Analysis of dynamic Ising model by a variational approximate method: Estimation of transfer entropy

Hirohito Kiwata[*]

Division of Natural Science, Osaka Kyoiku University, Kashiwara, Osaka 582-8582, Japan

(Received 21 November 2019; revised manuscript received 3 March 2020; accepted 5 March 2020; published 3 April 2020)

Transfer entropy is an important index for investigating the causal relationship between random variables. It is known that a phase transition can be detected by transfer entropy. In this work, I propose a method for obtaining the transfer entropy by applying the variational approximation method to a dynamic Ising model. Our method is also effective for dynamics that do not satisfy the detailed balance condition. Our method finds the transfer entropy by solving a transcendental equation. When compared with the transfer entropy obtained by Monte Carlo sampling, it was found that they qualitatively matched near the transition point; moreover, the match was obtained with good accuracy in the region deviated from the transition point.

DOI: 10.1103/PhysRevE.101.042102

I. INTRODUCTION

The problem of finding hidden patterns in datasets is fundamental in natural science and has a long history. Recently, as the performance of computers has improved, large datasets are easily obtained. Machine learning can automatically find regularity in a dataset through a calculation algorithm and classify the dataset into different categories using the regularity [1,2]. In actual application, to make effective use of the obtained dataset, the original dataset is converted into a new variable by preprocessing to make the problem easy to handle. This preprocessing stage is called feature extraction. In machine learning, features are extracted from dataset, and a machine is learned from the features. This machine learning method has been applied to various fields of natural science, and the physics field is no exception. In particular, the analysis of phase-transition phenomena using a machine learning method has been actively performed [3–15].

The correlation coefficient is a standard method of examining the relationship between two variables of a measured dataset such as $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(M)}, y^{(M)})\}$. In addition, as an information-theoretical method, the relationship can also be examined by mutual information analysis. In these cases, x and y are treated as random variables because the dataset includes stochastic properties, such as noise, during measurement. Although the dependency between variables can be known from the correlation coefficient or the mutual information, the dependency is symmetric and does not consider the direction from one variable to the other. To examine the causal relationship between the dataset variables, i.e., "event X has occurred because of event Y," it is necessary to know the temporal context. Therefore, instead of looking at the interdependence between dataset variables in an equilibrium state, like the correlation coefficient and mutual information do, it is necessary to examine the dependence of variables on the temporal change of their observed values.

Transfer entropy is known as an information-theoretic quantification of the causal effect from Y to X [16,17]. Assuming a time-series dataset $\{(x(t), y(t)), (x(t-1), y(t-1)), \dots, (x(0), y(0))\}$ (t is the current time), if the observation $x(t+1)$ at the subsequent time $t+1$ depends on the history of X so far, the probability distribution of obtaining $x(t+1)$ is $P(x(t+1)|x(t), \dots, x(0))$. Considering the dependence of Y on X, the above conditional probability distribution becomes $P(x(t+1)|x(t), \dots, x(0), y(t), \dots, y(0))$. If $x(t+1)$ has no effect by Y and is determined solely by X's own past observations, then the condition $P(x(t+1)|x(t), \dots, x(0), y(t), \dots, y(0)) = P(x(t+1)|x(t), \dots, x(0))$ is satisfied. Conversely, if Y has some effect on X (Y has a causal relationship to X), then $x(t+1)$ is determined depending on the value of Y observed in the past. The above condition regarding the probability distribution is not satisfied. Transfer entropy is a quantity used to evaluate whether these probability distributions are equal or unequal in terms of the Kullback-Leibler divergence. It is known that the distribution of the number of states (number of probabilistic events) must be obtained from an actual dataset by sampling, and a sufficiently long time-series dataset is necessary to obtain a reliable conclusion. Thus, for transfer entropy, whether or not there is a sufficient amount of datasets to construct a probability distribution is a criterion for determining whether or not the transfer entropy can be appropriately applied.

The Ising model is a prototype that investigates the phase-transition phenomenon of magnetic materials and has been studied in various ways. Unique phenomena occur at the phase-transition point, such as divergence of the magnetic susceptibility or the correlation length. It has been known, from the information-theoretical standpoint, that mutual information shows the maximum value at the phase-transition point [18]. Similarly, transfer entropy shows a maximum value at

[*] kiwata@cc.osaka-kyoiku.ac.jp

the transition point [19,20]. Thus, attempts to understand the phase transition in terms of the information theory are progressing. However, these studies have been conducted on the equilibrium state or in the process of reaching the equilibrium state and do not purely reveal the causal relationship of time-series data. The equilibrium state is reached by repeating a transition in the case of a dynamic process that satisfies the detailed balance condition. In this study, we do not necessarily assume the detailed balance condition but aim to obtain the transfer entropy for general dynamic processes. In general, the dynamic equation cannot be solved analytically, so we treat the dynamic equation approximately using the recently developed variational approximation [21–23]. By using our method, we can estimate the transfer entropy by solving the transcendental equations without sampling the dataset.

II. MODEL AND METHOD

A. Dynamic model

We consider a graph composed of N vertices and edges connecting some pairs of vertices. A set of edges is denoted by $\mathcal{E}$. A discrete random variable s_i is assigned to the ith vertex. The discrete random variable s_i is defined by the set of values it can take, $\mathcal{X}$. For simplicity, we restrict $\mathcal{X} = \{+1, -1\}$. A set of discrete random variables is defined as $\boldsymbol{s} = \{s_1, s_2, \ldots, s_N\}$. By interacting with adjacent vertices, each s_i changes with time. To demonstrate s_i depending on time, we explicitly express $s_i(t)$ and similarly $\boldsymbol{s}(t) = \{s_1(t), s_2(t), \ldots, s_N(t)\}$. We represent the probability distribution of $\boldsymbol{s}(t)$ as $P[\boldsymbol{s}(t)]$ and the joint probability distribution of $\boldsymbol{s}(t+\Delta)$ and $\boldsymbol{s}(t)$ as $P(\boldsymbol{s}(t+\Delta), \boldsymbol{s}(t))$, respectively. Although we use the same symbol, P, for all probability distributions, the probability distributions which have different arguments as random variables are different from each other. The quantity Δ is an increment of time and the value of it depends on update rules. The relationship between $P(\boldsymbol{s}(t+\Delta), \boldsymbol{s}(t))$ and $P[\boldsymbol{s}(t)]$ is given by

$$P(\boldsymbol{s}(t+\Delta), \boldsymbol{s}(t)) = W(\boldsymbol{s}(t+\Delta)|\boldsymbol{s}(t))P[\boldsymbol{s}(t)], \tag{1}$$

where $W(\boldsymbol{s}(t+\Delta)|\boldsymbol{s}(t))$ is a transition probability. Time is measured in units of the update timescale. The formula of $W(\boldsymbol{s}(t+\Delta)|\boldsymbol{s}(t))$ depends on the model dynamics. Marginalizing $\boldsymbol{s}(t)$ in Eq. (1), we obtain the following formula;

$$\begin{aligned} P[\boldsymbol{s}(t+\Delta)] &= \sum_{\boldsymbol{s}(t)\in\mathcal{X}^N} P(\boldsymbol{s}(t+\Delta), \boldsymbol{s}(t)) \\ &= \sum_{\boldsymbol{s}(t)\in\mathcal{X}^N} W(\boldsymbol{s}(t+\Delta)|\boldsymbol{s}(t))P[\boldsymbol{s}(t)]. \end{aligned} \tag{2}$$

We consider two types of dynamics of the system, synchronous and asynchronous updates [24–29]. In the asynchronous update, one vertex is chosen and the random variable on the vertex is changed by a prescribed probability. In the synchronous update, the discrete random variables are changed simultaneously following a prescribed probability. In the asynchronous update, the transition probability is given by

$$\begin{aligned} &w^{(a)}(s_i(t+\gamma\delta t)|s_i(t), s_{\partial i}(t)) \\ &\quad = (1-\gamma\delta t)\delta_{s_i(t+\gamma\delta t), s_i(t)} \\ &\qquad + \gamma\delta t \frac{\exp\left[s_i(t+\gamma\delta t)\sum_{j\in\partial i} J_{ij}s_j(t)\right]}{2\cosh\left[\sum_{j\in\partial i} J_{ij}s_j(t)\right]}, \end{aligned} \tag{3}$$

where δt is the discretized time step size, γ is a constant with dimension of inverse time, $\delta_{x,y}$ is the Kronecker delta, and J_{ij} is a coupling parameter between the ith and jth vertices [30]. The sign ∂i stands for a set of vertices neighboring the ith vertex, i.e., $\partial i = \{j|J_{ij} \neq 0\}$. In the case of $J_{ij} = 0$, no edge exists between the ith vertex and jth one. In the case of a directed graph, the coupling parameter is not symmetric, i.e., $J_{ij} \neq J_{ji}$. The superscript (a) stands for the asynchronous update. The quantity $\gamma\delta t$ corresponds to Δ. When $\delta t \to 0$, we obtain a differential equation for the expectation of $s_i(t)$. The expected change with time is obtained by solving the differential equation. However, we will proceed analysis in a different way. We set $\gamma\delta t = 1/N$. Substituting Eq. (3) for Eq. (1), we obtain

$$\begin{aligned} &P\left(\boldsymbol{s}\left(t+\frac{1}{N}\right), \boldsymbol{s}(t)\right) \\ &\quad = \prod_{i=1}^{N} w^{(a)}\left(s_i\left(t+\frac{1}{N}\right)\middle| s_i(t), s_{\partial i}(t)\right)P[\boldsymbol{s}(t)], \end{aligned} \tag{4}$$

and substituting Eq. (3) for Eq. (2), we similarly obtain

$$\begin{aligned} &P\left[\boldsymbol{s}\left(t+\frac{1}{N}\right)\right] \\ &\quad = \sum_{\boldsymbol{s}(t)\in\mathcal{X}^N} \prod_{i=1}^{N} w^{(a)}\left(s_i\left(t+\frac{1}{N}\right)\middle| s_i(t), s_{\partial i}(t)\right)P[\boldsymbol{s}(t)]. \end{aligned} \tag{5}$$

