( t \) in Monte Carlo steps. Here, \( \mathcal{O}_i \) represents the value of a physical quantity, or observable, \( \mathcal{O} \) at the \( i \)-th Monte Carlo step. The bracket \( \langle \cdot \rangle \) denotes the Monte Carlo average, taking the average over \( i \), and thus the autocorrelation function \[3\] is independent of \( i \). In most cases, the autocorrelation function is given by the sum of multiple exponential functions,\[12\] and at large \( t \) it decays as

\[
A_{\mathcal{O}}(t) \approx e^{-t/\tau_{\text{exp},\mathcal{O}}},
\]

where \( \tau_{\text{exp},\mathcal{O}} \) is the exponential autocorrelation time or relaxation time, indicating the degree of correlation with previous values. On the other hand, the mean squared error of physical quantities obtained from the MCMC method asymptotically follows

\[
\sigma_{\mathcal{O}}^2 \approx \frac{v_{\text{asymp},\mathcal{O}}}{M},
\]

where \( M \) is the total number of samples, and the asymptotic variance \( v_{\text{asymp},\mathcal{O}} \) quantifies the sampling efficiency, being often used for performance comparisons of algorithms. Assuming no bias in physical quantity calculations, the asymptotic variance can be expressed as

\[
v_{\text{asymp},\mathcal{O}} = 2\tau_{\text{int},\mathcal{O}} v_{\mathcal{O}},
\]
Fstochastic flows of the shift, $0$ flow from $i$ shifted towers. It is convenient to define the stochastic probability from the overlap between the original and shifted towers for each state. See the main text for the details. (c) When $s = \max i$, this approach reduces to the Suwa–Todo algorithm\textsuperscript{[15]}\textsuperscript{[16]} Taken from Ref. 17.

cumulative weight distribution:

$$F_i = \sum_{j=1}^{i} \pi_j \quad (1 \leq i \leq n) \quad (8)$$

and $F_0 = 0$. Here, the order of the states is arbitrary. Calculating the cumulative distribution is nontrivial for a huge $n$ but easy for a small $n$, such as in the update of local variables. Let us next consider the sequence of $F_i$, or the weight tower, as illustrated in Fig. 1(a) for $n = 6$. We then perform shifting the tower by a certain amount, as in Fig. 1(b). The shift is periodic; the weight shifted over the top of the tower is allocated to the lowest part of the tower.

Using this weight shift, we determine the transition probability from the overlap between the original and shifted towers. It is convenient to define the stochastic flow from $i$ to $j$: $v_{ij} = \pi_i P_{ij}$. Let $s$ denote the amount of the shift, $0 < s < F_n$. We focus on the overlap between the ranges $[F_{j-1}, F_j]$ and $[F_{i-1} + s, F_i + s]$. The stochastic flows $v_{ij}$ are set to the overlap for $i = 1, 2$ and 3 in the case of Fig. 1(b). If $F_i + s > F_n$, we take into account an additional overlap between the ranges $[F_{j-1} + F_n, F_j + F_n]$ and $[F_{i-1} + s, F_i + s]$, assuming that the tower is periodic, for example, for $i = 4, 5$ and 6 in Fig. 1(b). If a shifted weight range crosses the top of the original tower, nonzero flows are allocated from the overlaps in the above two cases, such as for $i = 4$ in Fig. 1(b) and $i = 2$ in Fig. 1(c). The resulting stochastic flow set by the periodic shift $s$ becomes\textsuperscript{[14]}

$$v_{ij} = \max(0, \min(\Delta_{ij}, \pi_i + \pi_j - \Delta_{ij}, \pi_i, \pi_j))$$

$$+ \max(0, \min(\Delta_{ij} - F_n, \pi_i + \pi_j + F_n - \Delta_{ij}, \pi_i, \pi_j)), \quad (9)$$

where

$$\Delta_{ij} = F_i - F_{j-1} + s. \quad (10)$$

The first term of Eq. (9) is the overlap between the ranges $[F_{j-1}, F_j]$ and $[F_{i-1} + s, F_i + s]$, and the second term is the overlap between the ranges $[F_{j-1} + F_n, F_j + F_n]$ and $[F_{i-1} + s, F_i + s]$. Refer to Ref. 17 for the derivation of the analytical expression. The transition probability is then set to $P_{ij} = v_{ij} / \pi_i$.

