

**Lifted worm algorithm for the Ising model**Eren Metin Elçi,<sup>1</sup> Jens Grimm,<sup>2</sup> Lijie Ding,<sup>3</sup> Abraham Nasrawi,<sup>2</sup> Timothy M. Garoni,<sup>2</sup> and Youjin Deng<sup>3,4</sup><sup>1</sup>*School of Mathematical Sciences, Monash University, Clayton, Victoria 3800, Australia*<sup>2</sup>*ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS), School of Mathematical Sciences, Monash University, Clayton, VIC 3800, Australia*<sup>3</sup>*Department of Modern Physics, University of Science and Technology of China, Hefei, Anhui 230026, China*<sup>4</sup>*National Laboratory for Physical Sciences at Microscale, University of Science and Technology of China, Hefei, Anhui 230026, China*

(Received 14 November 2017; published 18 April 2018)

We design an irreversible worm algorithm for the zero-field ferromagnetic Ising model by using the lifting technique. We study the dynamic critical behavior of an energylike observable on both the complete graph and toroidal grids, and compare our findings with reversible algorithms such as the Prokof'ev-Svistunov worm algorithm. Our results show that the lifted worm algorithm improves the dynamic exponent of the energylike observable on the complete graph and leads to a significant constant improvement on toroidal grids.

DOI: [10.1103/PhysRevE.97.042126](https://doi.org/10.1103/PhysRevE.97.042126)**I. INTRODUCTION**

Markov-chain Monte Carlo (MCMC) algorithms are a powerful and widely used tool in various areas of physics and other disciplines, such as in machine learning [1] and statistics [2]. In many practical applications, MCMC algorithms are constructed via the Metropolis [3] or heat bath update scheme [4]. Such algorithms are necessarily *reversible*.

One important example of a Metropolis algorithm is the Prokof'ev-Svistunov worm algorithm (P-S worm algorithm), which has widespread application for both classical and quantum systems [5,6]. As opposed to cluster algorithms such as the Wolff [7] or Swendsen-Wang [8] algorithms, the updates of the worm algorithm are purely *local*. On the simple-cubic lattice with periodic boundaries, it was numerically observed that the P-S worm algorithm for the zero-field ferromagnetic Ising model outperforms the Swendsen-Wang algorithm for simulating both the magnetic susceptibility and the second-moment correlation length [9]. Another numerical work suggested that the spin-spin correlation function can also be simulated efficiently [10]. Recently, it was rigorously established [11] that the P-S worm algorithm for the Ising model is rapidly mixing on any finite graph for the whole temperature range.

In recent years, various *irreversible* MCMC algorithms have also been studied [12–22]. Many of these algorithms are based on the lifting technique introduced in [23]. The general idea of lifting is to enlarge the original state space and define transition probabilities such that the lifted chain projects down to the original one. The intuition underlying a potential efficiency improvement is the reduction of diffusive behavior, compared with the original Markov chain. Rather than exploring states via random walk in the reversible chain, one introduces directed flows in the lifted chain to move between relevant states significantly faster.

Even though lifting is considered as a promising method to speed up MCMC algorithms, it is an open question how

it affects efficiency in specific examples [24]. For the Ising model on the complete graph, it was numerically observed that the lifted single-spin flip Metropolis algorithm improves the scaling (with volume) of the rate of decay of the autocorrelation function of the magnetization [14]. Another study [13] proved that a lifted MCMC algorithm for uniformly sampling leaves from a given tree reduces the mixing time. In other examples [16,20,22], it was numerically observed that lifting speeds up reversible MCMC algorithms by a possibly large constant factor but does not asymptotically affect the scaling with the system size.

In this work, we investigate how lifting affects worm algorithms. More precisely, we design a lifted worm algorithm for the zero-field ferromagnetic Ising model and numerically study the dynamic critical behavior of the number of occupied edges  $\mathcal{N}$ , which is related to the energy of the Ising system (see Sec. IV). Our simulations were performed on both the complete graph and toroidal grids in dimensions  $2 \leq d \leq 5$  at the (estimated, when  $d \geq 3$ ) infinite-volume critical point.

On the complete graph, we find that the lifted worm algorithm significantly improves the dynamic critical exponent  $z_{\text{int}}$  of  $\mathcal{N}$ . In particular, we show that  $\mathcal{N}$  exhibits critical speeding-up [25,26] in the lifted process ( $z_{\text{int}} \approx -0.5$ ), while we observe  $z_{\text{int}} \approx 0$  for the corresponding reversible counterpart. On toroidal grids, we find that the lifted worm algorithm does not affect the scaling with the system size. We emphasize, however, that the lifted process still reduces the variance of  $\mathcal{N}$  by a significant constant. This constant improvement becomes more pronounced for larger dimensions with up to a factor of approximately 141 for  $d = 5$ . We, finally, note that the improvements in high dimensions can be of practical relevance in the current debate about the correct finite-size scaling behavior above the upper critical dimension; see, e.g., [27–30].

This paper is organized as follows. Section II recalls the basic ideas of the P-S worm algorithm. In Sec. III, we construct an irreversible worm algorithm via the lifting technique. We

present the details of our numerical setup in Sec. IV. In Sec. V, we state our results for the dynamic properties of the lifted worm algorithm and compare our findings with reversible worm algorithms. Finally, in Sec. VI, we summarize and discuss our findings.

## II. P-S WORM ALGORITHM

As is well known [31], the zero-field ferromagnetic Ising model can be mapped to an ensemble of high-temperature graphs. Let  $G = (V, E)$  be a finite graph with vertex set  $V$  and edge set  $E$ . Define the closed loop space  $\mathcal{C}_0$  as the set of all configurations  $\omega \subseteq E$  such that every vertex has even degree, and  $\mathcal{C}_2$  as the set of all  $\omega \subseteq E$  where exactly two vertices have odd degree. We call the subgraph  $(V, \omega)$  Eulerian whenever  $\omega \in \mathcal{C}_0$ . In the high-temperature expansion, the partition function of the Ising model can be written as the sum over all Eulerian subgraphs [31], i.e.,

$$Z = 2^{|V|} \cosh^{|E|}(\beta) \sum_{\omega \in \mathcal{C}_0} \tanh^{|\omega|}(\beta), \quad (1)$$

where  $\beta$  denotes the inverse Ising temperature and  $|\cdot|$  denotes the cardinality of the corresponding set.

The P-S worm algorithm samples these high-temperature graphs via elementary local moves. The main idea is to enlarge the state space  $\mathcal{C}_0$  to  $\mathcal{W} := \mathcal{C}_0 \cup \mathcal{C}_2$  by introducing two vertices with odd degree (defects). These defects are moved through  $\mathcal{W}$  via random walk. Whenever the two defects meet, the subgraph becomes Eulerian and one reaches a state of the original configuration space  $\mathcal{C}_0$ .