In Appendix A, we prove that Eq. (5) coincides with the asynchronous update of the Glauber dynamics [31]. Marginalizing $\boldsymbol{s}(t+\frac{1}{N})$ and $\boldsymbol{s}(t)$ except $s_i(t+\frac{1}{N})$, $s_i(t)$, and $s_{\partial i}(t)$ in Eq. (4), we obtain

$$\begin{aligned} &P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_{\partial i}(t)\right) \\ &\quad = w^{(a)}\left(s_i\left(t+\frac{1}{N}\right)\middle| s_i(t), s_{\partial i}(t)\right)P(s_i(t), s_{\partial i}(t)), \end{aligned} \tag{6}$$

where we use the following condition;

$$\sum_{s_i(t+\frac{1}{N})\in\mathcal{X}} w^{(a)}\left(s_i\left(t+\frac{1}{N}\right)\middle| s_i(t), s_{\partial i}(t)\right) = 1. \tag{7}$$

Compared with the asynchronous update, the synchronous update is much easier to handle. In the case of the synchronous update, the transition probability is given by

$$w^{(s)}(s_i(t+1)|s_{\partial i}(t)) = \frac{\exp\left[s_i(t+1)\sum_{j\in\partial i} J_{ij}s_j(t)\right]}{2\cosh\left[\sum_{j\in\partial i} J_{ij}s_j(t)\right]}. \tag{8}$$

The superscript (s) stands for the synchronous update. The difference between the asynchronous and synchronous updates is that the transition probability depends on $s_i(t)$ or not. The time to update all vertices simultaneously is taken as the unit time. Substituting (8) for (1), we obtain the following formula:

$$P(\boldsymbol{s}(t+1), \boldsymbol{s}(t)) = \prod_{i=1}^{N} w^{(s)}(s_i(t+1)|s_{\partial i}(t))P[\boldsymbol{s}(t)]. \quad (9)$$

Marginalizing $\boldsymbol{s}(t+1)$ and $\boldsymbol{s}(t)$ except $s_i(t+1)$ and $s_{\partial i}(t)$, the following formula:

$$P(s_i(t+1), s_{\partial i}(t)) = w^{(s)}(s_i(t+1)|s_{\partial i}(t))P[s_{\partial i}(t)], \quad (10)$$

is derived by quoting the following condition:

$$\sum_{s_i(t+1)\in\mathcal{X}} w^{(s)}(s_i(t+1)|s_{\partial i}(t)) = 1. \quad (11)$$

As shown later, whether or not the joint probability distribution depends on $s_i(t)$ greatly affects the transfer entropy.

We mention probability distributions after performing the updates infinitely. The detailed balance condition is met only when the coupling parameter is symmetric, i.e., $J_{ij} = J_{ji}$, and the update is asynchronous. In the case of an update which satisfies the condition of the detailed balance, the probability distribution converges to the Boltzmann distribution. In other cases, it is not known exactly what the probability distribution converges to. However, our method is also effective for dynamics in which the probability distribution does not converge to the Boltzmann distribution.

B. Transfer entropy

The mutual information in the information theory examines interdependencies between random variables that do not change with time. The mutual information between Ising spins, which are random variables of the Ising model in the equilibrium state, shows that it behaves singularly at the critical point. This is similar in behavior to the magnetic susceptibility. Transfer entropy examines the dependence between random variables indicated by the time-varying observations. Usually, transfer entropy is used to examine causality, such as event X occurring as a result of event Y. In this study, the transfer entropy is applied to the dynamic Ising model to investigate the anomaly of the system. However, in this study, it is not assumed that the system approaches an equilibrium state by updating the system due to dynamics. Following Ref. [19], we define the transfer entropy as

$$\mathcal{T} \equiv \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \mathcal{T}_{s_j\to s_i}, \quad (12)$$

where the sum is performed for all edges and $\mathcal{T}_{s_j\to s_i}$ is defined by

$$\mathcal{T}_{s_j\to s_i} = \sum_{s_i(t+\Delta)\in\mathcal{X},}\ \sum_{s_i(t)\in\mathcal{X},}\ \sum_{s_j(t)\in\mathcal{X}} P(s_i(t+\Delta), s_i(t), s_j(t)) \times \ln\left\{\frac{P(s_i(t+\Delta)|s_i(t), s_j(t))}{P(s_i(t+\Delta)|s_i(t))}\right\}. \quad (13)$$

Here transfer entropy is defined so as to find the causal relationship that $s_j(t)$, where j is adjacent to i, has to $s_i(t+\Delta)$. To estimate Eq. (13), we have to evaluate the joint probability distributions by sampling. Because Eq. (13) is a quantity of order $O(\frac{1}{N})$ in the case of the asynchronous update, it is difficult to estimate it by sampling. In Ref. [19], the sampling is performed after Eq. (13) has been transformed. As a result, the sampling is performed in the equilibrium state.

C. Approximate method

To estimate the transfer entropy (12) with Eq. (13), we have to evaluate the joint probability distributions and the conditional probability distribution. Because the conditions are different between update rules, the case of asynchronous update and the case of synchronous update will be explained separately.

1. *Asynchronous update*

Transfer entropy (13) is transformed into

$$\mathcal{T}^{(a)}_{s_j\to s_i} = \sum_{s_i(t+\frac{1}{N})\in\mathcal{X},}\ \sum_{s_i(t)\in\mathcal{X},}\ \sum_{s_j(t)\in\mathcal{X}} P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right) \times \ln\left\{\frac{P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right)}{P(s_i(t), s_j(t))}\ \frac{P[s_i(t)]}{P\left(s_i\left(t+\frac{1}{N}\right), s_i(t)\right)}\right\}, \quad (14)$$

where we set Δ to $\frac{1}{N}$. The joint probability distributions which constitute the transfer entropy are evaluated by marginalizing the probability distribution, Eq. (6). However, because it is impossible to estimate the joint probability distribution, we adopt the cluster variational method as an approximation. In the cluster variational method, the simplest approximation is called a star approximation in Ref. [21]. In the star approximation, the joint probability distribution of right-hand side on Eq. (6) is factorized by a single-vertex probability distribution as follows:

$$P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_{\partial i}(t)\right) = w^{(a)}\left(s_i\left(t+\frac{1}{N}\right)|s_i(t), s_{\partial i}(t)\right)P[s_i(t)]\prod_{j\in\partial i} P[s_j(t)]. \quad (15)$$

Summing over $s_i(t+\frac{1}{N})$ in Eq. (15), we obtain

$$P(s_i(t), s_{\partial i}(t)) = \sum_{s_i(t+\frac{1}{N})\in\mathcal{X}} P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_{\partial i}(t)\right) = P[s_i(t)]\prod_{j\in\partial i} P[s_j(t)], \quad (16)$$

where Eq. (7) is utilized. The formula shows that the probability distribution of random variables at the same time is decomposed into probability distributions of individual variables. As shown in Eq. (14), we have to estimate the probability distributions, $P[s_i(t)]$, $P(s_i(t+\frac{1}{N}), s_i(t))$, and $P(s_i(t+\frac{1}{N}), s_i(t), s_j(t))$. In the star approximation, the probability distribution $P(s_i(t), s_j(t))$ is factorized as $P[s_i(t)]P[s_j(t)]$, because $s_i(t)$ and $s_j(t)$ are on the same time. Considering that $s_i(t)$ is binary, the above probability distributions can be expressed as follows:

$$P[s_i(t)] =\tfrac{1}{2}[1+s_i(t)m_i(t)], \tag{17}$$

$$P\left(s_i\left(t+\frac{1}{N}\right), s_i(t)\right) =\frac{1}{4}\left[1+s_i\left(t+\frac{1}{N}\right)m_i\left(t+\frac{1}{N}\right)+s_i(t)m_i(t)+s_i\left(t+\frac{1}{N}\right)s_i(t)c_i\left(t+\frac{1}{N},t\right)\right], \tag{18}$$