Clearly, each range of $\pi_i$ in the original tower is covered by the shifted tower due to the periodic shift. This shift algorithm always satisfies the following condition:

$$\pi_i = \sum_j v_{ji} \quad \forall i, \quad (11)$$

equivalent to the global balance\textsuperscript{[2]}. Using the shift algorithm for each local variable satisfies the global balance of the total system\textsuperscript{[17]}. The rejection-free condition is straightforward in the shift approach: $v_{ii} = \pi_i P_{ii} = 0 \forall i$, can be obtained for $\pi_{\text{max}} \leq s \leq F_n - \pi_{\text{max}}$ if $\pi_{\text{max}} \leq F_n/2$, and there is no rejection-free solution otherwise. When $s = \pi_{\text{max}}$, this approach reduces to the Suwa–Todo algorithm\textsuperscript{[15]} Setting $s = F_n/2$ is another optimal choice, minimizing the rejection probability. It provides one of the best local update methods in the Potts model as well as the Suwa–Todo algorithm. Thanks to the introduction of the shift parameter, we can readily control the rejection probability. The autocorrelation time of the order parameter in the Potts model exponentially decreases with the reduction of the rejection rate\textsuperscript{[17]}

Next, let us consider an update of a continuous state variable for which the inversion method can be used. It can be updated by the heat bath algorithm, namely the Gibbs sampler; a uniformly random variable $r \in [0, 1]$ is generated, and the next state is chosen from the inverse function of a conditional cumulative distribution. We can extend the shift approach to this case. In order to explain the shift in a continuous case, let us consider the bivariate Gaussian distribution as a simple example:

$$\pi(x_1, x_2) \propto e^{-\frac{(x_1 - x_2)^2}{2\sigma_1^2} - \frac{(x_1 + x_2)^2}{2\sigma_2^2}}. \quad (12)$$

Given $x_2$, the local variable $x_1$ is updated by using the conditional (cumulative) distribution

$$F(x_1|x_2) = \int_{-\infty}^{x_1} \pi(x, x_2)dx. \quad (13)$$

The Gibbs sampler determines the next state as

$$x_1' = F^{-1}(r), \quad (14)$$

where $r \in [0, 1]$ is a uniformly (pseudo) random variable. This process satisfies the detailed balance.

The overrelaxation method\textsuperscript{[15]} is known to be one of the best ways to update Gaussian variables. The name overrelaxation comes from the idea of making the Markov
chain have a negative correlation. In this method, for the generation of a variable from a conditional Gaussian distribution \( \pi(z_i | \cdot) \sim \mathcal{N}(\mu_i, \sigma_i^2) \), the next state is chosen as \( z_i' = \mu_i + \alpha (z_i - \mu_i) + \sigma_i \sqrt{1 - \alpha^2} \nu_i \), where \( \nu_i \) is a random variable generated from \( \mathcal{N}(0,1) \) and \( \alpha \) is a parameter \((-1 < \alpha < 1)\). For more than two Gaussian variables, each variable can be updated sequentially using the conditional distribution, similar to the bivariate Gaussian variables. Extending the overrelaxation method to a general distribution is an interesting problem. One of the extended approaches, called ordered overrelaxation, was proposed \(^{19}\) after some candidates are generated and ordered, the next state is chosen on the approximately opposite side from the current position.