In the original algorithm [5], only one of the defects is mobile and can be moved through  $\mathcal{W}$ . In this work, we use a slightly different worm version where we flip a fair coin to select the mobile defect in  $\mathcal{C}_2$ . The *transition probabilities*

$$q(\omega, \omega') = p(\omega, \omega') a(\omega, \omega') \quad (2)$$

from an initial state  $\omega$  to a target state  $\omega'$  can be calculated by metropolizing [3] the proposals  $p(\omega, \omega')$  with respect to the stationary measure,  $\pi(\omega) = \frac{1}{Z} \tanh^{|\omega|}(\beta) \Psi(\omega)$ . Here,  $Z$  is an appropriate normalization on  $\mathcal{W}$ ,  $\Psi(\omega) := |V|$  if  $\omega \in \mathcal{C}_0$ , and  $\Psi(\omega) := 2$  if  $\omega \in \mathcal{C}_2$ , respectively. Furthermore,  $p(\omega, \omega')$  denotes the *proposal probability* and  $a(\omega, \omega')$  the *acceptance probability*. See [11] for explicit expressions for the transition matrix (2).

The algorithm is presented in Algorithm 1 with  $\omega \Delta x x'$  denoting the symmetric difference of  $\omega$  and the edge  $x x'$ ; i.e.,

$x x' \in \omega \Delta x x'$  if and only if  $x x' \notin \omega$ . In other words, if  $x x' \in \omega$ , we propose to delete  $x x'$ , while if  $x x' \notin \omega$ , we propose to add it.

---

### Algorithm 1 P-S Worm Algorithm

---

**if**  $\omega \in \mathcal{C}_0$ , **then**

Choose a uniformly random vertex  $x$

**else**

Choose a uniformly random odd vertex  $x$

**end if**

Choose a uniformly random edge  $x x'$  among the set of edges incident to  $x$ . With probability  $a_{P-S}(\omega, \omega \Delta x x')$ , let  $\omega \rightarrow \omega \Delta x x'$ .

Otherwise,  $\omega \rightarrow \omega$

---

## III. IRREVERSIBLE WORM ALGORITHM

We construct the irreversible worm algorithm in two steps. In Sec. III A, we first define an alternative *reversible* worm algorithm. This worm algorithm will be an appropriate starting point to apply lifting. In Sec. III B, we use lifting to construct the irreversible counterpart.

### A. B-S-type worm algorithm

Since it is not obvious how to apply the lifting technique to the P-S process in a natural way, we first construct an alternative reversible worm algorithm with slightly different proposals. This algorithm can be seen as the Ising analog of the Berretti-Sokal (B-S) algorithm [32] for simulating self-avoiding walks in the grand canonical ensemble. We thus call it the *B-S-type worm algorithm*.

The proposals are as follows: We first decide to either increase (+) or decrease (−) the number of occupied edges by flipping a fair coin. Then, if the current state belongs to  $\mathcal{C}_2$ , we flip a fair coin to select one of the two defects as the mobile vertex. Otherwise, if the current state is an element of  $\mathcal{C}_0$ , we choose a uniformly random vertex as the mobile vertex. If we decide to add (delete) an edge, we select the next position of the mobile vertex uniformly at random among the set of vacant (occupied) edges incident to the current mobile vertex. We now construct the transition probabilities by metropolizing the proposals with respect to the same measure as in Sec. II.

For  $\omega \in \mathcal{W}$ ,  $v \in V$ , and  $\lambda \in \{-1, 1\}$ , define

$$N_\omega(x, \lambda) = \begin{cases} \{uv \notin \omega : u = x \text{ or } v = x\} & \text{if } \lambda = +1, \\ \{uv \in \omega : u = x \text{ or } v = x\} & \text{if } \lambda = -1. \end{cases} \quad (3)$$

Note that for any  $\omega \in \mathcal{W}$ ,  $N_\omega(x, +1) + N_\omega(x, -1)$  equals the degree of  $x$ .

---

Fix  $z := \tanh(\beta)$  and let  $\omega, \omega \Delta x x' \in \mathcal{W}$ . The proposal and acceptance probabilities for the transition  $\omega \rightarrow \omega \Delta x x'$  are as follows: (i) If  $\omega \in \mathcal{C}_0$ ,

$$p_{B-S}(\omega, \omega \Delta x x') = \frac{1}{2} \frac{1}{|V|} \left[ \frac{1}{|N_\omega(x, |\omega \Delta x x'| - |\omega|)|} + \frac{1}{|N_\omega(x', |\omega \Delta x x'| - |\omega|)|} \right], \quad (4)$$

$$a_{B-S}(\omega, \omega \Delta x x') = \min \left[ 1, \frac{z^{|\omega \Delta x x'|}}{z^{|\omega|}} \frac{|N_{\omega \Delta x x'}(x, |\omega| - |\omega \Delta x x'|)|^{-1} + |N_{\omega \Delta x x'}(x', |\omega| - |\omega \Delta x x'|)|^{-1}}{|N_\omega(x, |\omega \Delta x x'| - |\omega|)|^{-1} + |N_\omega(x', |\omega \Delta x x'| - |\omega|)|^{-1}} \right]. \quad (5)$$

(ii) If  $\omega \in \mathcal{C}_2$ ,  $\omega \Delta x x' \in \mathcal{C}_2$ , and  $x$  is a defect in  $\omega$ ,

$$p_{B-S}(\omega, \omega \Delta x x') = \frac{1}{2} \frac{1}{2} \frac{1}{|N_\omega(x, |\omega \Delta x x'| - |\omega|)|}, \quad (6)$$

$$a_{B-S}(\omega, \omega \Delta x x') = \min \left[ 1, \frac{z^{|\omega \Delta x x'|}}{z^{|\omega|}} \frac{|N_\omega(x, |\omega \Delta x x'| - |\omega|)|}{|N_{\omega \Delta x x'}(x', |\omega| - |\omega \Delta x x'|)|} \right]. \quad (7)$$

(iii) If  $\omega \in \mathcal{C}_2$ ,  $\omega \Delta x x' \in \mathcal{C}_0$ ,

$$p_{B-S}(\omega, \omega \Delta x x') = \frac{1}{2} \frac{1}{2} \left[ \frac{1}{|N_\omega(x, |\omega \Delta x x'| - |\omega|)|} + \frac{1}{|N_\omega(x', |\omega \Delta x x'| - |\omega|)|} \right], \quad (8)$$

$$a_{B-S}(\omega, \omega \Delta x x') = \min \left[ 1, \frac{z^{|\omega \Delta x x'|}}{z^{|\omega|}} \frac{|N_{\omega \Delta x x'}(x, |\omega| - |\omega \Delta x x'|)|^{-1} + |N_{\omega \Delta x x'}(x', |\omega| - |\omega \Delta x x'|)|^{-1}}{|N_\omega(x, |\omega \Delta x x'| - |\omega|)|^{-1} + |N_\omega(x', |\omega \Delta x x'| - |\omega|)|^{-1}} \right]. \quad (9)$$

We give an example for case (ii) in Fig. 1. All other off-diagonal transition probabilities are zero. The full algorithmic description is presented in Algorithm 2. We remark that the choice to allow both defects to move in Algorithm 2 is not actually necessary, and one can construct a modification of Algorithm 2 in which only one defect is mobile.