$$\begin{aligned}P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right) = \frac{1}{8}\Bigg[&1+s_i\left(t+\frac{1}{N}\right)m_i\left(t+\frac{1}{N}\right)+s_i(t)m_i(t)+s_j(t)m_j(t)+s_i\left(t+\frac{1}{N}\right)s_i(t)c_i\left(t+\frac{1}{N},t\right)\\&+s_i\left(t+\frac{1}{N}\right)s_j(t)c'_{ij}\left(t+\frac{1}{N},t\right)+s_i(t)s_j(t)m_i(t)m_j(t)+s_i\left(t+\frac{1}{N}\right)s_i(t)s_j(t)t_{ij}\left(t+\frac{1}{N},t\right)\Bigg],\end{aligned} \tag{19}$$

where $m_i(t)$, $c_i(t+\frac{1}{N},t)$, $c'_{ij}(t+\frac{1}{N},t)$, and $t_{ij}(t+\frac{1}{N},t)$ are defined as

$$m_i(t) = \sum_{s_i(t)\in\mathcal{X}} s_i(t)P[s_i(t)], \tag{20}$$

$$c_i\left(t+\frac{1}{N},t\right) = \sum_{s_i(t+\frac{1}{N})\in\mathcal{X},}\ \sum_{s_i(t)\in\mathcal{X}} s_i\left(t+\frac{1}{N}\right)s_i(t)P\left(s_i\left(t+\frac{1}{N}\right), s_i(t)\right), \tag{21}$$

$$c'_{ij}\left(t+\frac{1}{N},t\right) = \sum_{s_i(t+\frac{1}{N})\in\mathcal{X},}\ \sum_{s_i(t)\in\mathcal{X},}\ \sum_{s_j(t)\in\mathcal{X}} s_i\left(t+\frac{1}{N}\right)s_j(t)P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right), \tag{22}$$

$$t_{ij}\left(t+\frac{1}{N},t\right) = \sum_{s_i(t+\frac{1}{N})\in\mathcal{X},}\ \sum_{s_i(t)\in\mathcal{X},}\ \sum_{s_j(t)\in\mathcal{X}} s_i\left(t+\frac{1}{N}\right)s_i(t)s_j(t)P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right), \tag{23}$$

respectively. The derivation of Eqs. (18) and (19) appears in Appendix B. The probability distributions must satisfy the following marginalization constrains:

$$P[s_i(t)] = \sum_{s_i(t+\frac{1}{N})\in\mathcal{X}} P\left(s_i\left(t+\frac{1}{N}\right), s_i(t)\right), \tag{24}$$

$$P\left(s_i\left(t+\frac{1}{N}\right), s_i(t)\right) = \sum_{s_j(t)\in\mathcal{X}} P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right). \tag{25}$$

We can recognize that the probability distributions, Eqs. (17), (18), and (19), satisfy the marginalization constrains, Eqs. (24) and (25).

To obtain the renewal rule of $m_i(t)$, we multiply Eq. (15) by $s_i(t+\frac{1}{N})$ and sum over all the random variables, and then we obtain

$$m_i\left(t+\frac{1}{N}\right) = \left(1-\frac{1}{N}\right)m_i(t)+\frac{1}{N}\sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}}\tanh\left[\sum_{k\in\partial i}J_{ik}s_k(t)\right]\prod_{k'\in\partial i}\frac{1+s_{k'}(t)m_{k'}(t)}{2}. \tag{26}$$

In a similar way, multiplying Eq. (15) by $s_i(t+\frac{1}{N})$ and $s_i(t)$ and summing over all the random variables, we obtain

$$c_i\left(t+\frac{1}{N},t\right) = \left(1-\frac{1}{N}\right)+\frac{1}{N}m_i(t)\sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}}\tanh\left[\sum_{k\in\partial i}J_{ik}s_k(t)\right]\prod_{k'\in\partial i}\frac{1+s_{k'}(t)m_{k'}(t)}{2}. \tag{27}$$

Similarly, we obtain

$$c'_{ij}\left(t+\frac{1}{N},t\right) = \left(1-\frac{1}{N}\right)m_i(t)m_j(t)+\frac{1}{N}\sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}}\tanh\left[\sum_{k\in\partial i}J_{ik}s_k(t)\right]s_j(t)\prod_{k'\in\partial i}\frac{1+s_{k'}(t)m_{k'}(t)}{2}, \tag{28}$$

$$t_{ij}\left(t+\frac{1}{N},t\right) = \left(1-\frac{1}{N}\right)m_j(t)+\frac{1}{N}m_i(t)\sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}}\tanh\left[\sum_{k\in\partial i}J_{ik}s_k(t)\right]s_j(t)\prod_{k'\in\partial i}\frac{1+s_{k'}(t)m_{k'}(t)}{2}. \tag{29}$$

Substituting Eqs. (26), (27), (28), and (29) for the probability distributions, Eqs. (18) and (19), we recognize the probability distributions divide into $O(1)$ order terms and $O(\frac{1}{N})$ order terms. To save space, the following symbols are introduced to represent the formula:

$$\Delta = \sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}} \tanh\left[\sum_{k\in\partial i} J_{ik}s_k(t)\right]\prod_{k'\in\partial i}\frac{1+s_{k'}(t)m_{k'}(t)}{2}, \tag{30}$$

$$\Box_j = \sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}} \tanh\left[\sum_{k\in\partial i} J_{ik}s_k(t)\right]s_j(t)\prod_{k'\in\partial i}\frac{1+s_{k'}(t)m_{k'}(t)}{2}. \tag{31}$$

Using these symbols, Eq. (18) becomes

$$\begin{aligned} P\left(s_i\left(t+\frac{1}{N}\right), s_i(t)\right) &= \frac{1}{4}\left\{1+s_i\left(t+\frac{1}{N}\right)m_i(t)+s_i(t)m_i(t)+s_i\left(t+\frac{1}{N}\right)s_i(t)\right\}\\ &\quad+\frac{1}{N}\frac{1}{4}\left\{s_i\left(t+\frac{1}{N}\right)[-m_i(t)+\Delta]+s_i\left(t+\frac{1}{N}\right)s_i(t)[-1+m_i(t)\Delta]\right\}, \end{aligned} \tag{32}$$

and Eq. (19) becomes

$$\begin{aligned} P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right) &= \frac{1}{8}\left\{1+s_i\left(t+\frac{1}{N}\right)m_i(t)+s_i(t)m_i(t)+s_j(t)m_j(t)+s_i\left(t+\frac{1}{N}\right)s_i(t)\right.\\ &\quad\left.+\,s_i\left(t+\frac{1}{N}\right)s_j(t)m_i(t)m_j(t)+s_i(t)s_j(t)m_i(t)m_j(t)+s_i\left(t+\frac{1}{N}\right)s_i(t)s_j(t)m_j(t)\right\}\\ &\quad+\frac{1}{N}\frac{1}{8}\left\{s_i\left(t+\frac{1}{N}\right)[-m_i(t)+\Delta]+s_i\left(t+\frac{1}{N}\right)s_i(t)[-1+m_i(t)\Delta]\right.\\ &\quad\left.+\,s_i\left(t+\frac{1}{N}\right)s_j(t)[-m_i(t)m_j(t)+\Box_j]+s_i\left(t+\frac{1}{N}\right)s_i(t)s_j(t)[-m_j(t)+m_i(t)\Box_j]\right\}. \end{aligned} \tag{33}$$

Equations (17), (32), and (33) are substituted for Eq. (14), and then $O(1)$ order terms and $O(\frac{1}{N})$ order terms are separated. Before substitution, we transform Eq. (14) into the following formula

$$\begin{aligned} \mathcal{T}^{(a)}_{s_j\to s_i} &= \sum_{s_i(t+\frac{1}{N})\in\mathcal{X},}\sum_{s_i(t)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right)\ln\left\{\frac{P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right)}{P[s_j(t)]\;P\left(s_i\left(t+\frac{1}{N}\right), s_i(t)\right)}\right\}\\ &= \sum_{s_i(t+\frac{1}{N})\in\mathcal{X},}\sum_{s_i(t)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} [\delta_{s_i(t+\frac{1}{N}),s_i(t)}+\delta_{s_i(t+\frac{1}{N}),-s_i(t)}]P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right)\ln\left\{\frac{P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right)}{P[s_j(t)]\;P\left(s_i\left(t+\frac{1}{N}\right), s_i(t)\right)}\right\}\\ &= \sum_{s_i(t)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} P(s_i(t), s_i(t), s_j(t))\ln\left\{\frac{P(s_i(t), s_i(t), s_j(t))}{P[s_j(t)]\;P(s_i(t), s_i(t))}\right\}\\ &\quad+\sum_{s_i(t)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} P(-s_i(t), s_i(t), s_j(t))\ln\left\{\frac{P(-s_i(t), s_i(t), s_j(t))}{P[s_j(t)]\;P(-s_i(t), s_i(t))}\right\}, \end{aligned} \tag{34}$$

where the formula $P(s_i(t), s_j(t)) = P[s_i(t)]P[s_j(t)]$ is utilized in the derivation of the right-hand side of the first line. In the case of $s_i(t+\frac{1}{N}) = s_i(t)$, Eqs. (32) and (33) become

$$P(s_i(t), s_i(t)) = \frac{1}{2}\{1+s_i(t)m_i(t)\}+\frac{1}{N}\frac{1}{4}\{s_i(t)[-m_i(t)+\Delta]-1+m_i(t)\Delta\}, \tag{35}$$