Here, as another update method, \(^{21}\) let us choose the next state as

\[
x'_1 = F^{-1}( \{ F(x_1 | x_2) + c + wu \} ), \tag{15}
\]

where \( x_1 \) is the current state, \( c \) and \( w \) are positive real parameters with \( c \geq w \), and \( u \) is an uniformly random variable in \([-1,1] \), respectively. The symbol \( \{a \} \) takes the fractional portion of a real number \( a \). If we use \( w = 1/2 \), this process is nothing but the Gibbs sampler. On the other hand, when \( w < 1/2 \) and \( c \neq 0,1/2 \), it does not satisfy the detailed balance, and there is a net stochastic flow. This flow can push the configuration globally, as in Fig. 2. As a result, the autocorrelation time of \( (x_1 + x_2)^2 \) is significantly reduced, as shown in Fig. 3. In this figure, the Gibbs sampler, the overrelaxation methods with \( \alpha = -0.86 \), the ordered overrelaxation method (with 10 candidates), and the nonreversible update method with \( c = 0.4 \) and \( w = 0.05 \) are tested. The nonreversible kernel produces the smallest correlation for \( \sigma_2/\sigma_1 \geq 50 \) and achieves about 50 times as short the correlation time as the Gibbs sampler. We can surely find better parameter sets of the shift algorithm than the best parameter of the conventional overrelaxation methods in almost the whole region.

V. MULTIPLE PROPOSALS

In this section, we explain that it is possible to reduce the rejection rate for general cases significantly. When the direct inversion method, discussed in the previous section, cannot be applied, we typically resort to the Metropolis algorithm, where a candidate is generated and accepted/rejected according to the weight and proposal probability ratio. It has been a canonical MCMC method algorithm since its invention in 1953 (see Ref. 6). However, the inevitable rejection often obstructs efficient sampling. When the number of candidates is two (includ-
ing the current state), the Metropolis algorithm achieves a minimized rejection rate. To reduce the rejection rate, we must prepare more candidates. Several methods have been proposed as an alternative to the simple Metropolis algorithm. An example is the multipoint Metropolis method; after generating several candidate states, the next configuration is chosen stochastically with the detailed balance kept.

After creating a set of candidate states, we can apply the rejection-minimized algorithm [1]. Let us consider sampling from the wine-bottle (Mexican-hat) potential:

$$\pi(x_1, x_2) \propto e^{-\frac{(x_1-x_2)^2}{2\sigma_1^2} + \frac{(x_1+x_2)^2}{2\sigma_2^2} - h} + \frac{k^2}{4},$$

where $\sigma_1, \sigma_2$, and $h$ are positive parameters of the potential function. Then, we propose a candidate configuration using the isotropic bivariate Gaussian distribution $q(\Delta x_1, \Delta x_2) \propto \exp(-((\Delta x_1)^2 - (\Delta x_2)^2)).$ As $q$ is symmetric with respect to the current and proposed states, the proposal probability does not need to be taken into account in the global balance condition.

Here, we try to make multiple proposals. If we propose candidates from the current position and naively make the proposal probability does not need to be taken into account in the global balance condition.

We avoid this problem by introducing a hub (pivot) from which candidates are proposed according to $q(\Delta x_1, \Delta x_2)$, as shown in Fig. 4. Since $q$ is symmetric with respect to the hub and each proposed candidate, the joint proposal probability of multiple candidates becomes the same regardless of the current state. Specifically, we use the following multiple-proposal strategy for $n$ candidates:

1. A configuration is chosen as a hub (pivot) from the current configuration by a proposal distribution.
2. $(n-1)$ candidates are generated from the hub using the same proposal distribution as process 1.
3. The next state is chosen among the $n$ candidates (including the current state) using the transition probabilities, only taking into account the weights of the states.

This procedure example for $n = 4$ is shown in Fig. 4. In process 3, we can apply the shift method and minimize the rejection probability, as explained in Sec. IV.

Figure 5 shows that the rejection rate is indeed reduced by using this multiple-proposal algorithm and the nonreversible kernel with the choice of $s = \text{max}_i \pi_i$ as in the Suwa–Todo algorithm. The correlation time of $(x_1 + x_2)^2$ also gets shorter as the number of candidates is increased. The extension to higher-dimensional cases is straightforward.

