



This is the accepted manuscript made available via CHORUS. The article has been published as:

Finding community structure using the ordered random graph model

Masaki Ochi and Tatsuro Kawamoto

Phys. Rev. E **108**, 014303 — Published 7 July 2023

DOI: [10.1103/PhysRevE.108.014303](https://doi.org/10.1103/PhysRevE.108.014303)

Finding community structure using the ordered random graph model

Masaki Ochi*

*Department of Physics, The University of Tokyo,
5-1-5 Kashiwanoha, Kashiwa, Chiba, 277-8574, Japan*

Tatsuro Kawamoto†

*Artificial Intelligence Research Center,
National Institute of Advanced Industrial Science and Technology,
2-3-26 Aomi, Koto-ku, Tokyo, 135-0064, Japan*

(Dated: June 5, 2023)

Visualization of the adjacency matrix enables us to capture macroscopic features of a network when the matrix elements are aligned properly. Community structure, a network consisting of several densely connected components, is a particularly important feature, and the structure can be identified through the adjacency matrix when it is close to a block-diagonal form. However, classical ordering algorithms for matrices fail to align matrix elements such that the community structure is visible. In this study, we propose an ordering algorithm based on the maximum-likelihood estimate of the ordered random graph model. We show that the proposed method allows us to more clearly identify community structures than the existing ordering algorithms.

I. INTRODUCTION

In a structural analysis of networks, extracting macroscopic structures from network data is a central task. For instance, there is a plethora of studies on extracting a community structure, which is commonly defined as a network that consists of several sets of densely connected vertices [1–7]. A banded structure is another type of macroscopic structure, in which the vertices with close indices are more likely to be connected. Extraction of the banded structure is referred to as the linear arrangement or graph layout problem in graph theory [8–15], also termed envelope reduction problem [9, 16–18].

The alignment of adjacency matrix elements, or the ordering of vertices, is critical for the visual identification of macroscopic structures in networks [19, 20]. Let us demonstrate two examples. In Fig. 1(a), the network has a community structure. Whereas the community structure is not visible when vertices are not carefully aligned as shown in the left matrix, the adjacency matrix exhibits a nearly block-diagonal structure when vertices are properly aligned as shown in the right matrix. In Fig. 1(b), the network has a banded structure. Again, the banded structure is not visible in the left matrix. However, when vertices are properly aligned as shown in the right matrix, nonzero elements are rendered to concentrated around the diagonal part.

Importantly, a network may exhibit both community and banded structures simultaneously. However, as demonstrated in Sec. V, previous linear arrangement methods often fail to capture community structures, whereas they allow for the identification of banded structures. In this study, we propose a linear arrangement

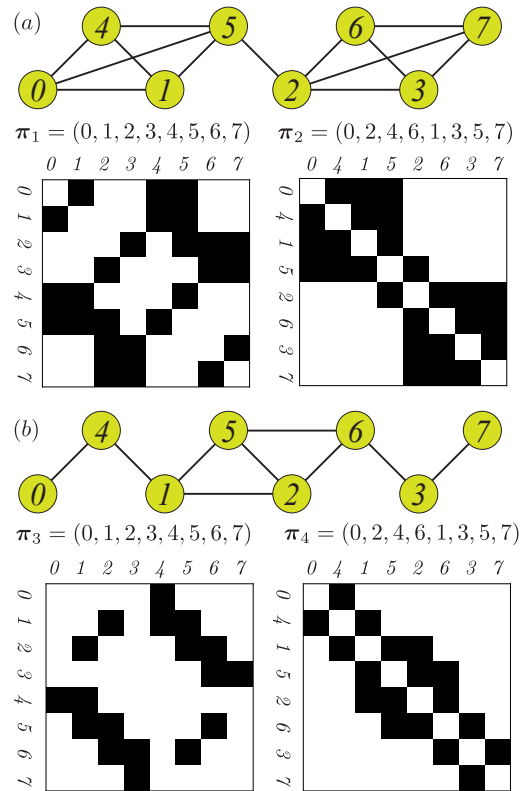


FIG. 1. Networks with (a) a community structure and (b) a banded structure. Each network is visualized using adjacency matrices with different alignment. We define π in Sec. II. For both networks, we align matrix elements according to two vertex sequences (π_1 and π_2 for (a), and π_3 and π_4 for (b)), where vertex i corresponds to the π_i th column (or row) of the adjacency matrix. The (π_i, π_j) element is filled with black when vertices i and j are connected. The numbers attached to the matrices represent the raw indices of the vertices.

* ochi@iis.u-tokyo.ac.jp

† kawamoto.tatsuro@aist.go.jp

method that allows us to capture community structures

in addition to banded structures. It is an inference method based on the maximum-likelihood estimate in which we use the ordered random graph model (ORGM) [21] as a network generative model.

The rest of the paper is organized as follows: In Sec. II, we introduce the basic terminology and related work. Following the introduction of the ORGM in Sec. III, we formulate the inference method and a greedy algorithm for the optimization of vertex sequence in Sec. IV. Section V describes the results of the application of the proposed algorithm to synthetic networks and real-world datasets. Finally, Sec. VI is devoted to a discussion of the proposed inference method.

II. DEFINITIONS AND RELATED WORK

Let $G = (V, E)$ be a network, where V ($|V| = N$) is the set of vertices and E ($|E| = M$) is the set of edges. Throughout this study, we consider undirected networks without multi-edges and self-loops. We denote the ordered set of raw vertex indices as $\{0, 1, \dots, N-1\} =: \mathcal{I}$. We assign each vertex to a raw vertex index $i \in \mathcal{I}$. We define the vertex sequence $\boldsymbol{\pi} = (\pi_0, \pi_1, \dots, \pi_{N-1})$ by a permutation of the raw vertex indices \mathcal{I} , where we use one-line notation [22, 23] to represent the permutation. For example, the raw vertex index $i \in \mathcal{I}$ is mapped to $\pi_i \in \mathcal{I}$ by $\boldsymbol{\pi}$. We denote the inferred vertex sequence by $\hat{\boldsymbol{\pi}}$. We let A be the adjacency matrix of a network, where $A_{ij} = 1$ if there exists an edge between vertices i and j , and 0 otherwise. Herein, A is a symmetric and binary matrix in which the diagonal elements are all zero.

Matrix reordering is considered in a variety of contexts. In the minimum linear arrangement problem, we search for the optimal vertex sequence that minimizes the following cost function [8]:

$$\sum_{i,j=0}^{N-1} A_{ij} |\pi_i - \pi_j|. \quad (1)$$

The exact minimization of Eq. (1) is \mathcal{NP} -complete [24], and several lower and upper bounds of the cost function and associated vertex ordering have been discussed in the literature [9, 11–15]. Reordering of symmetric matrices is considered in the context of envelope reduction problem for the efficient computation of sparse matrices. Several methods to reduce the size of the envelope, which is defined as a banded region around the diagonal part involving all nonzero elements in [16], have been proposed [9, 16, 17, 25, 26]. Reordering of rectangular matrices has been established as a seriation problem, which has its origin in archaeology [27, 28]. For example, the so-called consecutive ones problem considers the reordering of binary matrix such that the ones in each row are aligned consecutively [29–32]. Ranking of vertices is studied as a reordering problem of directed networks [33, 34]. Clustering methods associated with the ranking are reported

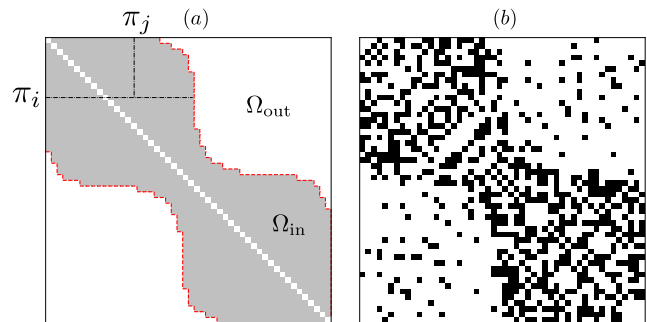


FIG. 2. Schematic of the ORGM. (a) The adjacency matrix is divided into the shaded region close to the diagonal (Ω_{in}) and the rest (Ω_{out}) by the envelope function F represented by red dashed lines. (b) An example of a network generated by (a) with $p_{in} = 0.5$ and $p_{out} = 0.1$.

in [35–37]. Recently, the reordering problem is also studied using neural networks [38, 39]. The ordering of vertices can also be considered in the context of latent space network model, as a way of one-dimensional embedding [40].