---

**Algorithm 2** B-S-type Worm Algorithm

---

Choose  $\lambda = \{+, -\}$  uniformly at random

**if**  $\omega \in \mathcal{C}_0$ , **then**

    Choose a uniformly random vertex  $x$

**else**

    Choose a uniformly random odd vertex  $x$

**end if**

**if**  $N_\omega(x, \lambda) = \emptyset$ , **then**

    Set  $\omega \rightarrow \omega$  and skip all following steps

**else**

    Choose a uniformly random edge  $x x' \in N_\omega(x, \lambda)$ . With Probability  $a_{B-S}(\omega, \omega \Delta x x')$ , let  $\omega \rightarrow \omega \Delta x x'$ . Otherwise,  $\omega \rightarrow \omega$ .

**end if**

---

### B. Irreversible worm algorithm

In the following, we construct the irreversible counterpart of the B-S-type worm algorithm. Consider the enlarged state

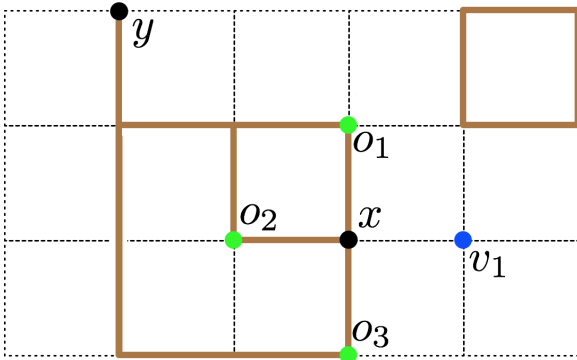


FIG. 1. Worm configuration  $\omega \in \mathcal{C}_2$  where  $x, y$  are the vertices with odd degree,  $N_\omega(x, -) = \{x o_1, x o_2, x o_3\}$ , and  $N_\omega(x, +) = \{x v_1\}$ . If  $x$  is selected as the mobile vertex and one proposes to increase the number of edges in the B-S-type worm algorithm, the only possible transition is  $\omega \rightarrow \omega \Delta x v_1$  where  $\omega \Delta x v_1 \in \mathcal{C}_2$ . The corresponding proposal and acceptance probabilities are stated in Eqs. (6) and (7).

space  $\mathcal{W}' := \mathcal{W} \times \{-, +\}$ , where  $\{-, +\}$  is a set to indicate to either choose to increase (+) or decrease (-) the number of edges. Our aim is to define a Markov chain on  $\mathcal{W}'$  such that we never propose to delete an edge if a state belongs to  $\mathcal{W} \times \{+\}$ , while we never propose to add edges as long as the chain belongs to  $\mathcal{W} \times \{-\}$ . If a move  $(\omega, \lambda) \rightarrow (\omega \Delta x x', \lambda)$  is rejected, we make the transition  $(\omega, \lambda) \rightarrow (\omega, -\lambda)$ . Note that this process does not allow diagonal transitions.

For  $(\omega, \lambda) \in \mathcal{W}'$ , let  $\tilde{\pi}(\omega, \lambda) = \frac{1}{2} \pi(\omega)$ . For  $x x' \in E$ , let

$$\tilde{q}[(\omega, +), (\omega \cup x x', +)] = q_{B-S}(\omega, \omega \cup x x') \text{ if } x x' \notin \omega, \quad (10)$$

$$\tilde{q}[(\omega, -), (\omega \setminus x x', -)] = q_{B-S}(\omega, \omega \setminus x x') \text{ if } x x' \in \omega. \quad (11)$$

All other entries in row  $\tilde{q}[(\omega, \lambda), (\cdot, \cdot)]$  are zero except  $\tilde{q}[(\omega, \lambda), (\omega, -\lambda)]$ , which is fixed by stochasticity. Observe that skew-detailed balance [14, 33] between  $\tilde{q}$  and  $\tilde{\pi}$  follows immediately from detailed balance between  $q$  and  $\pi$ , and so  $\tilde{q}$  has stationary distribution  $\tilde{\pi}$ .

The full algorithmic description of the lifted worm algorithm is given in Algorithm 3. From a practical perspective, we emphasize that only minor code changes to the B-S-type worm algorithm are needed to construct the irreversible counterpart.

---

**Algorithm 3** Irreversible Worm Algorithm

---

**if**  $\tilde{\omega} = (\omega, \lambda)$  where  $\omega \in \mathcal{C}_0$ , **then**

    Choose a uniformly random vertex  $x$

**else**

    Choose a uniformly random odd vertex  $x$

**end if**

**if**  $N_\omega(x, \lambda) = \emptyset$  **then**

    Set  $(\omega, \lambda) \rightarrow (\omega, -\lambda)$  and skip all following steps

**else**

    Choose a uniform random edge  $x x' \in N_\omega(x, \lambda)$ . With probability  $a_{B-S}(\omega, \omega \Delta x x')$ , let  $(\omega, \lambda) \rightarrow (\omega \Delta x x', \lambda)$ . Otherwise,  $(\omega, \lambda) \rightarrow (\omega, -\lambda)$

**end if**

---

## IV. NUMERICAL SETUP

Let  $\omega \in \mathcal{W}$ . We numerically study the dynamic properties of the number of occupied edges  $\mathcal{N} := |\omega|$ . We note that the mean energy of the Ising system is given by  $\langle \mathcal{E} \rangle = -\tanh(\beta) |E| +$

$\langle \mathcal{N} | \mathcal{C}_0 \rangle \sinh^2(\beta)$ ], where  $\langle \mathcal{N} | \mathcal{C}_0 \rangle$  denotes the expectation of  $\mathcal{N}$  conditioned on being in  $\mathcal{C}_0$ . The conditional variance of  $\mathcal{N}$  is similarly related to the specific heat [5]. We emphasize that in what follows,  $\langle \cdot \rangle$  denotes the expectation with respect to the stationary distribution of the worm algorithm on the full space  $\mathcal{W}$  (or  $\mathcal{W}'$  in the lifted case). Such expectations are then estimated via the sample mean; e.g.,  $\langle \mathcal{N} \rangle$  is estimated via  $\bar{\mathcal{N}} = \frac{1}{M} \sum_{i=1}^M \mathcal{N}_i$ , where  $M$  is the total number of measurements and  $\mathcal{N}_i$  is the value of the random variable at the  $i$ th Monte Carlo step after a sufficiently long burn-in sequence has been discarded.