We note that parallel computation can significantly help to reduce the computational cost of the multiple-proposal method. Using a single core, the computational cost in each Monte Carlo update increases proportionally to $n$ for large $n$. However, candidate states can be prepared in parallel using multicores, and the computation time can be kept almost independent of $n$. Then, the total computation time needed, the product of the correlation time and the wall clock time for each Monte Carlo step, decreases with increasing $n$. Such parallel computation helps simulate hardly relaxing problems, such as protein folding.

VI. LIFTING

Proposing good candidate states is crucial for efficient Monte Carlo sampling. To reduce correlation times, efficient transitions between states with high weights are necessary. However, since the topology of these high-weight states in state space is usually unknown, except for trivial cases, proposing good candidate states is difficult.

An effective idea is to expand the state space and connect high-weight states with relatively easy paths. A typical method using this idea is the hybrid (Hamiltonian) Monte Carlo method, which assigns virtual momenta to state variables and updates states according to
A rejection-free state update in interacting particle systems is achieved in one-dimensional Ising and Potts models.\(^3,4\) Newtonian dynamics, enabling efficient transitions between states with constant total energy.\(^29\) The worm algorithm, introduced later, applies to systems with conservation laws, expanding the state space to one where these laws are broken.

Originally, these algorithms were designed to satisfy the detailed balance. The lifting technique intentionally expands the state space and introduces probability flow in the expanded space. Skillfully designed nonreversible MCMC methods can significantly improve sampling efficiency.

The clear idea of lifting was proposed by Diaconis et al., who discussed sampling from \(n\) states arranged in one dimension, where only one neighboring state can be moved to each time step.\(^{26}\) Without bias, the dynamics are diffusive, and the mixing (relaxation) time is \(\tau_{\text{mix}} = O(n^2)\). Expanding the state space by preparing two states, + and −, for each state, and creating probability flow such that + states move in the positive direction and − states move in the negative direction, as shown in Fig. 6, the global balance is maintained using skewed detailed balance.\(^27\)

\[
\pi_i P_{(i, +)\to(i+1, +)} = \pi_{i+1} P_{(i+1, -)\to(i, -)} \quad (17)
\]

When such a probability flow is introduced, and the transition probabilities are optimally chosen, the time to traverse the entire system can be reduced to \(\tau_{\text{mix}} = O(n^2)\), a ballistic dynamics compared to diffusive dynamics without flow.\(^{26}\) Probability flow can generally shorten relaxation times by up to a square root. For two-dimensional uniform distributions, similar relaxation time reduction is possible.\(^{26,29}\) In the mean-field (fully connected) Ising model, state variables are essentially represented by total energy. Introducing probability flow reduces the relaxation time at the critical temperature from \(\tau_{\exp} \propto N^{3/2}\) to \(N^{3/4}\) (see Refs. \(^{27}\) and \(^{31}\)). A similar reduction is achieved in one-dimensional Ising and Potts models.\(^{31,32}\)

One of the successful applications of lifting is the event-chain Monte Carlo (ECMC) method, which performs a rejection-free state update in interacting particle systems in continuous space.\(^{31,33}\) The ECMC method was first applied to the two-dimensional hard-sphere model, demonstrating that equilibrium could be achieved even in systems with one million particles.\(^{34,35,36}\) This simulation clarified phase transition phenomena in two-dimensional systems, which had been debated for decades, including that the transition from the liquid phase to the hexatic phase is a Mayer-Wood type first-order transition, and the transition from the hexatic phase to the solid phase is a Berezinskii-Kosterlitz-Thouless transition. The method has also been extended to particle systems with many-body interactions involving three or more bodies and long-range interactions.\(^{37}\) In many cases, it provides more efficient sampling than molecular dynamics in single-threaded computations. Furthermore, the ECMC method can be applied not only to particle systems but also to classical spin models. For instance, in the three-dimensional classical Heisenberg spin system, the dynamic critical exponent \(z\) decreases to one at the critical temperature.\(^3\) Additionally, in the low-temperature phase of the three-dimensional XY spin glass, relaxation is accelerated compared to other Monte Carlo methods.\(^{38}\) demonstrating its effectiveness in various systems.\(^{39}\)