In this work, we employ two of the envelope reduction algorithms as a comparison with our method; spectral ordering and reverse Cuthill-McKee algorithm. Spectral ordering [9, 16–18] employs the eigenvector of the Laplacian matrix to find a vertex sequence that approximately minimizes the cost function

$$\sum_{i,j=0}^{N-1} A_{ij} (\pi_i - \pi_j)^2. \quad (2)$$

We sort vertices in the ascending (or descending) order of the eigenvector associated with the second-smallest eigenvalue of the Laplacian matrix. We employ the normalized Laplacian $\mathcal{L} = D^{-1/2} (D - A) D^{-1/2}$, where D is the degree matrix defined by $D = \text{diag}(d_0, d_1, \dots, d_{N-1})$ and $d_i := \sum_j A_{ij}$ is the degree. The reverse Cuthill-McKee algorithm [25, 26, 41] heuristically finds a vertex sequence. This algorithm permutes vertices using the breadth first search. See [41] for a detailed explanation.

III. ORDERED RANDOM GRAPH MODEL AND LIKELIHOOD FUNCTION

A. Ordered random graph model

The ORGM [21] is a random graph model that has heterogeneous connection probabilities between vertices based on the planted (or preassigned) vertex sequence $\boldsymbol{\pi}$. After the vertices are reordered based on $\boldsymbol{\pi}$, as illustrated in Fig. 2(a), we divide the upper-right elements of the adjacency matrix into two regions, Ω_{in} and Ω_{out} . These regions are separated by an envelope function $F(i) \in \mathcal{I}$, which is a discrete function with respect to the matrix

row index $i \in \mathcal{I}$, i.e.,

$$\begin{aligned}\Omega_{\text{in}} &= \{(\pi_i, \pi_j) \mid |\pi_i - \pi_j| \leq F(\pi_i), \pi_i < \pi_j\}, \\ \Omega_{\text{out}} &= \{(\pi_i, \pi_j) \mid |\pi_i - \pi_j| > F(\pi_i), \pi_i < \pi_j\}.\end{aligned}\quad (3)$$

Note that the envelope function is distinct from the envelope in the envelope reduction problem.

The vertices i and j are connected with probability p_{in} when the vertices satisfy $(\pi_i, \pi_j) \in \Omega_{\text{in}}$ and are connected with probability p_{out} when the vertices satisfy $(\pi_i, \pi_j) \in \Omega_{\text{out}}$. If we set $p_{\text{in}} > p_{\text{out}}$, vertices with close vertex indices are more likely to be connected. The envelope function F defines whether the vertices are deemed to be ‘‘close.’’ Therefore, as illustrated in Fig. 2(b), the ORGM can generate networks with a complex structure that is not restricted to a banded structure, which is referred to as the sequentially local structure [21].

The probability distribution for the set of the adjacency matrix elements $\{A_{ij}\}_{i < j}$ being generated by the ORGM is expressed as follows:

$$P(\{A_{ij}\}_{i < j} \mid \boldsymbol{\pi}, p_{\text{in}}, p_{\text{out}}, F) = \prod_{i < j} J_{ij}^{A_{ij}} (1 - J_{ij})^{1 - A_{ij}}, \quad (4)$$

where

$$\begin{aligned}J_{ij} &= \text{Prob}[A_{ij} = 1] \\ &= \begin{cases} p_{\text{in}} & \text{if } (\pi_i, \pi_j) \in \Omega_{\text{in}} \text{ or } (\pi_j, \pi_i) \in \Omega_{\text{in}}, \\ p_{\text{out}} & \text{if } (\pi_i, \pi_j) \in \Omega_{\text{out}} \text{ or } (\pi_j, \pi_i) \in \Omega_{\text{out}}. \end{cases}\end{aligned}\quad (5)$$

While the probability distribution by the Bernoulli distribution is straightforward, we hereafter assume a Poisson distribution instead as

$$P(\{A_{ij}\}_{i < j} \mid \boldsymbol{\pi}, p_{\text{in}}, p_{\text{out}}, F) = \prod_{i < j} \frac{J_{ij}^{A_{ij}}}{A_{ij}!} e^{-J_{ij}} \quad (6)$$

because it is easier to work with especially in the maximum-likelihood estimate. Although the Poisson distribution possibly generates multigraphs, in the sparse limit when $J_{ij} = \mathcal{O}(1/N)$ with $N \gg 1$, we can assume that p_{in} and p_{out} are sufficiently small, and thus Eqs. (4) and (6) are asymptotically identical.

In the inference problem, the vertex sequence $\boldsymbol{\pi}$ is the latent variable to be inferred and we estimate the envelope function F and the connection probabilities, p_{in} and p_{out} . For the maximum-likelihood inference of the vertex sequence $\hat{\boldsymbol{\pi}}$, we consider the log-likelihood function corresponding to Eq. (6):

$$\begin{aligned}L(\boldsymbol{\pi}, p_{\text{in}}, p_{\text{out}}, F \mid A) &= \sum_{i < j} (A_{ij} \log p_{\text{in}} - p_{\text{in}}) \\ &\quad (\pi_i, \pi_j) \in \Omega_{\text{in}} \text{ or } (\pi_j, \pi_i) \in \Omega_{\text{in}} \\ + \sum_{i < j} (A_{ij} \log p_{\text{out}} - p_{\text{out}}), \quad (7) \\ &\quad (\pi_i, \pi_j) \in \Omega_{\text{out}} \text{ or } (\pi_j, \pi_i) \in \Omega_{\text{out}}\end{aligned}$$

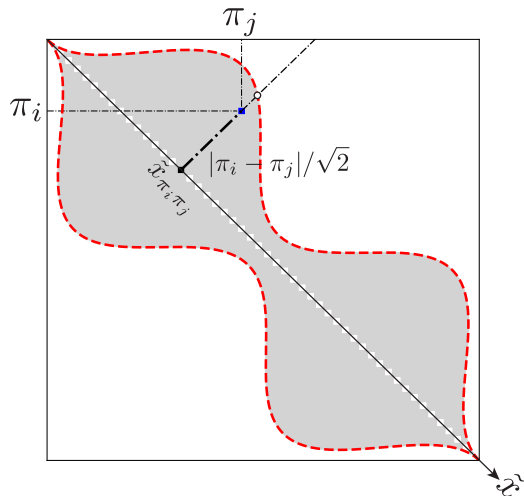


FIG. 3. Schematic of the ORGM parametrized by the continuous envelope function $\tilde{b}(\tilde{x})$ ($b(x)$) represented by a red dashed line, which is the boundary of the shaded Ω_{in} and unshaded regions Ω_{out} .

where we neglected the terms that are independent of the vertex sequence $\boldsymbol{\pi}$ and model parameters p_{in} , p_{out} , and F . The maximum-likelihood estimate of the ORGM for a given adjacency matrix A is defined as

$$\underset{\boldsymbol{\pi}, p_{\text{in}}, p_{\text{out}}, F}{\text{argmax}} L(\boldsymbol{\pi}, p_{\text{in}}, p_{\text{out}}, F \mid A). \quad (8)$$

The exact solution of Eq. (8) would be obtained by considering all possible vertex sequences, the discrete integer values of envelope function for each row, and connection probabilities. However, this is extremely expensive computationally.

To efficiently solve the maximum-likelihood estimate, we modify the definition of the envelope function F . Instead of a discrete envelope function on the matrix, as illustrate in Fig. 3, we treat the matrix as a two-dimensional plane such that (i, j) element of the matrix corresponds to the (i, j) coordinate and consider a continuous envelope function $\tilde{b}(\tilde{x})$ on this plane. We let the upper-left corner of the plane be the origin, $(0, 0)$. Given a point on the diagonal line, \tilde{x} ($0 \leq \tilde{x} \leq \sqrt{2}(N - 1)$) is the distance between $(0, 0)$ and the point, while $\tilde{b}(\tilde{x})$ represents the distance from the point to the boundary of Ω_{in} and Ω_{out} in the direction perpendicular to the diagonal line. When coordinate (π_i, π_j) is projected to the diagonal line, the distance between the origin and projected point is \tilde{x}_{π_i, π_j} , where $\tilde{x}_{ij} = (i + j)/\sqrt{2}$. The distance between (π_i, π_j) and the projected point is $|\pi_i - \pi_j|/\sqrt{2}$. Using the continuous envelope function, we redefine the regions Ω_{in} and Ω_{out} as

$$\begin{aligned}\Omega_{\text{in}} &= \left\{ (\pi_i, \pi_j) \mid |\pi_i - \pi_j| \leq \sqrt{2}\tilde{b}(\tilde{x}_{\pi_i, \pi_j}), \pi_i < \pi_j \right\}, \\ \Omega_{\text{out}} &= \left\{ (\pi_i, \pi_j) \mid |\pi_i - \pi_j| > \sqrt{2}\tilde{b}(\tilde{x}_{\pi_i, \pi_j}), \pi_i < \pi_j \right\}.\end{aligned}\quad (9)$$

Specifically, we define the continuous envelope function $\tilde{b}(\tilde{x})$ as a superposition of sine-squared functions:

$$\tilde{b}(\tilde{x}) = \sum_{k=1}^K a_k \sin^2 \left(\frac{\pi k}{\sqrt{2}(N-1)} \tilde{x} \right), \quad (10)$$

where K is a hyperparameter specifying the limit of complexity of the envelope function, which is treated as a fixed constant.