In Sec. V, we compare the variance of  $\bar{\mathcal{N}}$  (in the limit of  $M \rightarrow \infty$ ) by using

$$\text{Var}(\bar{\mathcal{N}}) \sim 2\tau_{\text{int}}^{(\mathcal{N})} \frac{\text{Var}(\mathcal{N}_0)}{M}, \quad M \rightarrow \infty, \quad (12)$$

among the P-S, B-S-type, and irreversible worm algorithms. Here,  $\tau_{\text{int}}^{(\mathcal{N})}$  is the *integrated autocorrelation time*,

$$\tau_{\text{int}}^{(\mathcal{N})} := \frac{1}{2} + \sum_{t=1}^{\infty} \rho^{(\mathcal{N})}(t), \quad (13)$$

where  $\rho^{(\mathcal{N})}(t)$  denotes the *normalized autocorrelation function*,

$$\rho^{(\mathcal{N})}(t) := \frac{\langle \mathcal{N}_0 \mathcal{N}_t \rangle - \langle \mathcal{N}_0 \rangle^2}{\text{Var}(\mathcal{N}_0)}. \quad (14)$$

Our simulations for the Ising model were performed on the complete graph on  $n$  vertices and on toroidal grids for  $2 \leq d \leq 5$ . On the complete graph, we simulated at the critical point  $\beta_{\text{crit}} = 1/n$ . On the torus, our simulations were performed at the exact critical point in two dimensions [34] and at the estimated critical points  $\beta_{\text{crit},3d} = 0.221\,654\,55(3)$  [35],  $\beta_{\text{crit},4d} = 0.149\,694\,7(5)$  [36], and  $\beta_{\text{crit},5d} = 0.113\,915\,0(4)$  [27], for  $d \geq 3$ .

In each time series, we truncated the summation in Eq. (13) self-consistently by using the windowing method [38]. We emphasize that particular care has to be taken when choosing the windowing parameter  $c$  for the irreversible worm algorithm; see Sec. V and Appendix A. For fitting and error estimations, we follow standard procedures; see, e.g., [37,38].

## V. RESULTS

### A. Toroidal grids

We will now study the dynamic properties of  $\mathcal{N}$  on  $d$ -dimensional toroidal grids. Note that  $\text{Var}(\mathcal{N}_0)$  in Eq. (12) coincides among all studied algorithms since it is a property of the stationary measure and does not depend on the details of the underlying Markov chain. Therefore, in the limit  $M \rightarrow \infty$ , we have  $\frac{\text{Var}_i(\bar{\mathcal{N}})}{\text{Var}_j(\bar{\mathcal{N}})} = \frac{\tau_{\text{int},i}^{(\mathcal{N})}}{\tau_{\text{int},j}^{(\mathcal{N})}}$ , where  $i, j \in \{\text{P-S, B-S-type, irre}\}$ .

In Fig. 2, we compare the integrated autocorrelation time among the B-S-type and lifted worm algorithms, and among the B-S-type and P-S worm algorithms. We perform least-squares fits of the form  $A + BL^{-\Delta}$ , where  $A, B, \Delta$  are free parameters. Our conclusion will be that both ratios are approaching constants for  $L \rightarrow \infty$ , with larger improvements for higher dimensions; see Table I. In two dimensions, our fits lead to the constant improvement  $A_{\text{B-S} \rightarrow \text{irre}} = 1.7(2)$ , and

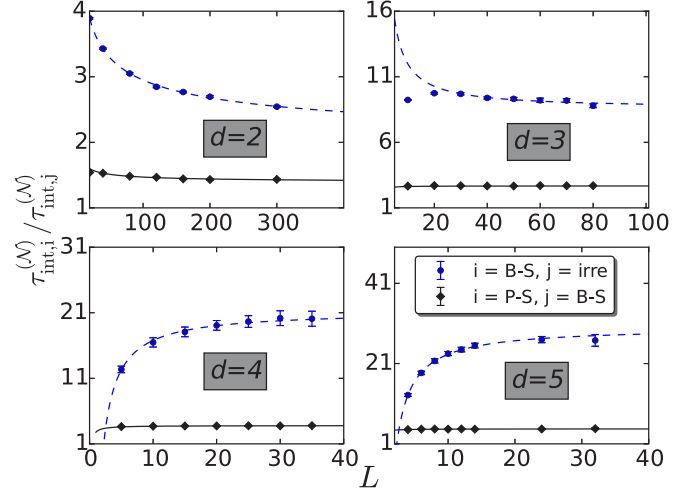


FIG. 2. Comparison of the integrated autocorrelation time among the P-S, B-S-type, and lifted worm algorithms. The black diamonds show the ratio of the integrated autocorrelation time of the P-S and B-S-type algorithms, while the blue circles compare  $\tau_{\text{int}}^{(\mathcal{N})}$  among the lifted and the B-S-type worm algorithms. The asymptotic improvement factors can be found in Table I. The lines correspond to the fits in Sec. V A.

$A_{\text{P-S} \rightarrow \text{B-S}} = 1.4(1)$  by discarding  $L < 40$ . For  $d = 3$ , we find  $A_{\text{B-S} \rightarrow \text{irre}} = 8.2(4)$  by discarding  $L < 40$ , and  $A_{\text{P-S} \rightarrow \text{B-S}} = 2.68(9)$  by discarding  $L < 20$ . In four dimensions, our fits lead to  $A_{\text{B-S} \rightarrow \text{irre}} = 24(1)$ , and  $A_{\text{P-S} \rightarrow \text{B-S}} = 3.79(3)$  by discarding  $L < 10$ . For  $d = 5$ , we find  $A_{\text{B-S} \rightarrow \text{irre}} = 30(1)$  and  $A_{\text{P-S} \rightarrow \text{B-S}} = 4.7(1)$ . In order to obtain stable fits, we fixed  $\Delta = 1$  for fitting the ratios of the integrated autocorrelation time of the B-S-type and lifted worm algorithms for  $d > 2$ , and for fitting the ratios of the P-S and B-S-type worm algorithms in four and five dimensions.

For estimating  $\tau_{\text{int,irre}}^{(\mathcal{N})}$  in four and five dimensions, we had to choose very large  $c$  values outside the common range  $c \in [6, 10]$  in the windowing algorithm [38] (see Appendix A for details). In order to understand this, it proves useful to study the autocorrelation function  $\rho_{\text{irre}}^{(\mathcal{N})}(t)$ , where  $t$  is measured in MC hits. For clarity, we will only focus on the five-dimensional case. Figure 3 shows  $\rho_{\text{irre}}^{(\mathcal{N})}(t)$  in the lifted worm algorithm for  $d = 5$ .  $\rho_{\text{irre}}^{(\mathcal{N})}(t)$  exhibits a two-time scaling: For small  $t$ ,  $\rho_{\text{irre}}^{(\mathcal{N})}(t)$  shows a quick exponential decay to a small but bounded value, while we observe a much slower decay with a different (larger) exponential scale for larger  $t$ . We note that this two-time scaling

TABLE I. Improvement factors  $\tau_{\text{int},i}^{(\mathcal{N})}/\tau_{\text{int},j}^{(\mathcal{N})}$  by changing from the P-S to the B-S-type worm algorithm, B-S-type to the irreversible worm algorithm, and P-S to the irreversible worm algorithm on the  $d$ -dimensional torus.