$$\begin{aligned} P(s_i(t), s_i(t), s_j(t)) &= \frac{1}{4}\{1+s_i(t)m_i(t)+s_j(t)m_j(t)+s_i(t)s_j(t)m_i(t)m_j(t)\}\\ &\quad+\frac{1}{N}\frac{1}{8}\{s_i(t)[-m_i(t)+\Delta]-1+m_i(t)\Delta+s_i(t)s_j(t)[-m_i(t)m_j(t)+\Box_j]+s_j(t)[-m_j(t)+m_i(t)\Box_j]\}, \end{aligned} \tag{36}$$

respectively. On the other hand, when $s_i(t+\frac{1}{N}) = -s_i(t)$, Eqs. (32) and (33) become

$$P(-s_i(t), s_i(t)) = \frac{1}{N}\frac{1}{4}\{s_i(t)[m_i(t)-\Delta] + 1 - m_i(t)\Delta\}, \tag{37}$$

$$P(-s_i(t), s_i(t), s_j(t)) = \frac{1}{N}\frac{1}{8}\{s_i(t)[m_i(t)-\Delta] + 1 - m_i(t)\Delta + s_i(t)s_j(t)[m_i(t)m_j(t) - \square_j] + s_j(t)[m_j(t) - m_i(t)\square_j]\}, \tag{38}$$

respectively. Note that Eqs. (37) and (38) are composed of only $O(\frac{1}{N})$ order terms. Substituting Eqs. (35), (36), (37), and (38) to the last two lines of Eq. (34), we expand it with respect to $\frac{1}{N}$. If the tedious intermediate derivation process is omitted and the transfer entropy after the order of $O(\frac{1}{N})$ is written out, Eq. (34) becomes

$$\begin{aligned}
\mathcal{T}^{(a)}_{s_j\to s_i} = {} & \frac{1}{N}\sum_{s_i(t)\in\mathcal{X},\, s_j(t)\in\mathcal{X}} \frac{1}{8}\{s_i(t)s_j(t)[-m_j(t)\Delta + \square_j] + s_j m_i(t)[-m_j(t)\Delta + \square_j]\} \\
& + \frac{1}{N}\sum_{s_i(t)\in\mathcal{X},\, s_j(t)\in\mathcal{X}} \frac{1}{8}\{s_i(t)[m_i(t)-\Delta] + 1 - m_i(t)\Delta + s_i(t)s_j(t)[m_i(t)m_j(t) - \square_j] + s_j(t)[m_j(t) - m_i(t)\square_j]\} \\
& \times \ln\left\{\frac{s_i(t)[m_i(t)-\Delta] + 1 - m_i(t)\Delta + s_i(t)s_j(t)[m_i(t)m_j(t) - \square_j] + s_j(t)[m_j(t) - m_i(t)\square_j]}{\{s_i(t)[m_i(t)-\Delta] + 1 - m_i(t)\Delta\}\{1 + s_j(t)m_j(t)\}}\right\} + O\left(\frac{1}{N^2}\right).
\end{aligned} \tag{39}$$

The first term originates from the term of $s_i(t+\frac{1}{N}) = s_i(t)$ in Eq. (34) and the second term from that of $s_i(t+\frac{1}{N}) = -s_i(t)$. Summing over $s_i(t)$ and $s_j(t)$, the first term becomes zero due to symmetry, and only the second term contributes to the transfer entropy. As a result, the transfer entropy is derived as

$$\begin{aligned}
\mathcal{T}^{(a)}_{s_j\to s_i} = {} & \frac{1}{N}\sum_{s_i(t)\in\mathcal{X},\, s_j(t)\in\mathcal{X}} \frac{1}{8}\{s_i(t)[m_i(t)-\Delta] + 1 - m_i(t)\Delta + s_i(t)s_j(t)[m_i(t)m_j(t) - \square_j] + s_j(t)[m_j(t) - m_i(t)\square_j]\} \\
& \times \ln\left\{\frac{s_i(t)[m_i(t)-\Delta] + 1 - m_i(t)\Delta + s_i(t)s_j(t)[m_i(t)m_j(t) - \square_j] + s_j(t)[m_j(t) - m_i(t)\square_j]}{\{s_i(t)[m_i(t)-\Delta] + 1 - m_i(t)\Delta\}\{1 + s_j(t)m_j(t)\}}\right\}.
\end{aligned} \tag{40}$$

By substituting $m_i(t)$ to Eq. (40), we can evaluate it. When the system is in a steady state, the condition $m_i(t+\frac{1}{N}) = m_i(t)$ is satisfied. Substituting $m_i(t+\frac{1}{N}) = m_i(t)$ to Eq. (26), we obtain

$$m_i(t) = \sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}} \tanh\left[\sum_{k\in\partial i} J_{ik}s_k(t)\right] \prod_{k'\in\partial i} \frac{1 + s_{k'}(t)m_{k'}(t)}{2}. \tag{41}$$

Equation (41) determines $m_i(t)$, and can be resolved by an iteration method. Using Eq. (41), i.e., $m_i(t) = \Delta$, we simplify Eq. (40) as follows;

$$\begin{aligned}
\mathcal{T}^{(a)}_{s_j\to s_i} = {} & \frac{1}{N}\sum_{s_i(t)\in\mathcal{X},\, s_j(t)\in\mathcal{X}} \frac{1}{8}\{1 - m_i(t)m_i(t) + s_i(t)s_j(t)[m_i(t)m_j(t) - \square_j] + s_j(t)[m_j(t) - m_i(t)\square_j]\} \\
& \times \ln\left\{\frac{1 - m_i(t)m_i(t) + s_i(t)s_j(t)[m_i(t)m_j(t) - \square_j] + s_j(t)[m_j(t) - m_i(t)\square_j]}{\{1 - m_i(t)m_i(t)\}\{1 + s_j(t)m_j(t)\}}\right\}.
\end{aligned} \tag{42}$$

2. Synchronous update

The derivation of the transfer entropy in the synchronous update is much easier than that in the asynchronous update. In the star approximation, Eq. (10) becomes

$$P(s_i(t+1), s_{\partial i}(t)) = w^{(s)}(s_i(t+1)|s_{\partial i}(t)) \prod_{j\in\partial i} P[s_j(t)]. \tag{43}$$

The difference from the case of the asynchronous update is that Eq. (43) does not depend on $s_i(t)$, i.e., $P(s_i(t+1), s_i(t), s_{\partial i}(t)) = P(s_i(t+1), s_{\partial i}(t))P[s_i(t)]$. The proof of it appears in Appendix C. This property greatly simplifies the problem. Taking account of this property, the transfer entropy becomes

$$\begin{aligned}
\mathcal{T}^{(s)}_{s_j\to s_i} & = \sum_{s_i(t+1)\in\mathcal{X},}\ \sum_{s_i(t)\in\mathcal{X},}\ \sum_{s_j(t)\in\mathcal{X}} P(s_i(t+1), s_i(t), s_j(t)) \ln\left\{\frac{P(s_i(t+1), s_i(t), s_j(t))}{P(s_i(t), s_j(t))}\,\frac{P[s_i(t)]}{P(s_i(t+1), s_i(t))}\right\} \\
& = \sum_{s_i(t+1)\in\mathcal{X},}\ \sum_{s_i(t)\in\mathcal{X},}\ \sum_{s_j(t)\in\mathcal{X}} P(s_i(t+1), s_j(t))P[s_i(t)] \ln\left\{\frac{P(s_i(t+1), s_j(t))P[s_i(t)]}{P[s_i(t)]P[s_j(t)]}\,\frac{P[s_i(t)]}{P[s_i(t+1)]P[s_i(t)]}\right\}
\end{aligned}$$

$$= -\sum_{s_i(t+1)\in\mathcal{X}} P[s_i(t+1)]\ln P[s_i(t+1)] - \sum_{s_j(t)\in\mathcal{X}} P[s_j(t)]\ln P[s_j(t)]$$
$$+ \sum_{s_i(t+1)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} P(s_i(t+1), s_j(t))\ln P(s_i(t+1), s_j(t)). \tag{44}$$

Note that, compared to the mutual information, $P(s_i(t), s_j(t))$ is replaced with $P(s_i(t+1), s_j(t))$. To evaluate the transfer entropy, the probability distributions, $P[s_i(t+1)]$, $P[s_j(t)]$, and $P(s_i(t+1), s_j(t))$ are required. In the same way as the asynchronous update, the probability distributions are expressed as

$$P[s_i(t)] = \tfrac{1}{2}[1 + s_i(t)m_i(t)], \tag{45}$$

$$P(s_i(t+1), s_j(t)) = \tfrac{1}{4}[1 + s_i(t+1)m_i(t+1) + s_j(t)m_j(t) + s_i(t+1)s_j(t)c'_{ij}(t+1,t)], \tag{46}$$

where $m_i(t)$ is defined as (20), and $c'_{ij}(t+1,t)$ is defined as

$$c'_{ij}(t+1,t) = \sum_{s_i(t+1)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} s_i(t+1)s_j(t)P(s_i(t+1), s_j(t)). \tag{47}$$

The renewal rule of $m_i(t)$ is given as

$$m_i(t+1) = \sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}} \tanh\left[\sum_{k\in\partial i} J_{ik}s_k(t)\right] \prod_{k'\in\partial i} \frac{1 + s_{k'}(t)m_{k'}(t)}{2}. \tag{48}$$