Hereafter, we rescale $\tilde{b}(\tilde{x})$, \tilde{x} , and \tilde{x}_{ij} such that

$$b(x) = \sqrt{2}\tilde{b}(\tilde{x}), \quad x = \frac{\tilde{x}}{\sqrt{2}}, \quad x_{ij} = \frac{\tilde{x}_{ij}}{\sqrt{2}} = \frac{i+j}{2}. \quad (11)$$

Therefore,

$$\begin{aligned} \Omega_{\text{in}} &= \{(\pi_i, \pi_j) \mid |\pi_i - \pi_j| \leq b(x_{\pi_i \pi_j}), \pi_i < \pi_j\}, \\ \Omega_{\text{out}} &= \{(\pi_i, \pi_j) \mid |\pi_i - \pi_j| > b(x_{\pi_i \pi_j}), \pi_i < \pi_j\}, \end{aligned} \quad (12)$$

$$b(x) = \sqrt{2} \sum_{k=1}^K a_k \sin^2 \left(\frac{\pi k}{N-1} x \right), \quad (13)$$

where $0 \leq x \leq N-1$. The coordinate (x_{ij}, x_{ij}) represents the projected point of (i, j) on the diagonal line.

The envelope function satisfies that $b(x) = 0$ at the both ends of the diagonal line, $x = 0$ and $x = N-1$. The envelope function is parametrized by the set of real-valued coefficients $\{a_k\} := \{a_1, \dots, a_K\}$. We restrict the set $\{a_k\}$ such that the envelope function should not leak out of the upper-triangle of the matrix; that is $0 \leq b(x) \leq \min\{2x, 2(N-1-x)\}$ for $0 \leq x \leq N-1$.

This modeling of the envelope function enables us to capture the community structures from the inferred vertex sequence. Note that, while the number of superpositions K may represent the number of groups, as demonstrated in Sec. V, it is not always the case. Although the inference will be computationally infeasible when K is considerably large, empirically, we found that $K \leq 2$ is often sufficient to identify community structures.

By using the continuous envelope function and the Heaviside step function

$$\Theta(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

the log-likelihood function Eq. (7) is expressed as

$$\begin{aligned} L(\boldsymbol{\pi}, p_{\text{in}}, p_{\text{out}}, \{a_k\} | A) &= (\log p_{\text{in}} - \log p_{\text{out}}) \sum_{i < j} A_{ij} \Theta(b(x_{\pi_i \pi_j}) - |\pi_i - \pi_j|) \\ &\quad - (p_{\text{in}} - p_{\text{out}}) \sum_{i < j} \Theta(b(x_{ij}) - |i - j|) \\ &\quad + M \log p_{\text{out}} - \frac{N(N-1)}{2} p_{\text{out}}. \end{aligned} \quad (15)$$

In the second term of Eq. (15), we used the fact that π_i and π_j can be replaced with i and j because this term is invariant under the permutation of vertices. As stated in Sec. IV B, this formulation renders the update of the vertex sequence more efficient.

IV. MAXIMUM-LIKELIHOOD ESTIMATE

In this section, we estimate model parameters as well as the vertex sequence $\hat{\boldsymbol{\pi}}$. We denote the estimated model parameters as \hat{p}_{in} , \hat{p}_{out} , and $\{\hat{a}_k\}$. We iteratively conduct the inference of the vertex sequence together with model-parameter learning until convergence. We show the pseudocode of this iterative process in Algorithm 1. It is usually infeasible to find the exact vertex sequence that maximizes the log-likelihood function by searching all possible solutions. Thus, we employ a greedy heuristic to infer the vertex sequence. For the efficient search of the optimal solution, we use the results of spectral ordering [18] as the initial vertex sequence, where nonzero elements have already been aligned near diagonal elements. We continue iteration until the variation of L gets smaller than a threshold ϵ_1 . To avoid being trapped into a local optimum, we conduct the entire optimization process several times with randomly chosen starting points with respect to $\{a_k\}$.

A. Parameter learning

Given an inferred vertex sequence $\hat{\boldsymbol{\pi}}$, we update \hat{p}_{in} and \hat{p}_{out} based on the saddle-point conditions of L . As a result, we obtain the following updating equations:

$$\hat{p}_{\text{in}} = \frac{\sum_{i < j} A_{ij} \Theta(b(x_{\hat{\pi}_i \hat{\pi}_j}) - |\hat{\pi}_i - \hat{\pi}_j|)}{\sum_{i < j} \Theta(b(x_{ij}) - |i - j|)}, \quad (16)$$

$$\hat{p}_{\text{out}} = \frac{\sum_{i < j} A_{ij} \Theta(-b(x_{\hat{\pi}_i \hat{\pi}_j}) + |\hat{\pi}_i - \hat{\pi}_j|)}{\sum_{i < j} \Theta(-b(x_{ij}) + |i - j|)}. \quad (17)$$

Before considering the update of the set of coefficients $\{\hat{a}_k\}$, we note that the log-likelihood function L is not differentiable with respect to $\{a_k\}$ because the Heaviside step function is discontinuous on the boundary of Ω_{in} and Ω_{out} . Instead, we approximate the log-likelihood function as follows:

$$\begin{aligned} L_{\beta}(\boldsymbol{\pi}, p_{\text{in}}, p_{\text{out}}, \{a_k\} | A) &= (\log p_{\text{in}} - \log p_{\text{out}}) \sum_{i < j} A_{ij} \Theta_{\beta}(b(x_{\pi_i \pi_j}) - |\pi_i - \pi_j|) \\ &\quad - (p_{\text{in}} - p_{\text{out}}) \sum_{i < j} \Theta_{\beta}(b(x_{ij}) - |i - j|) \\ &\quad + M \log p_{\text{out}} - \frac{N(N-1)}{2} p_{\text{out}}, \end{aligned} \quad (18)$$

where β is a hyperparameter and $\Theta_{\beta}(x)$ is a sigmoid function defined as

$$\Theta_{\beta}(x) = \frac{1}{1 + e^{-\beta x}}, \quad (19)$$

which approaches the Heaviside step function as $\beta \rightarrow \infty$.

To obtain the saddle-point condition for the set of coefficients $\{\hat{a}_k\}$, we solve the set of equations

$$\frac{\partial L_{\beta}}{\partial a_1} = \frac{\partial L_{\beta}}{\partial a_2} = \dots = \frac{\partial L_{\beta}}{\partial a_K} = 0. \quad (20)$$

Algorithm 1: The pseudocode of the maximum-likelihood estimate of the vertex sequence $\hat{\pi}$

Input: G
Output: $\hat{\pi}, \hat{p}_{\text{in}}, \hat{p}_{\text{out}}, \{\hat{a}_k\}$
Parameters : $K, \beta, \eta, \epsilon_1, \epsilon_2, N_s$

$\hat{\pi} \leftarrow \pi_{\text{spectral}};$ ▷ Initialize $\hat{\pi}$ by spectral ordering
 $\{\hat{a}_k\} \leftarrow \{a_k\}_{\text{init}};$ ▷ Initialize $\{\hat{a}_k\}$ with random value
 $\delta L \leftarrow \infty;$ ▷ Initialize
 $L_0 \leftarrow -\infty;$ ▷ Initialize

while $|\delta L| > \epsilon_1$ **do** ▷ Update \hat{p}_{in} and \hat{p}_{out} by Eq. (16),(17)
 $\{\hat{p}_{\text{in}}, \hat{p}_{\text{out}}\} \leftarrow \{p_{\text{in}}, p_{\text{out}}\};$
 while $|\nabla_{a_k} L_\beta| > \epsilon_2$ **do** ▷ Iteratively update $\{\hat{a}_k\}$ by Eq. (22)
 for $k = 1$ **to** K **do**
 $\hat{a}_k \leftarrow \hat{a}_k + \eta \frac{\partial L_\beta}{\partial a_k};$
 end
 end
 end
 for $s = 1$ **to** N_s **do** ▷ Calculate ΔL by Eq. (25)
 Randomly choose a pair of vertices $\{\ell, m\};$
 $\Delta L_{\ell m} \leftarrow \Delta L_{\ell m};$
 if $\Delta L_{\ell m} > 0$ **then** ▷ Update $\hat{\pi}$ by greedy algorithm
 $\{\hat{\pi}_\ell, \hat{\pi}_m\} \leftarrow \{\hat{\pi}_m, \hat{\pi}_\ell\};$
 end
 end
 $L \leftarrow L;$ ▷ Calculate L by Eq. (15)
 $\delta L \leftarrow L - L_0;$
 $L_0 \leftarrow L;$

end

The derivative $\partial L_\beta / \partial a_k$ is expressed as

$$\frac{\partial L_\beta}{\partial a_k} = -\frac{\log p_{\text{in}} - \log p_{\text{out}}}{4} \sum_{i < j} A_{ij} \frac{\sqrt{2}\beta \sin^2\left(\frac{\pi k}{N-1} x_{\pi_i \pi_j}\right)}{\cosh^2\left(\frac{\beta}{2}(b(x_{\pi_i \pi_j}) - |\pi_i - \pi_j|)\right)} + \frac{p_{\text{in}} - p_{\text{out}}}{4} \sum_{i < j} \frac{\sqrt{2}\beta \sin^2\left(\frac{\pi k}{N-1} x_{ij}\right)}{\cosh^2\left(\frac{\beta}{2}(b(x_{ij}) - |i - j|)\right)}. \quad (21)$$