VII. DIRECTED WORM ALGORITHM

We introduce another application of lifting to statistical mechanics problems, the directed worm algorithm. Initially proposed for quantum systems with constraints, such as particle number conservation, the worm algorithm extends the state space to include states violating these constraints.\(^{39,40}\) In the worldline representation, it efficiently samples paths that do not break particle conservation. However, local state updates while maintaining constraints are inefficient, and it is challenging to update topological quantities like the winding number of worldlines. Generally, sampling constrained state spaces is a nontrivial problem.

The worm algorithm inserts a pair of kinks that break constraints and performs state updates by using the random walk of these kinks. By traversing states that violate the constraints, it enables efficient updates and nonlocal state transitions, resolving the problem of sampling topological quantities. The kinks’ random walk resembles a worm, giving the algorithm its name.

The computational efficiency of the worm algorithm depends heavily on the behavior of the kinks’ random walk. As discussed in Sec. VII, minimizing randomness and achieving ballistic movement improves efficiency. The idea of lifting is applied to the kinks’ random walk by extending the state space to include directional degrees of freedom. Creating probability flow in this expanded space enhances sampling efficiency. The improved directed worm algorithm has become a standard method for quantum spins and boson systems.\(^41\)

The worm algorithm also applies to many classical sys-
tems. For instance, the Ising model can be reformulated as a constrained problem by converting spin variables into bond variables \( \sigma_i \). This constraint is naturally suited to the worm algorithm, making it the most efficient computational method for the Ising model. This reformulation is applicable to models like the Potts and \( \phi^4 \) models. The worm algorithm is widely used in spin glass models, \( O(n) \) loop models, and lattice QCD significantly reducing correlation times and becoming one of the most efficient state update methods. Here, we explain the recently proposed directed worm algorithm for classical systems.

Consider the Ising model on a bipartite lattice with nearest-neighbor interactions. Let the number of sites be \( N \), and the number of bonds be \( N_b \). Transform the site spin variables \( \sigma_i = \pm 1 \) to bond variables \( n_b = 0, 1 \). Defining the spin coupling constant as \( J \) and \( K = \beta J \), the partition function can be rewritten as

\[
Z = \sum_{\{\sigma_i\}} e^K \sum_{(i,j)} \sigma_i \sigma_j = (2 \cosh K)^{N_b} \sum_{\{n_b\}} \prod_{(i,j)} (\tanh K)^{n_b}, \tag{18}
\]

where the active bonds with \( n_b = 1 \) form loops in the state space (Fig. 7(a), (i)). Nonloop states do not contribute to the partition function. This representation of the partition function is physically associated with the high-temperature expansion.

Prokof'ev et al. proposed the worm algorithm for systems with loop constraints, inserting kinks, and updating bond variables while the kinks’ random-walk. The transition probabilities are determined by the state weights in the new ensemble. The directed worm algorithm, introduced by one of the present authors, extends the state transition probabilities to satisfy the loop constraint. (b) A bond is randomly selected, and a pair of kinks (red dots) is inserted. (c)-(g) The kinks perform a random walk while having a direction (blue arrows), updating the bond variables. (h) The process stops when the two kinks meet again. (i) The kinks are removed, and the system transitions to another state satisfying the loop constraint.

Even after introducing a degree of freedom in orientation, computational efficiency is reduced if there is a process where the kink immediately returns to the path it has followed. For optimizing such transition probabilities, the geometric allocation method developed in Ref. 15 is very useful. This method allows for the intuitive design of probability flows and allows optimizations such as rejection minimization to be performed.

Consider the case where there are \( n \) candidate states to transition to. If we define a stochastic flow \( v_{ij} = \pi_i P_{ij} \), the two conditions the stochastic flow must satisfy are

\[
\pi_i = \sum_{j=1}^{n} v_{ij} \quad \forall i \tag{19}
\]

\[
\pi_j = \sum_{i=1}^{n} v_{ij} \quad \forall j \tag{20}
\]

corresponding to probability conservation and the global balance conditions, respectively.