The formula $c'_{ij}(t+1,t)$ is derived as

$$c'_{ij}(t+1,t) = \sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}} \tanh\left[\sum_{k\in\partial i} J_{ik}s_k(t)\right] s_j(t) \prod_{k'\in\partial i} \frac{1 + s_{k'}(t)m_{k'}(t)}{2}. \tag{49}$$

Using Eqs. (45) and (46), the transfer entropy is given as

$$\mathcal{T}^{(s)}_{s_j\to s_i} = -\sum_{s_i(t+1)\in\mathcal{X}} \frac{1 + s_i(t+1)\Delta}{2}\ln\left[\frac{1 + s_i(t+1)\Delta}{2}\right] - \sum_{s_j(t)\in\mathcal{X}} \frac{1 + s_j(t)m_j(t)}{2}\ln\left[\frac{1 + s_j(t)m_j(t)}{2}\right]$$
$$+ \sum_{s_i(t+1)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} \frac{1 + s_i(t+1)\Delta + s_j(t)m_j(t) + s_i(t+1)s_j(t)\Box_j}{4}\ln\left[\frac{1 + s_i(t+1)\Delta + s_j(t)m_j(t) + s_i(t+1)s_j(t)\Box_j}{4}\right], \tag{50}$$

where we use the symbols defined in Eqs. (30) and (31). In the case of a system in steady state, the relation $m_i(t+1) = m_i(t)$ is satisfied, and then the transfer entropy is described by a simpler formula.

III. NUMERICAL SIMULATION RESULTS

To verify the effectiveness of our method, we compare our results with the transfer entropy obtained by the Monte Carlo simulation. We adopt Ising models on the square and cubic lattices as examples. We assume that the system is uniform so that $J_{ij} = J$. Because of the uniformity of space, $m_i(t)$ does not depend on i, i.e., $m_i(t) = m(t)$ for all i. In the case of the asynchronous update, the probability distribution converges to the Boltzmann distribution after iterating the asynchronous update infinitely. In the steady state, $m(t)$ does not depend on time, and we set $m(t) = m$. The transfer entropy becomes

$$\mathcal{T}^{(a)} = \frac{1}{N}\sum_{\sigma\in\mathcal{X},\,\sigma'\in\mathcal{X}} \frac{1}{8}\{1 - m^2 + \sigma\sigma'(m^2 - \Box) + \sigma' m(1 - \Box)\}$$
$$\times \ln\left\{\frac{1 - m^2 + \sigma\sigma'(m^2 - \Box) + \sigma' m(1 - \Box)}{(1 - m^2)(1 + \sigma' m)}\right\}, \tag{51}$$

where m is evaluated from

$$m = \sum_{\sigma_1\in\mathcal{X},}\cdots\sum_{\sigma_D\in\mathcal{X}} \tanh\left[J\sum_{k=1}^{D}\sigma_k\right]\prod_{k'=1}^{D}\frac{1 + \sigma_{k'}m}{2}. \tag{52}$$

Equation (52) is equivalent to m in the hard-spin mean-field theory [32–39]. The symbol $\Box$ is defined as

$$\Box = \sum_{\sigma_1\in\mathcal{X},}\cdots\sum_{\sigma_D\in\mathcal{X}} \tanh\left[J\sum_{k=1}^{D}\sigma_k\right]\sigma_1\prod_{k'=1}^{D}\frac{1 + \sigma_{k'}m}{2}, \tag{53}$$

where m obtained from Eq. (52) is substituted in. The quantity D is the number of neighboring vertices, and the value of D is 4 or 6 for the square lattice or cubic lattice, respectively. In the case of the synchronous update, the transfer entropy becomes

$$\mathcal{T}^{(s)} = -\sum_{\sigma\in\mathcal{X}}(1+\sigma m)\ln\left(\frac{1+\sigma m}{2}\right) + \sum_{\sigma\in\mathcal{X},}\sum_{\sigma'\in\mathcal{X}}\frac{1+(\sigma+\sigma')m+\sigma\sigma'\square}{4} \times \ln\left[\frac{1+(\sigma+\sigma')m+\sigma\sigma'\square}{4}\right], \tag{54}$$

where m is evaluated from (52), and $\square$ is defined as Eq. (53). Note that m is common to both the asynchronous and the synchronous updates in the star approximation.

Using dataset sampled by the Monte Carlo method, the transfer entropy is evaluated. In the case of asynchronous update and Ising model on a square lattice, the formula giving the transfer entropy is obtained in Ref. [19]. The formula uses both exact solution results, i.e., the exact energy and magnetization, and the Monte Carlo sampling. In the case of Ising model on a cubic lattice, the transfer entropy is estimated by dataset sampled by the Monte Carlo method. The equation for calculating transfer entropy on a cubic lattice by Monte Carlo sampling is present in Appendix D. In the case of synchronous update, there is no formula as in the asynchronous case for obtaining the transfer entropy, hence the transfer entropy is evaluated only by sampling. In that case, the frequency distribution substitutes for the probability distribution in the calculation of the transfer entropy. The Monte Carlo simulation is performed on a square lattice composed of 512 × 512 vertices and a cubic lattice composed of 64 × 64 × 64 vertices. Periodic boundary conditions are imposed on both lattices. Samples taken until the system relaxes to a steady state are discarded. The number of datasets is 10^4, and even if the number of datasets increases further, there is no difference in the results. Additionally, even if the size of the system is further increased, the result does not change.

Figures 1(a) and 1(b) show m as a function of J and compare m (filled circles) obtained by the Monte Carlo simulation with m (open circles) obtained by Eq. (52). The black curve in Fig. 1(a) shows the result of the exact solution and is in complete agreement with the filled circles. Figures 1(c) and 1(d) show the transfer entropy as a function of J when updating asynchronously. Here, the transfer entropy is multiplied by N. Figures 1(e) and 1(f) show the transfer entropy as a function of J when updating synchronously. As can be seen from Eq. (51), the transfer entropy of the asynchronous update, $\mathcal{T}^{(a)}$, is a quantity of $O(\frac{1}{N})$. As shown in Ref. [19] and Eq. (D1) in Appendix D, the formula of the transfer entropy required for the calculation by the Monte Carlo method is also in the order of $O(\frac{1}{N})$. However, as indicated by Eq. (54), the transfer entropy of the synchronous update, $\mathcal{T}^{(s)}$, is a quantity of $O(1)$. The vertical axis of Figs. 1(c) and 1(d) represent N times the amount of the transfer entropy for the asynchronous update. In Figs. 1(c)–1(f), the filled circles show the result by the Monte Carlo method, and the open circles show the result obtained by our method. The left column shows the result of the Ising model on the square lattice, and the right column shows the result of the Ising model on the cubic lattice. The vertical lines represent the critical points ranging from $m = 0$ to $m > 0$. The vertical solid lines show the critical points obtained by other methods, and the vertical broken lines show the critical points obtained by Eq. (52). In the case of the square lattice, the critical value of J is given by the exact solution, and in the case of the cubic lattice, the critical value of J is estimated by high-temperature expansions [40]. The critical values of J are summarized in Table I. As shown in Figs. 1(a) and 1(b), the behavior of m is the same except for the transition point in

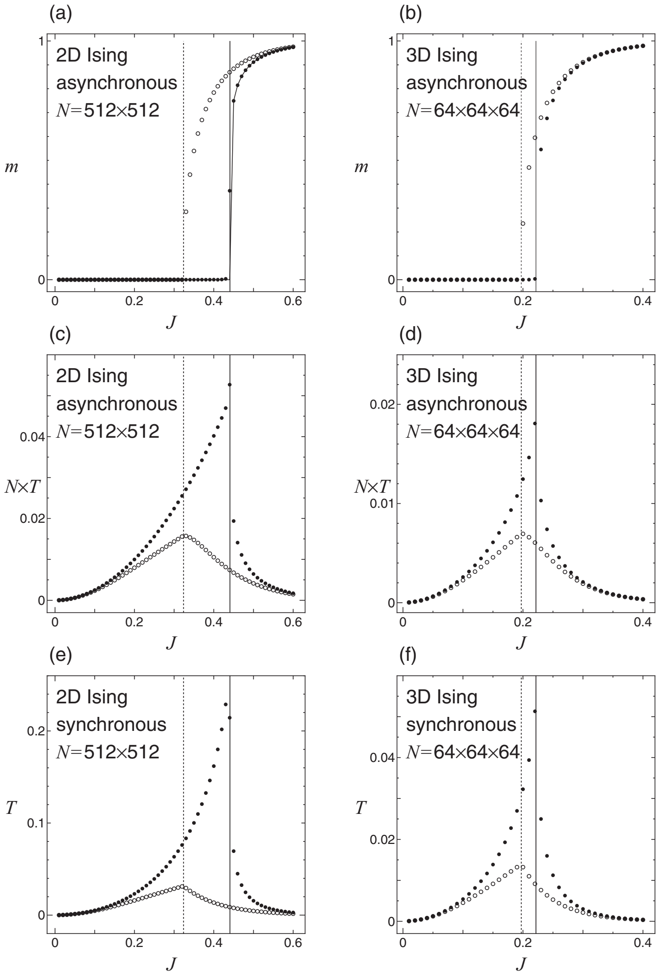

FIG. 1. Comparison of the results obtained using the proposed method (open circles) and Monte Carlo simulation results (filled circles). Figures in the first row represent the results obtained by expressing m as a function of J. Figures in the second row represent the N times transfer entropy expressed as a function of J in the case of asynchronous update. Figures in the third row indicate the transfer entropy by synchronous update, expressed as a function of J. Figures in the left column represent results of Ising model on a square lattice. Figures on the right column represent the results of Ising model on a cubic lattice. Refer to text for explanation of vertical lines.