	P-S $\rightarrow$ B-S	B-S $\rightarrow$ irre	P-S $\rightarrow$ irre
$d = 2$	1.4(1)	1.7(2)	2.4(4)
$d = 3$	2.68(9)	8.2(4)	22(2)
$d = 4$	3.79(3)	24(1)	91(4)
$d = 5$	4.7(1)	30(1)	141(6)

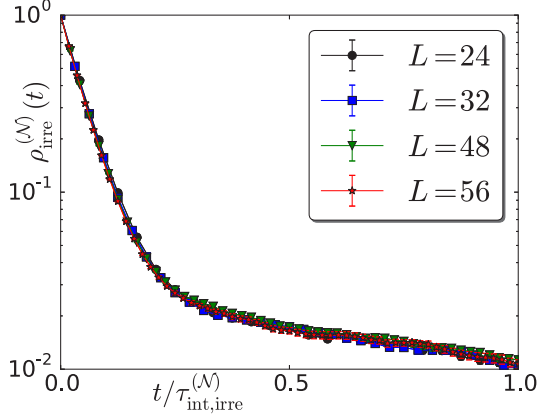


FIG. 3. Normalized autocorrelation function  $\rho_{\text{irre}}^{(N)}(t)$  ( $t$  in MC hits) for the irreversible worm algorithm in five dimensions. As stated in Sec. VA,  $\rho_{\text{irre}}^{(N)}(t)$  is well described by the *ansatz* in Eq. (15).

is absent for  $d \leq 3$ . Our data suggests that  $\rho_{\text{irre}}^{(N)}(t)$  can be described by the *ansatz*

$$\rho_{\text{irre}}^{(N)}(t) = \alpha_1 \exp(-t/\tau_1) + \alpha_2 \exp(-t/\tau_2), \quad (15)$$

where  $\alpha_1 + \alpha_2 = 1$ ,  $\tau_1 \geq \tau_2 > 0$ . In order to estimate the optimal least-squares parameters for this *ansatz*, we used an improved procedure described in Appendix B. We find that both the ratio  $\alpha_1/\alpha_2$  and  $\tau_2/\tau_1$  remain bounded as  $L \rightarrow \infty$ . We estimate the corresponding asymptotic constants using a least-squares fitting procedure with a constant and obtain

$$\frac{\alpha_2}{\alpha_1} \stackrel{L \rightarrow \infty}{\sim} 40.6(4), \quad \frac{\tau_1}{\tau_2} \stackrel{L \rightarrow \infty}{\sim} 32.4(6).$$

Our numerical observation that  $\alpha_1$  ( $\tau_2$ ) is significantly smaller than  $\alpha_2$  ( $\tau_1$ ) impacts the required choice of parameters in the windowing algorithm[38].

### B. Complete graph

We will now study the integrated autocorrelation time of the B-S-type and lifted worm algorithms on the complete graph with  $n$  vertices. Our main finding is that  $\mathcal{N}$  exhibits critical speeding-up for the lifted process. Figure 4 shows  $\tau_{\text{int}}^{(N)}/n$  for both algorithms, where  $\tau_{\text{int}}^{(N)}$  is measured in MC hits. The fitting *ansatz*  $An^z + B$  for  $\tau_{\text{int,irre}}^{(N)}$ , where  $A, B, z$  are free parameters, leads to  $z = 0.51(1)$  by discarding  $n < 2 \times 10^5$ . Thus, we have  $\tau_{\text{int,irre}}^{(N)}/n \sim n^{-1/2}$ . Note that we had to choose large windowing parameters in agreement with our findings on high-dimensional tori. For the corresponding reversible counterpart (B-S-type worm algorithm), it follows immediately from general arguments [[39], Corollary 9.2.3] that the integrated autocorrelation time satisfies a Li-Sokal-type bound,  $\tau_{\text{int,B-S}}^{(N)} \geq \text{const} \times \text{Var}(\mathcal{N}_0)$ , where  $\text{const} > 0$ . One can, furthermore, calculate that  $\lim_{n \rightarrow \infty} \frac{\text{Var}(\mathcal{N}_0)}{n} = \frac{9}{4} - \frac{24\Gamma(5/4)^4}{\pi^2}$  leading to  $\tau_{\text{int,B-S}}^{(N)}/n \geq \text{const}$ . Thus, lifting improves the dynamic critical exponent on the complete graph. Our numerics suggest that this bound is sharp, i.e.,  $\tau_{\text{int,B-S}}^{(N)}/n \sim \text{const}$ . More precisely, our fitting *ansatz*  $An^z + B$  for  $\tau_{\text{int,B-S}}^{(N)}$  leads to  $z = 1.00(2)$  by discarding  $n < 3000$ .

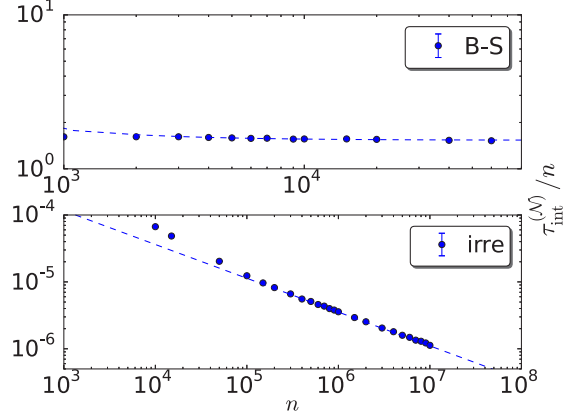


FIG. 4. Finite-size scaling of  $\tau_{\text{int}}^{(N)}/n$  for the B-S-type worm algorithm (upper panel) and lifted worm algorithm (lower panel) on the complete graph with  $n$  vertices. The dashed lines correspond to the fits in Sec. VB.