Unfortunately, the set of equations Eq. (20) is implicit with respect to $\{a_k\}$ and, thus, is not easy to solve. Therefore, we employ the gradient descent method. That is, we maximize L_β by iteratively updating each \hat{a}_k as

$$\hat{a}_k \leftarrow \hat{a}_k + \eta \frac{\partial L_\beta}{\partial a_k}, \quad (22)$$

where η is the learning rate that determines the rate of update. We iterate the update until the convergence specified by $|\nabla_{a_k} L_\beta| = \sqrt{\sum_k (\partial L_\beta / \partial a_k)^2} < \epsilon_2$, where ϵ_2 is a predetermined threshold. For the efficient search of the saddle point, we decrease the learning rate as iteration goes on as $\eta = \eta_0 / t$, where η_0 is a hyperparameter and t is the iteration step.

Even when we use the gradient descent method, the update for $\{\hat{a}_k\}$ is still computationally expensive because we need to evaluate with respect to $N(N-1)/2$

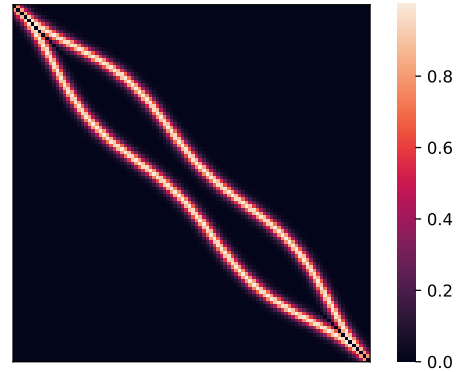


FIG. 4. Density plot of $\beta / \cosh^2(\beta(b(x_{ij}) - |i - j|)/2)$ with respect to i and j when $\beta = 1$. Here, we set $N = 100$, $K = 2$, and $a_1 = a_2 = 5$. The values are almost equal to zero except for the band around the envelope function.

pairs of vertices in the second summation in Eq. (21),

$$\sum_{i < j} \frac{\sqrt{2}\beta \sin^2\left(\frac{\pi k}{N-1}x_{ij}\right)}{\cosh^2\left(\frac{\beta}{2}(b(x_{ij}) - |i - j|)\right)}. \quad (23)$$

However, when β is sufficiently large, the contribution from a term in this summation is negligible when $|b(x_{ij}) - |i - j|| \gg 0$ because $\beta/\cosh^2(\beta x/2) \approx 0$ if $|\beta x| \gg 1$, as shown in Fig. 4. In other words, this term is well-approximated by the sum over the elements near the envelope function:

$$\sum_{d=0}^{2(N-1)} \sum_{\substack{i < j \\ i+j=d, |b(d/2)-|i-j|| \leq \delta}} \frac{\sqrt{2}\beta \sin^2\left(\frac{\pi k}{N-1}\frac{d}{2}\right)}{\cosh^2\left(\frac{\beta}{2}\left(b\left(\frac{d}{2}\right) - |i - j|\right)\right)}, \quad (24)$$

$$\Delta L_{\ell m} = (\log \hat{p}_{\text{in}} - \log \hat{p}_{\text{out}}) \left(\sum_{\substack{j > \ell \\ j \neq m}} A_{\ell j} (\Theta(b(x_{\hat{\pi}_m \hat{\pi}_j}) - |\hat{\pi}_m - \hat{\pi}_j|) - \Theta(b(x_{\hat{\pi}_\ell \hat{\pi}_j}) - |\hat{\pi}_\ell - \hat{\pi}_j|)) \right. \\ \left. + \sum_{\substack{i > m \\ i \neq \ell}} A_{mi} (\Theta(b(x_{\hat{\pi}_i \hat{\pi}_\ell}) - |\hat{\pi}_i - \hat{\pi}_\ell|) - \Theta(b(x_{\hat{\pi}_i \hat{\pi}_m}) - |\hat{\pi}_i - \hat{\pi}_m|)) \right). \quad (25)$$

Here, we used the fact that only the first summation in Eq. (15) is relevant to the variation of the log-likelihood function in terms of the vertex sequence as we mentioned before. For each pair of vertices (ℓ, m) , if $\Delta L_{\ell m} > 0$, we update the vertex sequence to be $\tilde{\pi}_\ell = \hat{\pi}_m$ and $\tilde{\pi}_m = \hat{\pi}_\ell$.

We summarize how this iterative process identifies an optimal solution. In the optimally aligned matrix, the density of nonzero elements in Ω_{in} is high, and the densely aligned nonzero elements are tightly surrounded by the envelope function such that zero elements are likely to be excluded from Ω_{in} . The combination of updates of the vertex sequence and the model parameters realizes such an alignment. First, \hat{p}_{in} and \hat{p}_{out} are updated to the densities of nonzero elements in Ω_{in} and Ω_{out} , as shown in Eq. (16) and (17), respectively. Second, we consider the update of the shape of the envelope function $\{\hat{a}_k\}$. Let us assume that one of the nonzero elements moves from Ω_{out} to Ω_{in} by altering the envelope function. In this case, Eq. (15) shows us that L increases by $(\log \hat{p}_{\text{in}} - \log \hat{p}_{\text{out}}) - (\hat{p}_{\text{in}} - \hat{p}_{\text{out}})$. Therefore, when Ω_{out} is sufficiently sparse, the nonzero elements are more likely to be included in Ω_{in} . In contrast, when a zero element moves from Ω_{in} to Ω_{out} , L increases by $\hat{p}_{\text{in}} - \hat{p}_{\text{out}}$. Thus, zero elements are more likely to be excluded from Ω_{in} . Finally, L is increased by updating $\hat{\pi}$ such that more

nonzero elements are included in Ω_{in} when $\hat{p}_{\text{in}} > \hat{p}_{\text{out}}$, as observed in Eq. (15). Thus, the vertex sequence is updated so that more nonzero elements are included in Ω_{in} .

V. RESULTS

A. Synthetic networks

We examine the performance of the maximum-likelihood estimate of the ORGM by applying to synthetic networks. In this section, we set the hyperparameters to be $\beta = 10$, $\eta_0 = 0.1$, $\epsilon_1 = 10^{-6}$, $\epsilon_2 = 0.1$, $\delta = 2$, and $n_s = 10$, and we consider 100 initial states for $\{a_k\}$ and employ the solution with the largest likelihood.

First, we apply the maximum-likelihood estimate to the networks generated by the ORGM and examine whether the algorithm can correctly infer the model parameters. We generate networks by setting the model parameters of the ORGM to be $N = 100$, $K = 1$, $p_{\text{in}} = 0.8$ with various values of a_1 and p_{out} . Figure 5 shows the model parameters estimated by the maximum-likelihood estimate with $K = 1$. We observe that the estimated

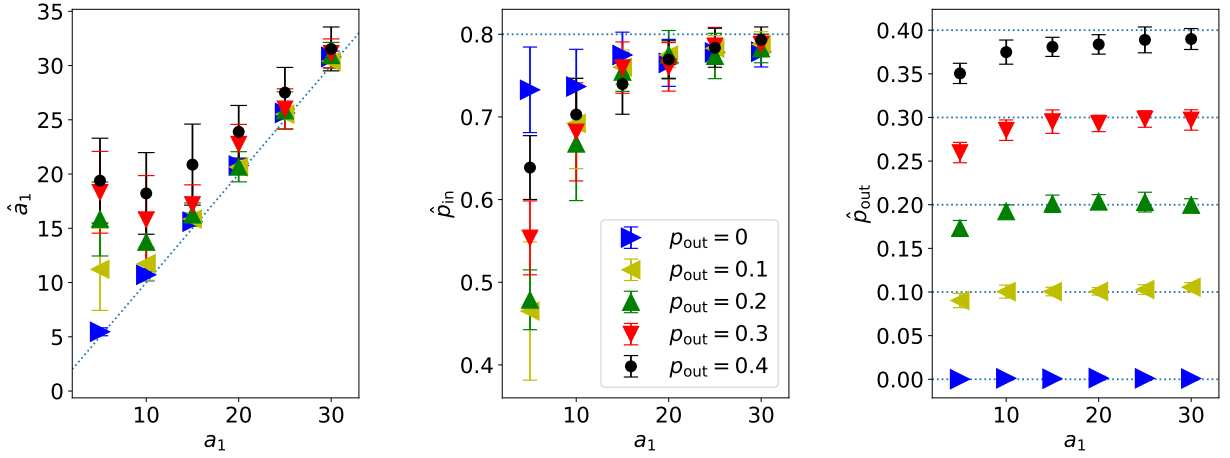


FIG. 5. The estimated model parameters of the ORGM by the maximum-likelihood estimate with $K = 1$. We generate networks by the ORGM with $N = 100$, $K = 1$, $p_{\text{in}} = 0.8$ with various values of a_1 and p_{out} . The dashed lines represent the cases where the estimated values coincide with the true values of the model. Each symbol and bar represent the average and standard deviation over 20 network instances, respectively.