TABLE I. Critical values of J.

	Exact and expansion	Eq. (52)
Square lattice	0.440686···	0.323643
Cubic lattice	0.221654	0.197109

each case. The difference in the critical value J between other methods and our approximation method is smaller in the cubic lattice than in the square lattice. The behavior of m is identical in both synchronous and asynchronous updates. The transfer entropy computed by our method exhibits a peak at the critical J evaluated by Eq. (52). In all calculated cases, the transfer entropy derived by our method for small and large J is in good agreement with the one derived by the Monte Carlo method. However, in the vicinity of the transition point, the difference in the transfer entropy calculated by these two methods is large. In the case of the cubic lattice, the range in which the two results match is greater than in the case of the square lattice. The behavior of the transfer entropy is qualitatively the same for synchronous and asynchronous updates.

IV. SUMMARY AND DISCUSSION

In this study, transfer entropy was investigated for the asynchronous update and also for updates that do not reach equilibrium, such as the synchronous update. The cluster variational method was adopted as a method to approximate the dynamic Ising model. We have developed a method that approximates the transfer entropy for both the asynchronous and synchronous updates. Only when the update is asynchronous and the coupling parameters between vertices are symmetric is the detailed balance condition satisfied, and the equilibrium state is reached by repeating the update infinitely. Our method is applicable not only to the equilibrium state but also to the nonequilibrium state that does not satisfy the detailed balance condition. Our method is in very good agreement with the Monte Carlo simulation results in the J region excluding the transition point. Even near the transition point, the simulation results are well reproduced qualitatively. The proposed approximate method can determine transfer entropy by solving transcendental equations without the need for sampling, as is case with Monte Carlo simulation. Moreover, even after deviation from the transition point, the transfer entropy can be determined by our method with considerable accuracy. The proposed method is expected to find suitable applications in models wherein the transition probabilities are known.

Equations (42) and (50) show that the transfer entropy for asynchronous updates is in the order of $O(\frac{1}{N})$ and the transfer entropy for synchronous updates is in the order of $O(1)$. Intuitively, the differences in the transfer entropy between the two types of updates are as follows: In the asynchronous update, we select one spin with a probability of $1/N$ per unit time, and then the spin is flipped over according to the transition probability. On the contrary, in the synchronous update, it is simultaneously determined whether or not each spin is flipped over according to the transition probability. This difference manifests in the form of a difference in the order of the transfer entropy.

In the case of the asynchronous update, by combining Eqs. (2) and (3) we can obtain

$$\frac{1}{\gamma}\frac{d}{dt}m(t) = -m(t) + \sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}} \tanh\left[\sum_{j\in\partial i} J_{ij}s_j(t)\right] P[s_{\partial i}(t)], \tag{55}$$

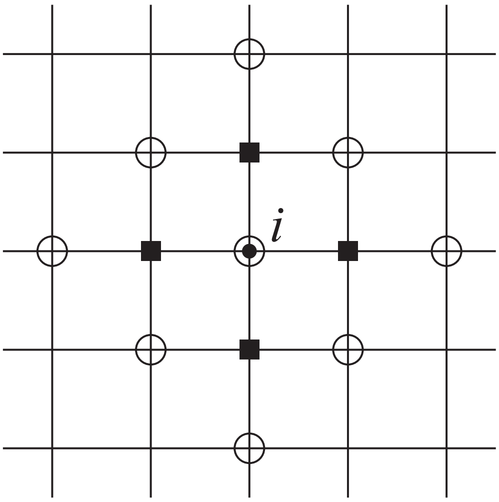

FIG. 2. Schematic diagram of a square lattice. The symbol ● on the center represents ith vertex. The symbols ■ represent vertices adjacent to the ith vertex and correspond to ∂i. The symbols ○ represent vertices adjacent to ■ and correspond to $\partial(\partial i)$.

which determines the time variation of $m(t)$. We have demonstrated the relation between the above differential equation and the Glauber dynamics. Taking into account terms up to the order $O(\frac{1}{N})$, the differential equation and Glauber dynamics coincide with each other. The transfer entropy for the asynchronous update is evaluated up to the order $O(\frac{1}{N})$, with no contradiction in terms of considering the order $O(\frac{1}{N})$, while for the synchronous update, the transfer entropy is evaluated up to the order of $O(1)$. As the transfer entropy in the asynchronous case is a quantity on the order of $\frac{1}{N}$, it is found that it is very difficult to directly obtain the transfer entropy from the empirical distribution.

We discuss the physical consideration of the star approximation and the method of further improving the approximation accuracy. As shown in Appendix E, $m_i(t+1)$ is determined by the sum of the past states of $s_{\partial i}(t)$, and the states of $s_{\partial i}(t)$ are also related to the past states of $s_{\partial(\partial i)}(t-1)$ on the vertices adjacent to a set of ∂i. The more we incorporate the effects of the past, the more distant the vertices will be. The star approximation only considers the most recent past states, i.e., only the states of vertices adjacent to the vertex of interest. Because the star approximation does not incorporate long-range correlation, the accuracy of the approximation became poor near the transition point. It can be seen that to improve the accuracy, it is better to incorporate information from the past state, i.e., to incorporate the influence from the farther vertex state. A similar argument holds for the asynchronous case.

On comparing the two-dimensional results against the three-dimensional results, it was found that using our method, the accuracy of the three-dimensional results was relatively higher. Moreover, the accuracy is expected to improve further as the dimension increases. To illustrate this, let us consider the case where the dimensions are sufficiently large. For the sake of simplicity, let us assume an infinite-range model where all possible pairs of vertices have been connected.

Equation (52) becomes

$$
\begin{aligned}
m &= \sum_{\sigma_1\in\mathcal{X},}\cdots\sum_{\sigma_N\in\mathcal{X}}\tanh\left(J\sum_{k=1}^{N}\sigma_k\right)\prod_{k'=1}^{N}\frac{1+\sigma_{k'}m}{2}\\
&= \sum_{\sigma_1\in\mathcal{X},}\cdots\sum_{\sigma_N\in\mathcal{X}}\tanh(JNm)\prod_{k'=1}^{N}\frac{1+\sigma_{k'}m}{2}\\
&= \tanh(JNm)\prod_{k=1}^{N}\sum_{\sigma_k\in\mathcal{X}}\frac{1+\sigma_k m}{2} = \tanh(JNm).
\end{aligned}
\tag{56}
$$

By assuming that N is sufficiently large, the relation $m = \frac{1}{N}\sum\sigma_k$ can be used to transform the first line to the second line. By replacing JN with J_0, Eq. (56) reproduces the mean-field equation [41]. Given that mean-field theory can determine the exact solution for the infinite-range model; hence, it can be seen that for the infinite dimension, Eq. (52) leads to the exact solution. Considering that Eq. (52) gives an exact solution of m in the infinite dimension, it is expected that using our approximate method, exact values of transfer entropy can be determined in the infinite dimension.

We have derived the formula for the transfer entropy by using dataset under the condition that coupling parameters J_{ij} are given. Recently, a method that infers the coupling parameter J_{ij} via the transfer entropy has been developed [42]. We expect that our method can be used effectively for parameter estimation [43].

APPENDIX A: PROOF OF THE EQUIVALENCE BETWEEN EQ. (5) AND GLAUBER DYNAMICS

Because γ is arbitrary, set $\gamma\delta t = \frac{1}{N}$. We transform Eq. (3) into

$$
\begin{aligned}
w^{(a)}\left(s_i\left(t+\frac{1}{N}\right)|s_i(t), s_{\partial i}(t)\right) &= \left(1-\frac{1}{N}\right)\delta_{s_i(t+\frac{1}{N}),s_i(t)} + \left[\delta_{s_i(t+\frac{1}{N}),s_i(t)} + \delta_{s_i(t+\frac{1}{N}),-s_i(t)}\right]\frac{1}{N}\frac{\exp\left[s_i\left(t+\frac{1}{N}t\right)\sum_{j\in\partial i}J_{ij}s_j(t)\right]}{2\cosh\left(\sum_{j\in\partial i}J_{ij}s_j(t)\right)}\\
&= \delta_{s_i(t+\frac{1}{N}),s_i(t)}\left\{1-\frac{1}{N}\frac{1}{1+\exp\left[2s_i(t)\sum_{j\in\partial i}J_{ij}s_j(t)\right]}\right\}\\
&\quad + \delta_{s_i(t+\frac{1}{N}),-s_i(t)}\left\{\frac{1}{N}\frac{1}{1+\exp\left[2s_i(t)\sum_{j\in\partial i}J_{ij}s_j(t)\right]}\right\}\\
&= \delta_{s_i(t+\frac{1}{N}),s_i(t)}\left\{1-\frac{1}{N}P_i[\boldsymbol{s}(t)]\right\} + \delta_{s_i(t+\frac{1}{N}),-s_i(t)}\frac{1}{N}P_i[\boldsymbol{s}(t)],
\end{aligned}
\tag{A1}
$$

where $P_i[\boldsymbol{s}(t)]$ is defined as

$$
P_i[\boldsymbol{s}(t)] = \frac{1}{1+\exp\left[2s_i(t)\sum_{j\in\partial i}J_{ij}s_j(t)\right]}.
\tag{A2}
$$