Figure 5 shows the autocorrelation function  $\rho_{\text{irre}}^{(N)}(t)$  for the lifted process, where  $t$  is measured in MC hits. Our data suggest that  $\rho_{\text{irre}}^{(N)}(t)$  is well described by the following equation:

$$\rho_{\text{irre}}^{(N)}(t) = (1 - \alpha) \exp(-t/\tau_1) + \alpha \cos(\omega t + \phi) \exp(-t/\tau_2). \quad (16)$$

Similar *Ansätze* were used in other studies on lifting; see [14,33]. Our fits lead to  $\tau_1 \propto n^{2/3}$ ,  $\tau_2 \propto n^{1/2}$ , and  $\omega \propto n^{-1/2}$ . Moreover, we find that the amplitude of the first term vanishes as  $n \rightarrow \infty$ . We note that the cosine in Eq. (16) is motivated by the fact that the eigenvalues of the transition matrix in the lifted process need not be real, unlike the spectrum of reversible chains.

Finally, we numerically observe that the average number of consecutive steps the chain spends in each replica,  $\tau_{\text{intra}}$ , scales as  $n^{1/2}$ . More precisely, the *ansatz*  $An^z + B$  leads to  $z = 0.49(1)$  by discarding  $n < 25$ . It is interesting to observe that

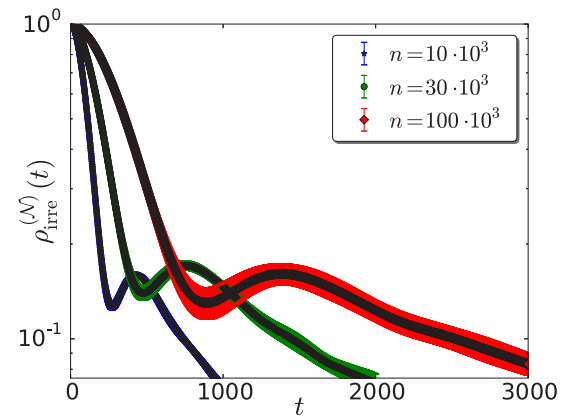


FIG. 5. Normalized autocorrelation function  $\rho_{\text{irre}}^{(N)}(t)$  ( $t$  in MC hits) for the lifted worm algorithm on the complete graph with  $n$  vertices. The blue (left) curve corresponds to  $n = 10 \times 10^3$ , the green (middle) curve corresponds  $n = 30 \times 10^3$ , and the red (right) curve corresponds to  $n = 100 \times 10^3$ . As stated in Sec. VB,  $\rho_{\text{irre}}^{(N)}(t)$  is well described by the *ansatz* in Eq. (16).

we have thus identified three time scales, i.e.,  $\tau_{\text{int,irre}}^{(\mathcal{N})}, \tau_2, \tau_{\text{intra}}$ , which scale as  $n^{1/2}$ .

## VI. DISCUSSION

We constructed an irreversible MCMC worm algorithm for the zero-field ferromagnetic Ising model via the lifting technique. Since it is not obvious how to lift the standard P-S worm algorithm to generate ballistic motion for the number of occupied edges, we first constructed an alternative worm algorithm with different proposals (B-S type) and lifted this chain. We emphasize that this construction can also be used to design lifted worm algorithms for other important models in statistical mechanics such as the  $XY$  model. The  $XY$  model shares many universal properties of the Bose-Hubbard model [40], which is actively studied in ultracold atom physics. We also emphasize that the lifted worm algorithm for the Ising model can be implemented by changing only a few code lines in the B-S-type worm algorithm.

We studied the dynamical critical behavior of the number of occupied edges  $\mathcal{N}$  on both the complete graph and toroidal grids in various dimensions. On the complete graph, we numerically established  $z_{\text{int,irre},\mathcal{N}} \approx -0.5$  for the lifted worm algorithm, while we found  $z_{\text{int,B-S},\mathcal{N}} \approx 0$  for the corresponding reversible counterpart. It is interesting to observe that the mixing time of the Swendsen-Wang algorithm scales as  $t_{\text{mix}}/n \sim n^{1/4}$  [41]. On the torus, we numerically established that the lifted chain leads to a constant improvement, with larger improvements for higher dimensions. Even though we did not find a dynamic improvement for  $d \leq 5$ , it is an open question if  $\mathcal{N}$  exhibits critical speeding-up on higher-dimensional tori.

We also studied the return time to the Eulerian subspace  $\mathcal{C}_0$ . We find that while the mean is the same in both the lifted and unlifted processes, the variance is a constant factor smaller in the lifted version.

Finally, we note that a recent study [42] constructed an alternative irreversible worm algorithm which is based on the so-called geometric allocation approach [12]. Similar to our findings, this irreversible worm algorithm does not improve the dynamical critical exponent for  $\mathcal{N}$  on the three-dimensional torus, but leads to a significant constant improvement.

## ACKNOWLEDGMENTS

We would like to thank Zongzheng Zhou for fruitful discussions. E.M.E. is grateful to Nikolaos G. Fytas for fruitful discussions regarding various finite-size scaling *Ansätze*. This work was supported under the Australian Research Council's Discovery Projects funding scheme (Project No. DP140100559). It was undertaken with the assistance of resources from the National Computational Infrastructure (NCI), which is supported by the Australian Government. Furthermore, we would like to acknowledge the Monash eResearch Centre and eSolutions-Research Support Services through the use of the Monash Campus HPC Cluster. Y.D. thanks the National Natural Science Foundation of China for their support under Grant No. 11625522 and the Fundamental Research Funds for the Central Universities under Grant No. 2340000034. Y.D. also thanks MOST for their support under Grant No. 2016YFA0301604.

## APPENDIX A: ESTIMATION WITH THE MADRAS-SOKAL AUTOMATIC WINDOWING ALGORITHM AND SUPPRESSED SLOW MODES

A widely used numerical method to estimate the integrated autocorrelation time of a weakly stationary time series [43] is to consider a truncated version of (13), i.e.,

$$\widehat{\tau}_{\text{int}}^{(\mathcal{N})}(n) := \frac{1}{2} + \sum_{t=1}^n \rho^{(\mathcal{N})}(t). \quad (\text{A1})$$

The reason for truncating the time series is well known [38,43]: The standard deviation of (A1) is  $O(\sqrt{n/M})$  and, hence, (for fixed  $M$ ) grows with  $n$ . The choice of  $n$  is, thus, a compromise between bias and standard error. The Madras-Sokal automatic windowing method determines  $n$  self-consistently as the smallest positive integer  $n$  that fulfills (numerically)

$$c\tau_{\text{int}}^{(\mathcal{N})}(n) \leq n,$$

where  $c$  is a free real parameter, typically manually chosen to lie in  $[6, 10]$ ; cf. [38] for a justification of this particular choice of  $c$ .

While analyzing the time series for  $\mathcal{N}$  for the lifted worm algorithm in  $d \geq 4$ , we made the observation that the choice  $c \in [6, 10]$  significantly underestimates the integrated autocorrelation time. Intuitively, what we implement with (A1) is a “discrete integration,” which we utilize to extract the decay constant (the analog of the relation  $\int_0^\infty dx e^{-x/\tau} = \tau$  in a continuous setting). However, in our setting, we only obtain a significant contribution from  $\alpha_1 e^{-t/\tau_1}$  to (A1) when including lags for which the contribution of  $\alpha_2 e^{-t/\tau_2}$  to  $\rho^{(\mathcal{N})}(t)$  is too small to be separated from noise. The combination of small amplitude and large-time scale, thus, requires adjusting  $c$ . In Fig. 6, we illustrate this for the particular choice of  $L = 56$  of the lifted worm algorithm in  $d=5$ .