In the geometric allocation method, the transition probabilities can be determined intuitively, as shown in Fig. 8. In this figure, the transition probabilities of the directed worm algorithm in the two-dimensional system \( (n = 4) \) are optimized. Determining the probability flow to satisfy Eqs. (19) and (20) is equivalent to rearranging the colors while preserving the area of each color in
FIG. 8. Optimized transition probabilities for the directed worm algorithm in two-dimensional systems using geometric allocation methods. The probability of the kink reflecting back to the original bond (rejection rate) is minimized. Furthermore, under this condition, the probability of the kink moving straight is maximized. On a square lattice, this results in either case (A) or (B) using rotation and reflection. Here, the weight ratio is $\frac{\pi/4}{\pi/1} = \tanh K$ according to Eq. (18). These transition probabilities satisfy the skewed detailed balance ($v_{ij} = v_{ji}$). (A) For example, the transition probabilities in the case of Fig. 7(d). With $v_{12} = \pi/4$, $v_{13} = v_{14} = \frac{1}{2}(\pi_1 - \pi_4)$, and $v_{34} = \frac{1}{2}(3\pi_1 - \pi_4)$. (B) For example, the transition probabilities in the case of Fig. 7(e) (rotated 90 degrees clockwise). With $v_{12} = \frac{1}{2}(\pi_1 + \pi_4)$, $v_{13} = v_{23} = \frac{1}{2}(\pi_1 - \pi_4)$, and $v_{34} = \pi_4$. Partially modified from Ref. 46.

the diagram and the overall box shape; this is the heart of this approach. The probability of rejection $\sum_i v_{ii}$ is the sum of the areas of the colors allocated to the original boxes. In the allocation depicted in Fig. 8, it can be seen that the transition probability has no rejection ($v_{ii} = 0$). In the $d$-dimensional Ising model, the rejection rate can be set to zero in the temperature region $T \leq \frac{2}{d\pi} = T_{\text{Bethe}}$, which is the transition temperature in the Bethe approximation and always higher than the true transition temperature for $d < \infty$ (see Ref. 17). Furthermore, here, under a minimal rejection rate, the probability of a straight transition was allocated to be maximal. In addition, this transition probability satisfies the skewed detailed balance ($v_{ij} = v_{ji}$) [Eq. (17)]. However, it is easy to break this condition and further increase the irreversibility. This intuitive and flexible optimization is a significant property of the geometric allocation method. As described in the caption of Fig. 8, the transition probabilities can be written down analytically and efficiently implemented in programming codes. Similar improvements and optimizations can also be made for higher-dimensional Ising models.

In such optimized directed worms, the randomness is reduced, and the kink diffusion coefficient at the critical temperature of the two-dimensional Ising model is six times as large as that of the worm algorithm with a random walk on the site. Figure 9 shows the integrated autocorrelation time and the asymptotic variance of the total energy as a function of the system length at the critical temperature of the three-dimensional Ising...
model. The directed worm algorithm produces a significantly shorter autocorrelation time than the Prokof’ev–Svistunov worm and the Wolff cluster algorithms. Although the dynamic critical exponent is presumably the same in these algorithms, the sampling efficiency of the directed worm algorithm, measured by the asymptotic variance, is approximately 27 and 2.2 times as high as that of the Prokof’ev–Svistunov worm and the Wolff cluster algorithms, respectively. Similar efficiency enhancement is confirmed for the susceptibility. Note that the dynamic critical exponent of the Wolff algorithm is estimated to be $z = 0.24(2)$ in the three-dimensional Ising model. Also at the critical temperature of the four-dimensional Ising model, the asymptotic variances of the energy and susceptibility are reduced by a factor of 80. The directed worm algorithm is more efficient than the widely used Wolff cluster algorithm, one of the most efficient Monte Carlo update methods. The dynamic critical exponents of the three- and four-dimensional Ising models in the worm algorithm are $z \approx 0.27$ and 0, respectively, significantly smaller than in the usual local state update ($z \approx 2$) (See Ref. [46] and [47]). Thus, the worm algorithm significantly reduces the dynamic critical exponents of various models, making it a powerful numerical method for studying critical phenomena.