We consider the transition probability

$$
\begin{aligned}
W\left(\boldsymbol{s}\left(t+\frac{1}{N}\right)|\boldsymbol{s}(t)\right) &= \prod_{i=1}^{N} w^{(a)}\left(s_i\left(t+\frac{1}{N}\right)|s_i(t), s_{\partial i}(t)\right)\\
&= \prod_{i=1}^{N}\left(\delta_{s_i(t+\frac{1}{N}),s_i(t)}\underbrace{\left\{1-\frac{1}{N}P_i[\boldsymbol{s}(t)]\right\}}_{O(1)} + \delta_{s_i(t+\frac{1}{N}),-s_i(t)}\underbrace{\frac{1}{N}P_i[\boldsymbol{s}(t)]}_{O(\frac{1}{N})}\right).
\end{aligned}
\tag{A3}
$$

Here, the terms in the parentheses are divided into a term of order $O(1)$ and a term of order $O(\frac{1}{N})$. We expand Eq. (A3) up to $O(\frac{1}{N})$, and then

$$
\begin{aligned}
W\left(\boldsymbol{s}\left(t+\frac{1}{N}\right)|\boldsymbol{s}(t)\right) &= \prod_{i=1}^{N}\delta_{s_i(t+\frac{1}{N}),s_i(t)}\left\{1-\frac{1}{N}P_i[\boldsymbol{s}(t)]\right\} + \sum_{i=1}^{N}\delta_{s_i(t+\frac{1}{N}),-s_i(t)}\frac{1}{N}P_i[\boldsymbol{s}(t)]\prod_{j\neq i}^{N}\delta_{s_j(t+\frac{1}{N}),s_j(t)}\left[1-\frac{1}{N}P_j[\boldsymbol{s}(t)]\right] + \cdots\\
&= \prod_{i=1}^{N}\delta_{s_i(t+\frac{1}{N}),s_i(t)}\left\{1-\frac{1}{N}\sum_{k=1}^{N}P_k[\boldsymbol{s}(t)]\right\} + \sum_{i=1}^{N}\delta_{s_i(t+\frac{1}{N}),-s_i(t)}\prod_{j\neq i}^{N}\delta_{s_j(t+\frac{1}{N}),s_j(t)}\frac{1}{N}P_i[\boldsymbol{s}(t)] + O\left(\frac{1}{N^2}\right).
\end{aligned}
\tag{A4}
$$

Equation (A4) agrees with the transition probability of the Glauber dynamics [31]. The first term corresponds to the case where the state does not change, and the second term corresponds to the transition where one spin is turned over.

APPENDIX B: DERIVATION OF EQS. (18) AND (19)

Given that $s_i(t+\frac{1}{N})$, $s_i(t)$, and $s_j(t)$ take the value of $+1$ or -1, we can expand Eq. (19) as

$$P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right) = \frac{1}{8}\left[1 + s_i\left(t+\frac{1}{N}\right)a + s_i(t)b + s_j(t)c + s_i\left(t+\frac{1}{N}\right)s_i(t)d + s_i\left(t+\frac{1}{N}\right)s_j(t)e \right.$$
$$\left. + \, s_i(t)s_j(t)f + s_i\left(t+\frac{1}{N}\right)s_i(t)s_j(t)g\right], \tag{B1}$$

where a, b, c, d, e, f, and g are unknown constants. The coefficient $1/8$ originates from the normalization condition,

$$\sum_{s_i(t+\frac{1}{N})\in\mathcal{X},}\sum_{s_i(t)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right) = 1. \tag{B2}$$

By multiplying both sides of Eq. (B1) by $s_i(t+\frac{1}{N})$ and adding $s_i(t+\frac{1}{N})$, $s_i(t)$, and $s_j(t)$, we obtain

$$\sum_{s_i(t+\frac{1}{N})\in\mathcal{X},}\sum_{s_i(t)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} s_i\left(t+\frac{1}{N}\right)P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right) = a. \tag{B3}$$

Using Eqs. (20), (24), and (25), we can demonstrate that $a = m_i(t+\frac{1}{N})$. By multiplying both sides of Eq. (B1) by $s_i(t+\frac{1}{N})s_i(t)s_j(t)$ and adding $s_i(t+\frac{1}{N})$, $s_i(t)$, and $s_j(t)$, we obtain

$$\sum_{s_i(t+\frac{1}{N})\in\mathcal{X},}\sum_{s_i(t)\in\mathcal{X},}\sum_{s_j(t)\in\mathcal{X}} s_i\left(t+\frac{1}{N}\right)s_i(t)s_j(t)P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right) = g. \tag{B4}$$

Using Eq. (23), we can demonstrate that $g = t_{ij}(t+\frac{1}{N}, t)$. The other unknown constants, except f, can be determined in a similar manner. To determine f, we evaluate

$$P(s_i(t), s_j(t)) = \sum_{s_i(t+\frac{1}{N})\in\mathcal{X}} P\left(s_i\left(t+\frac{1}{N}\right), s_i(t), s_j(t)\right) = \frac{1}{4}[1 + s_i(t)b + s_j(t)c + s_i(t)s_j(t)f], \tag{B5}$$

where b and c are $m_i(t)$ and $m_j(t)$, respectively. In the star approximation, the probability distribution, Eq. (B5), is factorized as $P(s_i(t), s_j(t)) = P[s_i(t)]P[s_j(t)]$. Using Eq. (17), we can demonstrate that the assertion $f = m_i(t)m_j(t)$ holds. The equation (18) is derived by substituting Eq. (19) in the right-hand side of Eq. (25).

APPENDIX C: PROOF OF $P(s_i(t+1), s_i(t), s_{\partial i}(t)) = P(s_i(t+1), s_{\partial i}(t))P[s_i(t)]$ IN THE CASE OF SYNCHRONOUS UPDATE

In Eq. (9), we marginalize $\boldsymbol{s}(t+1)$ and $\boldsymbol{s}(t)$ except $s_i(t+1)$, $s_i(t)$, and $s_{\partial i}(t)$, the following formula

$$P(s_i(t+1), s_i(t), s_{\partial i}(t)) = w^{(s)}(s_i(t+1)|s_{\partial i}(t))P(s_i(t), s_{\partial i}(t)), \tag{C1}$$

is derived. In the star approximation, Eq. (C1) becomes

$$P(s_i(t+1), s_i(t), s_{\partial i}(t)) = w^{(s)}(s_i(t+1)|s_{\partial i}(t))P[s_i(t)]\prod_{j\in\partial i} P[s_i(t)]. \tag{C2}$$

Because $P(s_i(t+1), s_{\partial i}(t)|s_i(t)) = P(s_i(t+1), s_{\partial i}(t))$, the proposition is proved.

APPENDIX D: TRANSFER ENTROPY ON A CUBIC LATTICE BY MONTE CARLO METHOD IN THE CASE OF ASYNCHRONOUS UPDATE

We present determination of transfer entropy on a cubic lattice using Monte Carlo sampling in the case of the asynchronous update. It is derived by referring to the derivation in Ref. [19]. The transfer entropy is given by

$$\mathcal{T}^{(a)} = \frac{1}{N}\left\{-\sum_{\sigma\in\mathcal{X}} q \ln\left[\frac{q}{\frac{1}{2}(1+\sigma m)}\right] + \sum_{\sigma\in\mathcal{X},}\sum_{\sigma'\in\mathcal{X}} q_{\sigma'} \ln\left[\frac{q_{\sigma'}}{\frac{1}{4}(1+\sigma m+\sigma' m-\frac{1}{3}\sigma\sigma'\mathcal{U})}\right]\right\}, \tag{D1}$$

where m and $\mathcal{U}$ are the magnetization and internal energy, respectively. It should be noted that coefficient $1/3$ is different from the transfer entropy on a square lattice. In the case of Ising model on a square lattice, the exact solution can be used for m and $\mathcal{U}$. However, in the case of Ising model on a cubic lattice, m and $\mathcal{U}$ are evaluated by the Monte Carlo method. The quantities q and q_σ are defined by the following equations.

$$q = \tfrac{1}{2}\langle P_i(\boldsymbol{s})\rangle, \quad q_\sigma = \tfrac{1}{4}\{\langle P_i(\boldsymbol{s})\rangle + \sigma\langle s_j P_i(\boldsymbol{s})\rangle\}, \tag{D2}$$

where $P_i(\boldsymbol{s})$ is given by Eq. (A2), $\langle\cdots\rangle$ denotes an ensemble average on equilibrium, and $j \in \partial i$. The quantities $\langle P_i(\boldsymbol{s})\rangle$ and $\langle s_j P_i(\boldsymbol{s})\rangle$ are evaluated using the Monte Carlo method for both square and cubic lattices.