model parameters are highly accurate when $p_{\text{out}} = 0$. As p_{out} increases, the estimated parameters divert from the true values for smaller values of a_1 . This experiment implies that the maximum-likelihood estimate of the ORGM can estimate the model parameters correctly when the number of nonzero elements in Ω_{in} is large. In Appendix A, we investigate how the performance of the estimate gets deteriorated through the experiments with smaller p_{in} .

Next, we examine to what extent our algorithm can infer the community structure. To this end, we apply the maximum-likelihood estimate to the networks with community structure generated by stochastic block model (SBM) [42]. The SBM generates a network as follows. We first define the number of group B , and assign the group memberships to the vertices $\sigma = (\sigma_0, \sigma_1, \dots, \sigma_{N-1})$, $\sigma_i \in \{0, 1, \dots, B-1\}$. Then, vertices i and j are connected with the probability q_{in} when $\sigma_i = \sigma_j$ and with the probability q_{out} when $\sigma_i \neq \sigma_j$. Hereafter, we assume that the size of each group is equal. The strength of community structure is represented by the ratio of the probabilities $\epsilon = q_{\text{out}}/q_{\text{in}}$. The adjacency matrix exhibits a block-diagonal structure when $\epsilon = 0$ while the network is uniformly random when $\epsilon = 1$. Note that q_{in} and q_{out} are uniquely determined when N , B , ϵ and the average degree $c = 2M/N$ are given.

We generate networks by setting the model parameter of the SBM to be $N = 50$ and $c = 6$ with various values of ϵ and we consider 20 network instances for each ϵ . We perform the maximum-likelihood estimate with $K = 1$ and $K = 2$. We consider the spectral ordering [18] and the reverse Cuthill-McKee algorithm [25, 26] for comparison. Figure 6 shows the estimated model parameters $\{\hat{a}_k\}$ of the ORGM and the heatmaps of the adjacency matrix based on the estimated vertex sequence. We observe that a_2 decreases as the strength of commu-

nity structure becomes weaker when $K = 2$. This result indicates that the envelope function has two peaks corresponding to the groups when ϵ is smaller, while these peaks decay as ϵ increases and the community structure becomes weaker. When $\epsilon = 0.05$, the community structure is visible through the heatmap using the maximum-likelihood method with $K = 1$. Furthermore, we can observe that the network exhibits a mixture of the community and banded structures when $K = 2$ is used. Note that it is natural that the instances of the SBM also exhibits a banded structure because a sparse uniform random graph often exhibits a banded structure when the vertices are properly aligned [21]. The banded structure is also visible in the heatmap using the spectral ordering. The reverse Cuthill-McKee algorithm fails to exhibit the banded structure whereas the community structure is visible. In contrast, when $\epsilon = 0.25$, we cannot find the community structure in all the four heatmaps while we can observe the banded structure in the heatmap using the spectral ordering.

Let us numerically evaluate how the estimated vertex sequence $\hat{\pi}$ is consistent with the preassigned group membership σ . We use normalized label continuity error [43], which is defined by

$$L(\hat{\pi}, \sigma) = \frac{N - B - \sum_{i'=0}^{N-2} \delta(\sigma(\hat{\pi}^{-1}(i')), \sigma(\hat{\pi}^{-1}(i'+1)))}{N - B - (N - 1)/B}, \quad (26)$$

where $\delta(x, y)$ is the Kronecker delta and $\sigma(\hat{\pi}^{-1}(i))$ is the group label of the vertex i' whose vertex index is $\hat{\pi}_{i'} = i$. The normalized label continuity error takes zero when all vertices in each group are aligned maximally consecutively. In other words, the result of ordering is consistent with the group labels. It takes a higher value

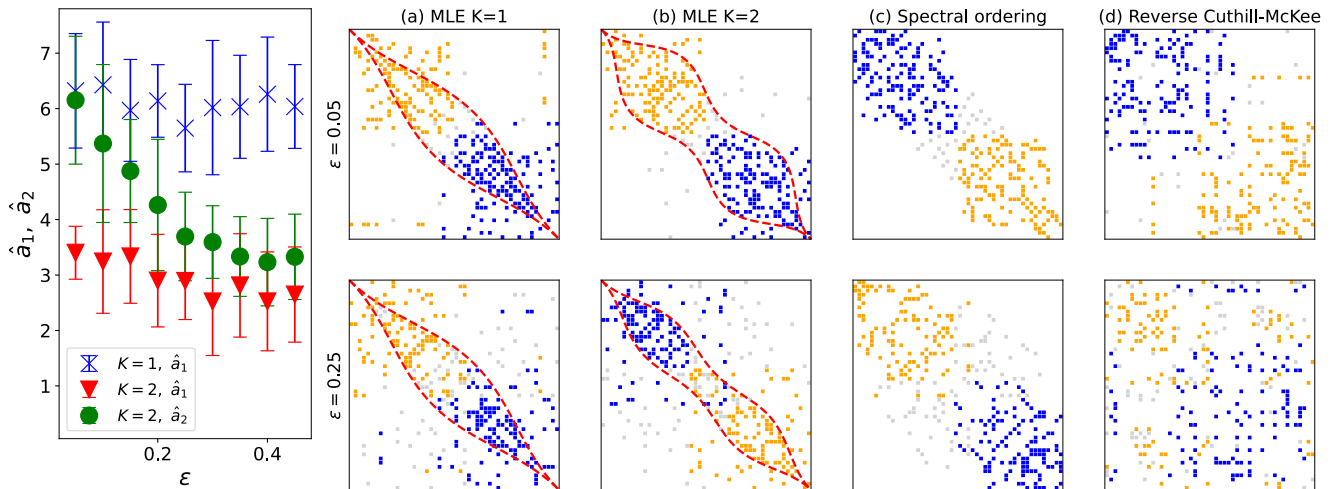


FIG. 6. (Left) Estimated $\{\hat{a}_k\}$ with the maximum-likelihood estimate when applied to the networks generated by the SBM with $N = 50$, $B = 2$ and $c = 6$. Each symbol and bar represent the average and standard deviation over 20 network instances, respectively. (Right) Heatmaps of adjacency matrices for a single network instance when $\epsilon = 0.05$ (upper row) and $\epsilon = 0.25$ (lower row). The dashed (red) curves represent the envelope function. We estimate the vertex sequence with four algorithms: the maximum-likelihood estimate (MLE) with (a) $K = 1$ and (b) $K = 2$, (c) spectral ordering, and (d) reverse Cuthill-McKee algorithm. If vertices i and j are connected and belong to the same group, the matrix element A_{ij} is filled with a dark color (blue) or pale color (yellow) depending on the preassigned group label. When i and j are connected and belong to the different groups, the matrix element A_{ij} is filled with an even paler color (gray).

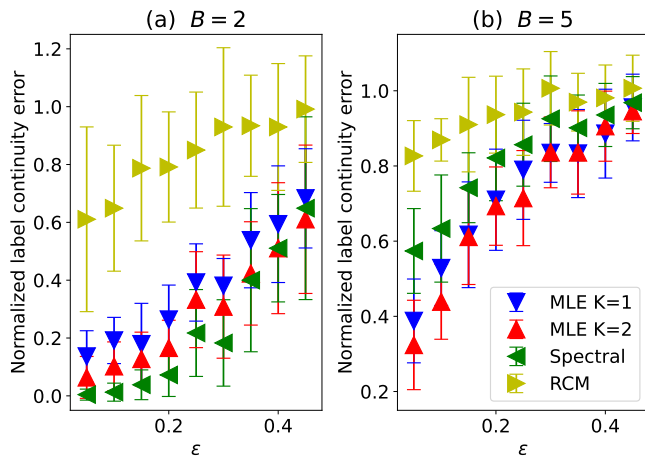


FIG. 7. The normalized label continuity error between the estimated vertex sequence and the preassigned group labels of the SBM using the maximum-likelihood estimate (MLE) ($K=1$), maximum-likelihood estimate (MLE) ($K=2$), spectral ordering (Spectral), and reverse Cuthill-McKee algorithm (RCM). The number of groups are (a) two and (b) five respectively. In both cases, we set $N = 50$ and $c = 6$, and each symbol and bar represent the average and standard deviation over 20 network instances, respectively.

when the neighboring vertices often belong to different groups.