VIII. FUTURE PROSPECTS

Finally, we discuss future prospects for MCMC methods that design probability flow. Currently, molecular dynamics methods are often used for particle systems in continuous space, but slow relaxation due to interactions remains a major issue. The ECMC method is expected to contribute significantly to sampling from the Boltzmann distribution of systems. However, it is challenging to restrict the probability processes to local regions, making parallel computation difficult. Recently, shared-memory parallel computation methods for ECMC have been developed, accelerating calculations by about ten times compared to single-threaded computations. Developing distributed-memory parallel computation methods could lead to a significant breakthrough.

The directed worm algorithm is applicable not only to the Ising model but also to many quantum and classical models. As another application of lifting, a worm algorithm introducing probability flow along the energy axis has been proposed. The fusion of these methods and the introduction of different types of probability flow are expected to lead to further improvements.

In addition to the algorithms introduced in this paper, MCMC methods introducing probability flow are used in various contexts. For example, simulated tempering and replica exchange methods can significantly reduce relaxation times by using lifting. Optimizing replica exchange probabilities by breaking detailed balance with geometric allocation methods can further enhance efficiency. These attempts to introduce probability flow into extended ensemble methods have broad applications and are important research topics. Additionally, nonreversible Metropolis-Hastings methods introducing probability vortices have been proposed. The development of various nonreversible MCMC methods is expected to continue.

Not only MCMC methods but also stochastic processes like Langevin dynamics are gaining attention for dynamics that do not satisfy detailed balance. In non-reversible Langevin dynamics, net probability flow accelerates distribution convergence and reduces asymptotic variance.

Nonequilibrium steady states are being actively researched not only for sampling but also for extending equilibrium statistical mechanics and studying open systems. Mathematically, convergence and ergodicity of nonreversible Markov chains are major research themes. These new developments are expected to form the foundational theory for nonreversible MCMC methods and guide the development of efficient algorithms.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP20H01824, JP22K03508, JP24H01609, and JP24K00543 and the Center of Innovation for Sustainable Quantum AI (SQAI), JST Grant Number JPMJPF2221, CREST Grant Number JPMJCR24I1.

REFERENCES

1. M. Newman and G. Barkema, Monte Carlo methods in statistical physics (Oxford University Press, 1999).
2. D. P. Landau and K. Binder, A Guide to Monte Carlo Simulations in Statistical Physics, 4th ed. (Cambridge University Press, Cambridge, 2014).
3. Y. Sakai and K. Hukushima, J. Phys. Soc. Jpn. 82, 064003 (2013).
4. F. Paizi, G. Deligiannidis, and E. Rosta, Journal of Chemical Theory and Computation 16, 2124 (2020).
5. Y. Nishikawa, M. Michel, W. Krauth, and K. Hukushima, Phys. Rev. E 92, 063306 (2015).
6. W. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. 21, 1087 (1953).
7. M. Creutz, Phys. Rev. D 21, 2308 (1980).
8. S. Geman and D. Geman, IEEE Trans. Pattern Anal. Mach. Intell. 7, 721 (1984).
9. C. P. Robert and G. Casella, Monte Carlo Statistical Methods 2nd ed. (Springer, New York, 2004).
10. In the present paper, we assume irreducibility of the Markov chain, which ensures, under global balance, convergence of the running average to the expectation value with respect to the target.
11. C.-R. Hwang, S.-Y. Hwang-Ma, and S.-J. Sheu, Ann. Appl. Probab. 15, 1433 (2005).
12. A. Ichiki and M. Ohzeki, Phys. Rev. E 88, 020101 (2013).
13. A. B. Duncan, T. Lelievre, and G. A. Pavliotis, J. Stat. Phys. 163, 457 (2016).