APPENDIX E: PHYSICAL CONSIDERATION OF THE STAR APPROXIMATION

We will discuss the star approximation from the physical point of view and the way in which the accuracy of the approximation can be increased. For simplicity, we assume that the dynamics are provided by the synchronous update. Let us start with Eq. (10). The magnetization of the ith vertex is obtained from Eq. (10) as

$$\begin{aligned} m_i(t+1) &= \sum_{s_i(t+1)\in\mathcal{X},}\ \sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}} s_i(t+1)P(s_i(t+1), s_{\partial i}(t)) \\ &= \sum_{s_i(t+1)\in\mathcal{X},}\ \sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}} s_i(t+1)w^{(s)}(s_i(t+1)|s_{\partial i}(t)]P[s_{\partial i}(t)] \\ &= \sum_{s_{\partial i}(t)\in\mathcal{X}^{|\partial i|}} \tanh\left[\sum_{j\in\partial i} J_{ij}s_j(t)\right]P[s_{\partial i}(t)]. \end{aligned} \tag{E1}$$

In the star approximation, the joint probability distribution $P[s_{\partial i}(t)]$ is factorized by a single-vertex probability distribution. Without the approximation, $P[s_{\partial i}(t)]$ is obtained from Eq. (9) as

$$P[s_{\partial i}(t)] = \sum_{s_{\partial(\partial i)}(t-1)\in\mathcal{X}^{|\partial(\partial i)|}}\ \prod_{j\in\partial i} w^{(s)}(s_j(t)|s_{\partial j}(t-1))P[s_{\partial(\partial i)}(t-1)], \tag{E2}$$

where $\partial(\partial i)$ represents a set of vertices that are further adjacent to the vertices adjacent to the ith vertex. For a more concrete explanation, let us represent them in the case of a square lattice in Fig. 2. The symbols ■ and ◯ represent ∂i and $\partial(\partial i)$, respectively. Substituting Eq. (E2) into Eq. (E1), we obtain the following formula,

$$\begin{aligned} m_i(t+1) &= \sum_{s_{\partial i}(t),}\ \sum_{s_{\partial(\partial i)}(t-1)} \tanh\left[\sum_{j\in\partial i} J_{ij}s_j(t)\right]\prod_{j\in\partial i} w^{(s)}(s_j(t)|s_{\partial j}(t-1))P[s_{\partial(\partial i)}(t-1)] \\ &= \sum_{s_{\partial i}(t),}\ \sum_{s_{\partial(\partial i)}(t-1)} \tanh\left[\sum_{j\in\partial i} J_{ij}s_j(t)\right]\prod_{j\in\partial i} \frac{\exp\left[s_j(t)\sum_{k\in\partial j} J_{jk}s_k(t-1)\right]}{2\cosh\left[\sum_{k\in\partial j} J_{jk}s_k(t-1)\right]}P[s_{\partial(\partial i)}(t-1)]. \end{aligned} \tag{E3}$$

Equation (E1) shows that the sum is performed for s on the vertices • and ■ in Fig. 2. Equation (E3) shows that the sum is performed for s on the vertices •, ■, and ◯. To find out how the accuracy of the approximation has changed, we approximate $P[s_{\partial(\partial i)}(t-1)]$ as a product of single-vertex probability distributions. Equation (E3) then becomes

$$m_i(t+1) = \sum_{s_{\partial i}(t),}\ \sum_{s_{\partial(\partial i)}(t-1)} \tanh\left[\sum_{j\in\partial i} J_{ij}s_j(t)\right]\prod_{j\in\partial i} \frac{\exp\left[s_j(t)\sum_{k\in\partial j} J_{jk}s_k(t-1)\right]}{2\cosh\left[\sum_{k\in\partial j} J_{jk}s_k(t-1)\right]}\prod_{k\in\partial(\partial i)} \frac{1+s_k(t-1)m_k(t-1)}{2}. \tag{E4}$$

For comparison, we assume the system is in the steady state and the coupling parameter J_{ij} is uniform, i.e., $J_{ij} = J$. We can then set $m_i(t+1) = m$ and $m_k(t-1) = m$. The critical value of J, where the solution m of the transcendental equation changes from a value of zero to a positive value, is determined as $J = 0.334736$. This value is an improvement from the case of the star approximation and is closer to the exact value in Table I than the value calculated by the star approximation. It is expected that the accuracy will improve if calculations are performed by incorporating older s in the past.

[1] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer Science+ Business Media, New York, 2006).

[2] K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT Press, Cambridge, MA, 2012).

[3] G. Torlai and R. G. Melko, Phys. Rev. B **94**, 165134 (2016).

[4] T. Ohtsuki, J. Phys. Soc. Jpn. **85**, 123706 (2016).

[5] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).

[6] E. P. Van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nat. Phys. **13**, 435 (2017).

[7] S. J. Wetzel and M. Scherzer, Phys. Rev. B **96**, 184410 (2017).

[8] P. Ponte and R. G. Melko, Phys. Rev. B **96**, 205146 (2017).

[9] A. Morningstar and R. G. Melko, J. Mach. Learn. Res. **18**, 5975 (2017).

[10] D. Kim and D.-H. Kim, Phys. Rev. E **98**, 022138 (2018).

[11] Y.-H. Liu and E. P. L. van Nieuwenburg, Phys. Rev. Lett. **120**, 176401 (2018).

[12] L. Wang, Phys. Rev. B **94**, 195105 (2016).

[13] W. Hu, R. R. P. Singh, and R. T. Scalettar, Phys. Rev. E **95**, 062122 (2017).

[14] S. J. Wetzel, Phys. Rev. E **96**, 022140 (2017).

[15] H. Kiwata, Phys. Rev. E **99**, 063304 (2019).

[16] T. Schreiber, Phys. Rev. Lett. **85**, 461 (2000).

[17] T. Bossomaier, L. Barnett, M. Harré, and J. T. Lizier, *An Introduction to Transfer Entropy* (Springer, Berlin, 2016).

[18] H. Matsuda, K. Kudo, R. Nakamura, O. Yamakawa, and T. Murata, Int. J. Theor. Phys. **35**, 839 (1996).

[19] L. Barnett, J. T. Lizier, M. Harré, A. K. Seth, and T. Bossomaier, Phys. Rev. Lett. **111**, 177203 (2013).

[20] Z. Deng, J. Wu, and W. Guo, Phys. Rev. E **90**, 063308 (2014).

[21] A. Pelizzola, Eur. Phys. J. B **86**, 120 (2013).

[22] A. Pelizzola and M. Pretti, J. Stat. Mech.: Theor. Exp. (2017) 073406.

[23] E. D. Vázquez, G. Del Ferraro, and F. Ricci-Tersenghi, J. Stat. Mech.: Theor. Exp. (2017) 033303.

[24] H. J. Kappen and J. J. Spanjers, Phys. Rev. E **61**, 5658 (2000).

[25] H.-L. Zeng, E. Aurell, M. Alava, and H. Mahmoudi, Phys. Rev. E **83**, 041135 (2011).

[26] E. Aurell and H. Mahmoudi, J. Stat. Mech.: Theor. Exp. (2011) P04014.

[27] E. Aurell and H. Mahmoudi, Commun. Theor. Phys. **56**, 157 (2011).

[28] Y. Roudi and J. Hertz, Phys. Rev. Lett. **106**, 048702 (2011).

[29] E. Aurell and H. Mahmoudi, Phys. Rev. E **85**, 031119 (2012).

[30] H.-L. Zeng, M. Alava, E. Aurell, J. Hertz, and Y. Roudi, Phys. Rev. Lett. **110**, 210601 (2013).

[31] R. J. Glauber, J. Math. Phys. **4**, 294 (1963).

[32] J. R. Banavar, M. Cieplak, and A. Maritan, Phys. Rev. Lett. **67**, 1807 (1991).

[33] R. R. Netz and A. N. Berker, Phys. Rev. Lett. **67**, 1808 (1991).

[34] A. Kabakçiog¯lu, A. N. Berker, and M. C. Yalabik, Phys. Rev. E **49**, 2680 (1994).

[35] E. F. Henriques, V. B. Henriques, and S. R. Salinas, Phys. Rev. B **51**, 8621 (1995).

[36] A. Kabakçıoğlu, Phys. Rev. E **61**, 3366 (2000).

[37] B. Yücesoy and A. N. Berker, Phys. Rev. B **76**, 014417 (2007).

[38] T. Çağlar and A. N. Berker, Phys. Rev. E **84**, 051129 (2011).

[39] T. Çağlar and A. N. Berker, Phys. Rev. E **92**, 062131 (2015).

[40] P. Butera and M. Comi, Phys. Rev. B **62**, 14837 (2000).

[41] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction* (Clarendon Press, London, 2001), p. 111.

[42] F. Goetze and P.-Y. Lai, Phys. Rev. E **100**, 012121 (2019).

[43] H. C. Nguyen, R. Zecchina, and J. Berg, Adv. Phys. **66**, 197 (2017).