In Fig. 7(a), we show the normalized label continuity error for the estimated vertex sequences when the number of groups is two ($B = 2$); this is the case we

considered in Fig. 6. When ϵ is small, we find that the vertex sequences from the maximum-likelihood estimate with $K = 2$ and the spectral ordering are highly consistent with the preassigned group labels, while the reverse Cuthill-McKee algorithm is less consistent (larger normalized label continuity error). We consider the SBM with five groups ($B = 5$) in Fig. 7(b). In this case, the vertex sequence of the maximum-likelihood estimate with both $K = 1$ and $K = 2$ are more consistent with the group labels than that of the spectral ordering contrary to the case with two groups. These experiments show that the maximum-likelihood estimate aligns vertices so that the community structure is visible even when the number of planted peaks in the envelope function K is smaller than B .

B. Real-world datasets

We demonstrate the performance of the maximum-likelihood estimate of the ORGM using three real-world network datasets that are often used in network science; the networks of Political Books [44], Les Misérables [45], and Football [1, 46] (Table I).

The visualization of the adjacency matrices based on the inferred vertex sequence is shown in Fig. 8. We show the results with $K = 1$ and $K = 2$, and set the parameters to be $\beta = 10$, $\eta_0 = 0.1$, $\epsilon_1 = 10^{-6}$, $\epsilon_2 = 0.1$, $\delta = 2$, and $n_s = 10$. For each case, we consider 1000 initial states for $\{a_k\}$ and employ the solution with the largest likelihood. Our results are compared with those of the

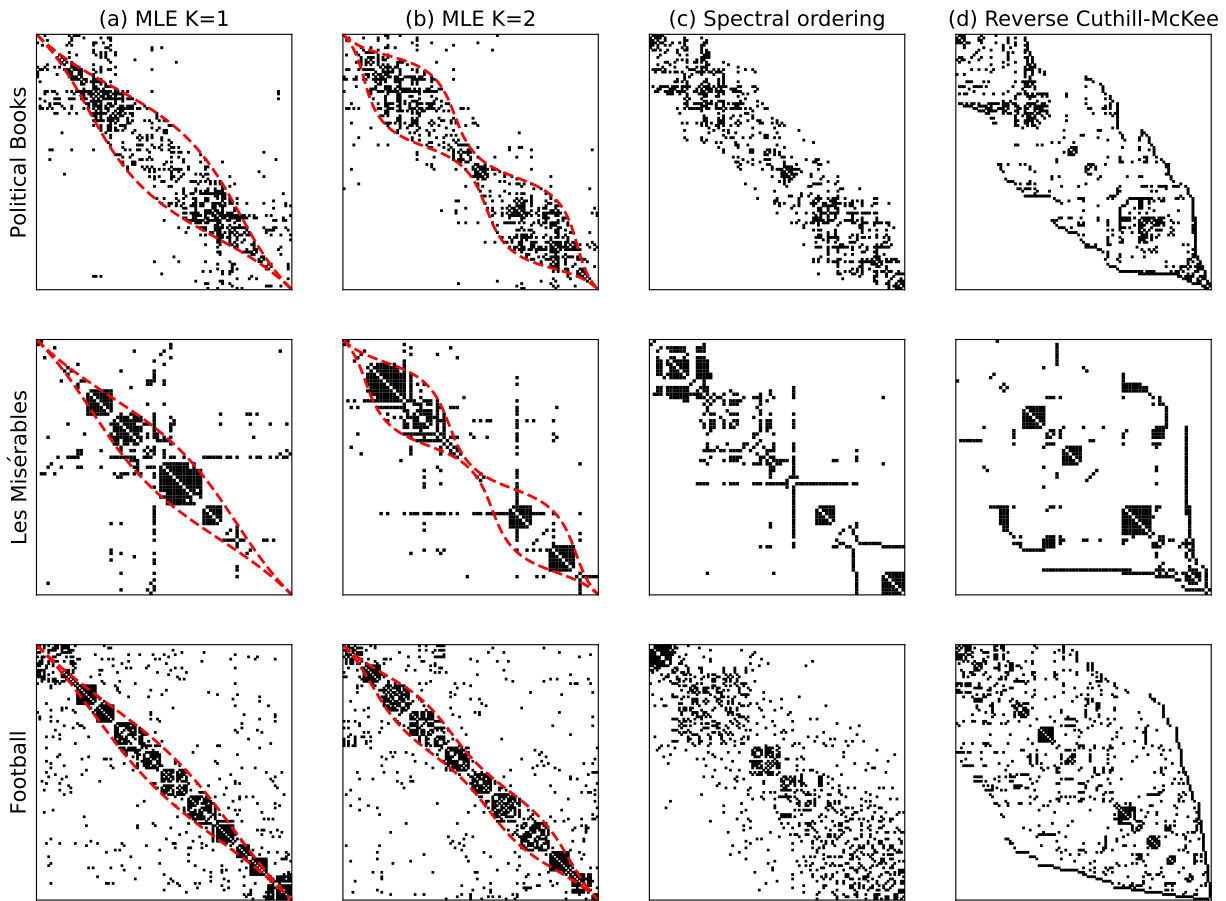


FIG. 8. Heatmaps of the adjacency matrices of three real-world datasets ordered by the four algorithms. The matrix element filled with black represents $A_{ij} = 1$. The red dashed line represents the estimated envelope function. The top, middle, and bottom panels represent the results for the networks of Political Books, Les Misérables, and Football, respectively. Each column represents the results of different algorithms: our maximum-likelihood estimates of the ORGM with (a) $K = 1$ and (b) $K = 2$, (c) spectral ordering, and (d) reverse Cuthill-McKee algorithm.

Dataset	N	M	Data description
Political Books [44]	105	441	Co-purchase network of books on US politics.
Les Misérables [45]	77	254	Character co-appearance network of <i>Les Misérables</i> .
Football [1, 46]	115	613	Network of American football games between Division IA colleges.

TABLE I. Description of empirical datasets. All the data can be loaded from graph-tool [47].

spectral ordering [18] and the reverse Cuthill-McKee algorithm [25, 26], which are also shown in Fig. 8.

The maximum-likelihood estimate allows us to capture the sequentially local structure through the linear arrangement for all three datasets. The envelope functions, illustrated as red curves, are inferred relatively close to the diagonal elements, and the nonzero elements, illus-

trated as black elements, are more likely to be placed inside the envelope function for both cases when $K = 1$ and $K = 2$. The maximum-likelihood estimate of the ORGM exhibits a block-diagonal structure more clearly compared with the results of the spectral ordering and the reverse Cuthill-McKee algorithm. We also show the results with $K = 3$ and $K = 4$ in Appendix B, where we observe that larger value of K does not necessarily allow us to capture clear community structures.

The result on the Political Books network with $K = 2$ exhibits two groups corresponding to the two peaks of the envelope function. We can also confirm that the Political Books network has a sequentially local structure that is a mixture of the community and banded structures. For the Les Misérables and Football networks, we can identify several clique-like groups even when $K = 2$ is employed. In contrast, the spectral ordering and the reverse Cuthill-McKee algorithm often focus more on banded structures.