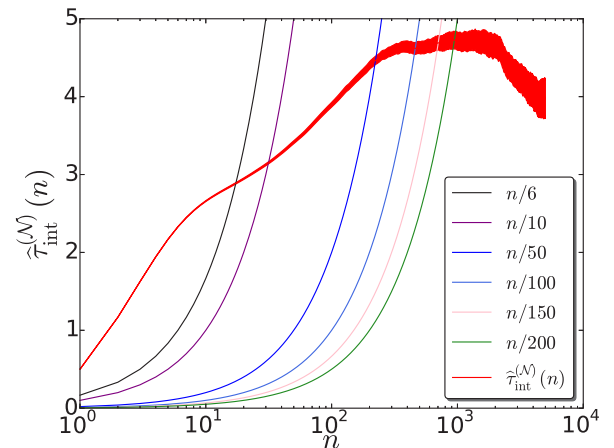


FIG. 6. Dependence on the cutoff time  $n$  of the truncated integration autocorrelation time  $\widehat{\tau}_{\text{int}}^{(\mathcal{N})}(n)$  defined in (A1) for the lifted worm algorithm in five dimensions and  $L = 56$  at criticality. The thick red line shows  $\widehat{\tau}_{\text{int}}^{(\mathcal{N})}(n)$  averaged over 100 independent runs, whereas the surrounding region corresponds to  $\pm 1\sigma$ . The thin lines correspond to the curves  $n/c$  with increasing  $c$  from left ( $c = 6$ ) to right ( $c = 200$ ). One clearly sees how  $n < 100$  underestimates  $\widehat{\tau}_{\text{int}}^{(\mathcal{N})}(n)$  and much larger  $n$  are required to capture contributions corresponding to the suppressed mode.

## APPENDIX B: LEAST-SQUARES FITTING A WEIGHTED EXPONENTIAL ANSATZ

While fitting the normalized autocorrelation function for  $\mathcal{N}$  in dimensions four and five, we made the observation that it is numerically very hard to obtain reliable estimates of the parameters involved in a functional model of the two-time scale behavior using the method of least squares. We, therefore, manually analyzed the least-squares conditions. As result of this, we were able to reduce the free parameters which we need to numerically minimize from 4 to 2. This significantly improved the numerical precision and stability of the fitting procedure. To this end, our model has free parameters  $\alpha_1 > 0, \alpha_2 > 0, 1 > \lambda_1 > 0, 1 > \lambda_2 > 0$ , and we would like to minimize the sum of the first  $T_{\max}$  squared residuals,

$$r_k^2 \equiv [\alpha_1 \lambda_1^k + \alpha_2 \lambda_2^k - \rho_{\mathcal{N}}(k)]^2.$$

That is, our goal is to minimize the function  $\Theta$  defined as

$$\Theta(\alpha_1, \alpha_2, \lambda_1, \lambda_2) \equiv \sum_{k=0}^{T_{\max}-1} r_k^2.$$

A necessary condition for the minimum is

$$\frac{\partial \Theta}{\partial \alpha_1} = \frac{\partial \Theta}{\partial \alpha_2} = \frac{\partial \Theta}{\partial \lambda_1} = \frac{\partial \Theta}{\partial \lambda_2} = 0.$$

We start by evaluating the partial derivatives with respect to  $\alpha_1$  and  $\alpha_2$  and, as a consequence of this, obtain the following set

of two linear equations:

$$\begin{aligned} \alpha_1 P(\lambda_1) + \alpha_2 S(\lambda_1, \lambda_2) &= Q(\lambda_1), \\ \alpha_1 S(\lambda_1, \lambda_2) + \alpha_2 P(\lambda_2) &= Q(\lambda_2). \end{aligned}$$

Here we defined the following three polynomials over  $(0,1)$  or  $(0,1)^2$ :

$$\begin{aligned} P(\lambda) &\equiv \sum_{k=0}^{T_{\max}-1} \lambda^{2k} = \frac{\lambda^{2T_{\max}} - 1}{\lambda^2 - 1}, \\ Q(\lambda) &\equiv \sum_{k=0}^{T_{\max}-1} \lambda^k \rho_{\mathcal{N}}(k), \quad S(\lambda_1, \lambda_2) \equiv \sum_{k=0}^{T_{\max}-1} \lambda_1^k \lambda_2^k. \end{aligned}$$

Note that  $S$  is symmetric in  $\lambda_1, \lambda_2$ . The above set of linear equations has the unique solution

$$\alpha_1(\lambda_1, \lambda_2) = \frac{S(\lambda_1, \lambda_2)Q(\lambda_2) - Q(\lambda_1)P(\lambda_2)}{S(\lambda_1, \lambda_2)^2 - P(\lambda_1)P(\lambda_2)},$$

and  $\alpha_2(\lambda_1, \lambda_2)$  is obtained by replacing  $1 \rightarrow 2, 2 \rightarrow 1$  in the right-hand side above. Thus, this allows us to substitute  $\alpha_1, \alpha_2$  as a function of  $\lambda_1, \lambda_2$  in  $\Theta$  and minimize  $\Theta$  only with respect to  $\lambda_1, \lambda_2$ , with the constraints that  $\lambda_1, \lambda_2 \in (0,1)$ . We remark that completely analogous arguments can be used to show that a similar improvement can be achieved for a general *ansatz*  $\sum_{j=1}^n \alpha_j \lambda_j^t$ , where one reduces the parameters to numerically minimize from  $2n$  to  $n$ . The calculations done here generalize the method very recently proposed in [44] for a single exponential *ansatz*.