We next examine whether the optimized vertex sequences are consistent with an existing community detection method. We use the Markov-chain Monte Carlo

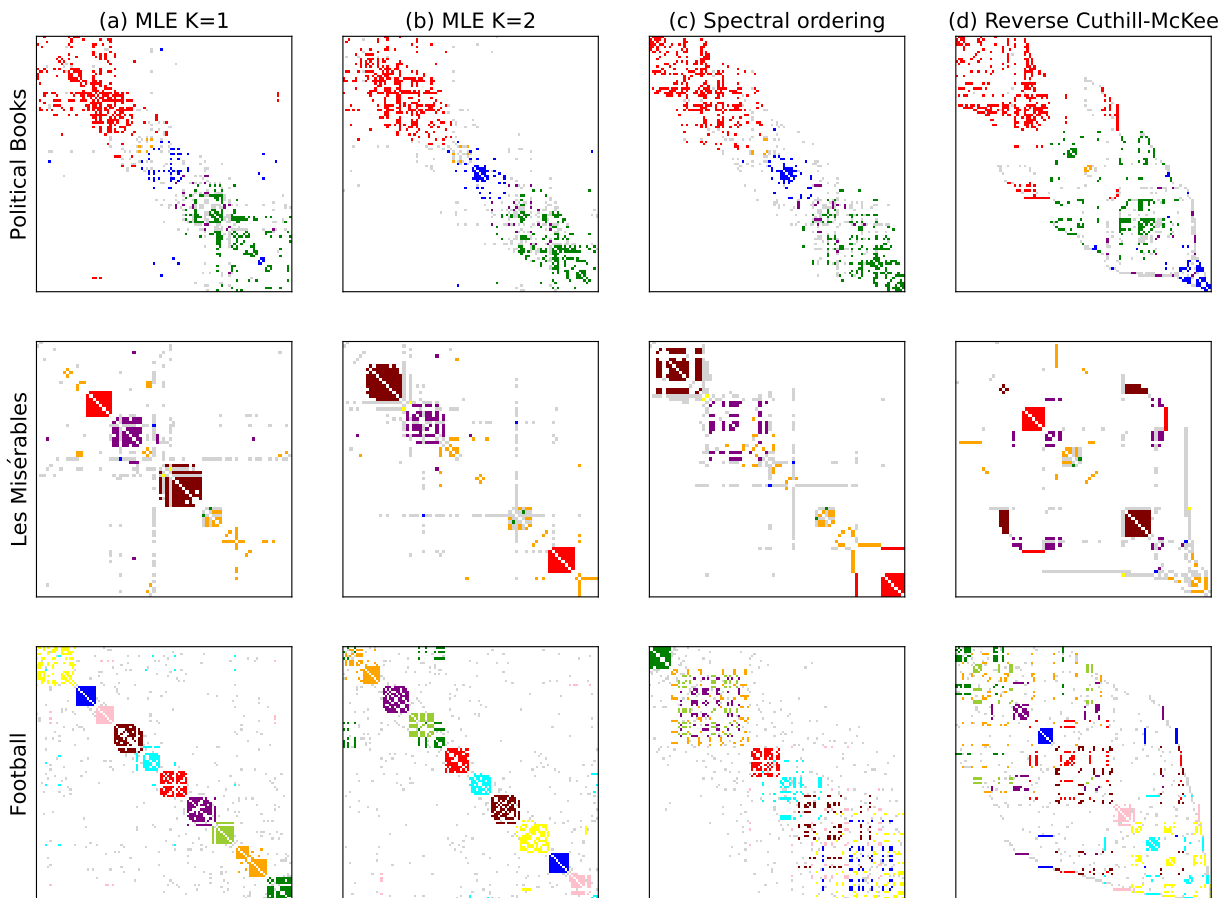


FIG. 9. Heatmaps of adjacency matrices that reflect the results of community detection. Matrix elements filled with the same (non-gray) color represent the vertex pairs belonging to the same group, while other nonzero elements are filled with the gray color. For the alignment of matrix elements, we use the same vertex sequence as in Fig. 8.

algorithm based on non-degree-corrected version of the SBM [48, 49], which is implemented in *graph-tool* [47]. This algorithm automatically determines the number of groups. We obtained five groups for the Political Books network, seven for the Les Misérables network, and 10 for the Football network. We can visually confirm from Fig. 9 that the vertices in the same group are more closely aligned, indicating that the vertex sequence obtained via our linear alignment algorithm is consistent with the results of the community detection algorithm.

Interestingly, in the Political Books network, the optimal vertex sequence shows that the two large groups, which are respectively represented by red and green, are connected via a small number of vertices, while such vertices in the middle are regarded to be forming another group, which is represented by blue, in the community detection method. That is, the group in the middle serves as a bridge at the ambiguous boundary of two large groups. Note that, if we inferred the group labels only and aligned the vertices based on those labels, we would simply conclude that the network has a community structure with two large groups and one small group. However, a careful alignment of the vertices allows us to

interpret that the small group is in fact a subgraph that smoothly connect the other large groups. Therefore, we can obtain a better understanding of a dataset by applying both linear alignment and community detection algorithms when the results of these two algorithms are consistent with each other.

VI. DISCUSSION

In this study, we proposed a linear arrangement algorithm based on statistical modeling. We employed the ORGM as a generative model of networks and performed its maximum-likelihood estimate. Our statistical framework to infer the sequentially local structure provides a new perspective to the community detection in a network. While we employed the maximum-likelihood estimate in this work, our framework can be extended to Bayesian inference.

Our modeling is more flexible than the classical linear arrangement methods that are based on a cost function, such as Eq. (1), and thus, our method allows us to capture a sequentially local structure that is a mix-

ture of community and banded structures. For example, in the Football network, the community structure is not observed in the results of the spectral ordering and the reverse Cuthill-McKee algorithm. This is because these methods strongly penalize the nonzero elements far from the diagonal part. In contrast, our method is not prone to such a problem because it penalizes matrix elements in a more flexible manner.

Our linear arrangement algorithm can detect community structures with ambiguous boundaries, as we observed from the result of the Political Books network (Fig. 9). In contrast, many of the previous community detection methods perform “hard clustering”, in which the detected groups have clear boundaries because a group label is assigned to each vertex. Such a hard clustering method is not suitable for some networks, and several other flexible methods were proposed in the literature. Overlapping community detection methods [50, 51] deal with ambiguous boundaries by assigning multiple group labels to each vertex. Bayesian inference methods [5, 52] infer a probability distribution of group assignment for each vertex. Although the interpretation of ambiguous boundaries of groups in these methods is not identical, they achieve “soft clustering” using a higher-order group membership. Our algorithm simply estimates the vertex sequence and allows us to visually identify the fuzziness of community structures without introducing higher-order group memberships.

Another advantage of our approach is that we neither need to give nor infer the number of groups. For example, in clustering methods, such as an inference method based on the maximum-likelihood estimate of the SBM [53] and spectral clustering [54], the number of groups needs to be provided as input. In some methods, such as the Bayesian inference, the number of groups is determined in a nonparametric manner [5, 52]. In contrast, as observed in the results of the Les Misérables and Football networks (Fig. 8), we can visually estimate an appropriate number of groups when the vertices are optimally aligned. There are pros and cons to our method. Specification of the number of groups is ambiguous compared with the clustering methods. On the other hand, our approach is more flexible for detecting the mixture of the community and banded structures because the specification of the number of groups in the clustering methods restricts the structures to be detected.

In our method, it is important that $p_{\text{in}} > p_{\text{out}}$ is satisfied. When $p_{\text{in}} > p_{\text{out}}$, as described in Sec. IV B, the nonzero elements tend to be included in Ω_{in} as the vertex sequence is updated. In contrast, when $p_{\text{in}} < p_{\text{out}}$, the nonzero elements tend to be excluded from Ω_{in} as the vertex sequence is updated. Conducting the spectral ordering as a preprocess helps us prevent this problem, in addition to the efficient search of the optimal vertex sequence, because the spectral ordering aligns more nonzero elements near the diagonal elements.

We characterized the envelope function as a superposition of the sine-squared functions. This modeling makes

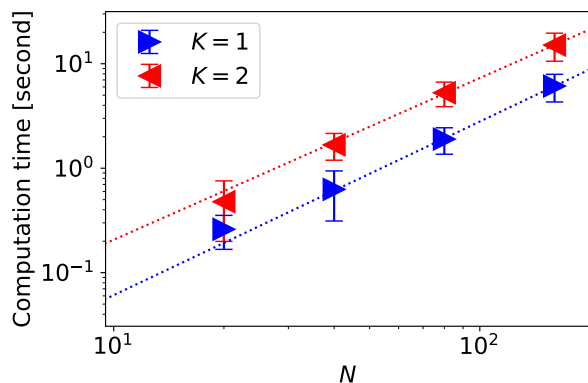


FIG. 10. Computation time of the maximum-likelihood estimate for a single initial value of $\{a_k\}$. We consider regular random networks with $c = 6$, and perform the maximum-likelihood estimate with $K = 1$ and $K = 2$. Each symbol and bar respectively represent the average and the standard deviation of the 100 samples of the initial value, and the dotted lines is the fitted curve by $y = ax^b$ using the least squares. When $K = 1$, $a = 1.33 \times 10^{-3}$ and $b = 1.66$, and when $K = 2$, $a = 5.85 \times 10^{-3}$ and $b = 1.55$. For the calculation, we use the 2020 Macbook Pro with Apple M1 chip.

computation efficient because we have only to tune $\{a_k\}$ and is effective for identifying community structures. However, other modelings are also possible, and exploring a better modeling for the envelope function would be a possible research direction. In addition, identifying the optimal number of sine waves K is another task. In this study, although we have treated K as a hyperparameter and successfully found the sequentially local structures with $K = 1$ and $K = 2$, a good model selection method can potentially find an optimal value of K . For example, it can be done by considering a Bayesian framework and adding a prior distribution of K .