- 
- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, 2016).
- [2] B. Efron and T. Hastie, *Computer Age Statistical Inference* (Cambridge University Press, Cambridge, 2016), Vol. 5.
- [3] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equations of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087 (1953).
- [4] W. Janke, Monte Carlo methods in classical statistical physics, in *Computational Many-Particle Physics*, Lecture Notes in Physics, Vol. 739 (Springer-Verlag, Berlin, 2008).
- [5] N. Prokof'ev and B. Svistunov, Worm Algorithms for Classical Statistical Models, *Phys. Rev. Lett.* **87**, 160601 (2001).
- [6] M. Boninsegni, Nikolay Prokof'ev, and B. Svistunov, Worm Algorithm for Continuous-space Path Integral Monte Carlo Simulations, *Phys. Rev. Lett.* **96**, 070601 (2006).
- [7] U. Wolff, Collective Monte Carlo Updating for Spin Systems, *Phys. Rev. Lett.* **62**, 361 (1989).
- [8] R. H. Swendsen and J.-S. Wang, Nonuniversal Critical Dynamics in Monte Carlo Simulations, *Phys. Rev. Lett.* **58**, 86 (1987).
- [9] Y. Deng, T. M. Garoni, and A. D. Sokal, Dynamic Critical Behavior of the Worm Algorithm for the Ising Model, *Phys. Rev. Lett.* **99**, 110601 (2007).
- [10] U. Wolff, Simulating the all-order strong coupling expansion I: Ising model demo, *Nucl. Phys. B* **810**, 491 (2009).
- [11] A. Collecchio, T. M. Garoni, T. Hyndman, and D. Tokarev, The worm process for the Ising model is rapidly mixing, *J. Stat. Phys.* **164**, 1082 (2016).
- [12] H. Suwa and S. Todo, Markov Chain Monte Carlo Method without Detailed Balance, *Phys. Rev. Lett.* **105**, 120603 (2010).
- [13] T. P. Hayes and A. Sinclair, Liftings of Tree-structured Markov Chains, in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, edited by M. Serna, R. Shaltiel, K. Jansen, and J. Rolim, Lecture Notes in Computer Science, Vol. 6302 (Springer, Berlin 2010).
- [14] K. Turitsyn, M. Chertkov, and M. Vucelja, Irreversible Monte Carlo algorithms for efficient sampling, *Physica D* **240**, 410 (2011).
- [15] Y. Sakai and K. Hukushima, Dynamics of one-dimensional Ising model without detailed balance condition, *J. Phys. Soc. Japan* **82**, 064003 (2013).
- [16] H. C. M. Fernandes and M. Weigel, Non-reversible Monte Carlo simulations of spin models, *Comput. Phys. Commun.* **182**, 1856 (2011).
- [17] E. P. Bernard, W. Krauth, and D. B. Wilson, Event-chain Monte Carlo algorithms for hard-sphere systems, *Phys. Rev. E* **80**, 056704 (2009).
- [18] M. Engel, J. A. Anderson, S. C. Glotzer, M. Isobe, E. P. Bernard, and W. Krauth, Hard-disk equation of state: First-order

- liquid-hexatic transition in two dimensions with three simulation methods, *Phys. Rev. E* **87**, 042134 (2013).
- [19] S. C. Kapfer and W. Krauth, Soft-Disk Melting: From Liquid-Hexatic Coexistence to Continuous Transitions, *Phys. Rev. Lett.* **114**, 035702 (2015).
- [20] M. Michel, J. Mayer, and W. Krauth, Event-chain Monte Carlo for classical continuous spin models, *Europhys. Lett.* **112**, 20003 (2015).
- [21] Y. Nishikawa, M. Michel, W. Krauth, and K. Hukushima, Event-chain algorithm for the Heisenberg model: Evidence for  $z \sim 1$  dynamic scaling, *Phys. Rev. E* **92**, 063306 (2015).
- [22] H. Hu, X. Chen, and Y. Deng, Irreversible Markov chain Monte Carlo algorithm for self-avoiding walk, *Front. Phys.* **12**, 120503 (2017).
- [23] P. Diaconis, S. Holmes, and R. M. Neal, Analysis of a non-reversible Markov chain sampler, *Ann. Appl. Probab.* **10**, 726 (2000).
- [24] P. Diaconis, Some things we've learned (about Markov chain Monte Carlo), *Bernoulli* **19**, 1294 (2013).
- [25] Y. Deng, T. M. Garoni, and A. D. Sokal, Critical Speeding-Up in the Local Dynamics of the Random-Cluster Model, *Phys. Rev. Lett.* **98**, 230602 (2007).
- [26] Y. Pomeau and M. Le Berre, Critical speed-up vs critical slow-down: A new kind of relaxation oscillation with application to stick-slip phenomena, [arXiv:1107.3331](https://arxiv.org/abs/1107.3331).
- [27] P. H. Lundow and K. Markström, Finite size scaling of the 5D Ising model with free boundary conditions, *Nucl. Phys. B* **889**, 249 (2014).
- [28] M. Wittmann and A. P. Young, Finite-size scaling above the upper critical dimension, *Phys. Rev. E* **90**, 062137 (2014).
- [29] E. Flores-Sola, B. Berche, R. Kenna, and M. Weigel, Role of Fourier Modes in Finite-Size Scaling above the Upper Critical Dimension, *Phys. Rev. Lett.* **116**, 115701 (2016).
- [30] J. Grimm, E. M. Elçi, Z. Zhou, T. M. Garoni, and Y. Deng, Geometric Explanation of Anomalous Finite-Size Scaling in High Dimensions, *Phys. Rev. Lett.* **118**, 115701 (2017).
- [31] C. J. Thompson, *Mathematical Statistical Mechanics* (Princeton University Press, Princeton, NJ, 1979).
- [32] A. Berretti and A. D. Sokal, New Monte Carlo method for the self-avoiding walk, *J. Stat. Phys.* **40**, 483 (1985).
- [33] M. Vucelja, Lifting - A nonreversible Markov chain Monte Carlo algorithm, *Am. J. Phys.* **84**, 958 (2016).
- [34] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics* (Academic/Harcourt Brace Jovanovich, London, 1982).
- [35] Y. Deng and H. W. J. Blöte, Simultaneous analysis of several models in the three-dimensional Ising universality class, *Phys. Rev. E* **68**, 036125 (2003).
- [36] P. H. Lundow and K. Markström, Critical behavior of the Ising model on the four-dimensional cubic lattice, *Phys. Rev. E* **80**, 031104 (2009).
- [37] P. Young, *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask* (Springer, New York, 2015).
- [38] A. Sokal, in *Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms*, edited by C. DeWitt-Morette, P. Cartier, and A. Folacci, Functional Integration, NATO ASI Series (Series B: Physics), Vol. 361 (Springer, Boston, MA, 1997).
- [39] N. Madras and G. Slade, *The Self-Avoiding Walk* (Birkhäuser, Boston, 1996).
- [40] B. Svistunov, E. Babaev, and N. Prokof'ev, *Superfluid States of Matter* (CRC, Boca Raton, FL, 2015).
- [41] Y. Long, Mixing time of the Swendsen-Wang dynamics on the complete graph and trees, Ph.D. thesis, University of California Berkeley, 2009.
- [42] H. Suwa, Directed worm algorithm with geometric optimization, [arXiv:1703.03136](https://arxiv.org/abs/1703.03136).
- [43] M. B. Priestley, *Spectral Analysis and Time Series* (Academic, Cambridge, MA, 1981).
- [44] Y. Fang, Y. Cao and R. D. Skeel, Quasi-reliable estimates of effective sample size, [arXiv:1705.03831](https://arxiv.org/abs/1705.03831).