Finding an optimal vertex sequence generally requires a large computational cost. To make the optimization tractable, we restricted the envelope function to a superposition of sine-squared functions in the proposed method. We also applied the transformation in Eq. (15) and the approximation in Eq. (21) to the log-likelihood function. Furthermore, we employed a greedy heuristic to realize a fast optimization. Despite these treatments, the maximum-likelihood estimate of the ORGM is still computationally expensive. However, our method at least improves the efficiency of the optimization compared with the brute-force method which requires $\mathcal{O}(N!)$ operations. In Fig. 10, we show the computation time to find the optimal vertex sequence when we apply the maximum-likelihood estimate to the regular random networks. We find that the computation time increases polynomially as the number of vertices increases, with the exponent less than two. In addition, although the computational cost increases when we use a larger value of K , the exponent is almost same between the cases with $K = 1$ and $K = 2$.

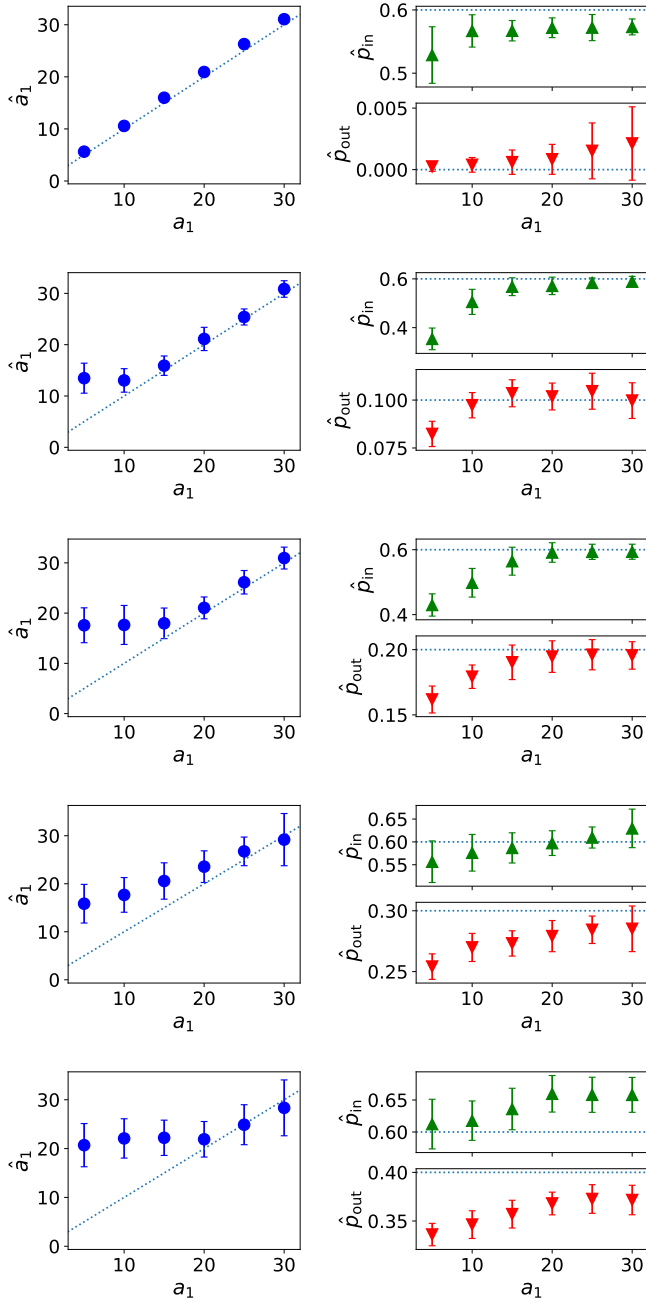


FIG. 11. The estimated model parameters of the ORGM by the maximum-likelihood estimate with $K = 1$. The networks are generated by the ORGM with $N = 100$, $K = 1$ and $p_{\text{in}} = 0.6$. The symbols, bars, and dashed lines are used in the same manner as in Fig. 5.

Appendix A: Accuracy of the ORGM parameters

In the main text, we observed that the maximum-likelihood estimate correctly infers the true model parameters of the ORGM for larger values of a_1 when $p_{\text{in}} = 0.8$. Here, we examine how the performance of the maximum-likelihood estimate gets deteriorated as we make p_{in} and p_{out} closer to those for uniformly random networks. Fig-

ure 11 shows the performance of the ORGM parameter estimation. Herein, we set $p_{\text{in}} = 0.6$ for the true parameter and considered various values of p_{out} . As observed in the case with $p_{\text{in}} = 0.8$ in the main text, a_1 is over-estimated when the true value is small; we confirm that this tendency is enhanced as p_{in} becomes smaller. For the networks with $p_{\text{out}} \leq 0.2$, the estimates \hat{p}_{in} and \hat{p}_{out} are accurate whenever \hat{a}_1 is accurate. However, for the networks with larger values of p_{out} , \hat{p}_{in} and \hat{p}_{out} are not accurate even when a_1 and \hat{a}_1 apparently coincide, implying that the maximum-likelihood estimate may have reached its performance limit.

Appendix B: Maximum-likelihood estimate with $K = 3$ and $K = 4$ on real-world datasets

In the main text, we showed the result of the maximum-likelihood estimates with $K = 1$ and $K =$

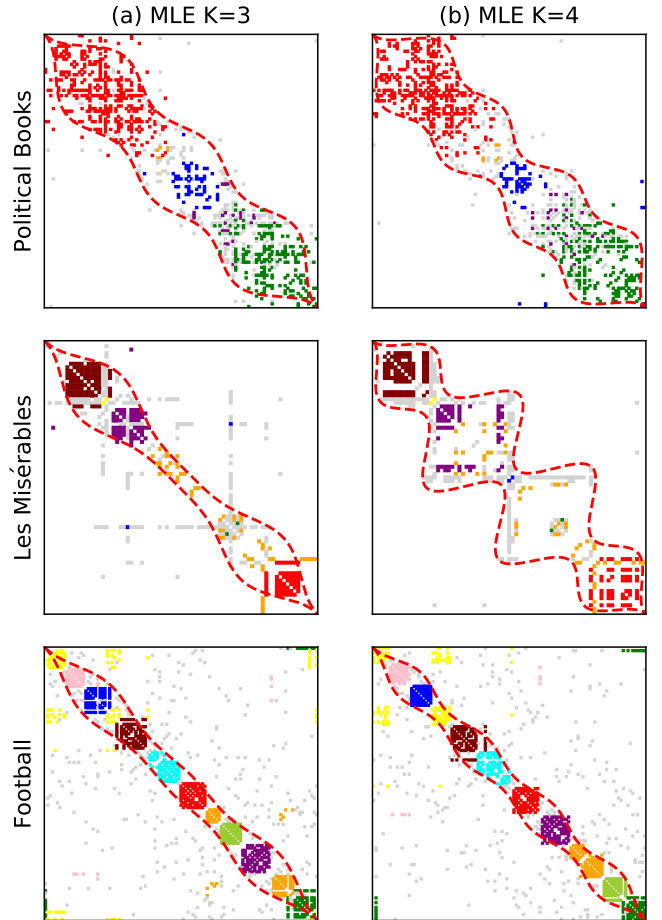


FIG. 12. Heatmaps of adjacency matrices aligned according to the results of our maximum-likelihood estimates of the ORGM with (a) $K = 3$ and (b) $K = 4$. The red dashed line again represents the estimated envelope function. The filling colors of each element represent the same results of community detection as in Fig. 9.

2. Here, we show the results with of the maximum-likelihood estimates with $K = 3$ and $K = 4$ in Fig. 12. We observe that, except for the result of the Les Misérables network with $K = 4$, the inferred vertex sequence is similar to the results with $K = 2$, while the envelope functions are more complicated. In contrast, in the result of the Les Misérables network with $K = 4$, the clique-like structure is no longer visible because the log-likelihood function is larger when the elements associated with a hub at the center of the matrix are included in Ω_{in} . This example illustrates that a larger value of K does not always exhibits a finer community structure and implies the existence of an optimal value of K .

ACKNOWLEDGMENTS

This work was supported by JST ACT-X Grant No. JPMJAX21A8 (T.K.), JSPS KAKENHI 19H01506, 22H00827 (T.K.), and Quantum Science and Technology Fellowship Program (Q-STEP) (M.O.).

DATA AVAILABILITY

The code of the maximum-likelihood estimate is available at [55].

-
- [1] M. Girvan and M. E. J. Newman, Community structure in social and biological networks, *Proc. Natl. Acad. Sci. U.S.A.* **99**, 7821 (2002).
- [2] M. Rosvall and C. T. Bergstrom, Maps of random walks on complex networks reveal community structure, *Proc. Natl. Acad. Sci. U.S.A.* **105**, 1118 (2008).
- [3] S. Fortunato, Community detection in graphs, *Phys. Rep.* **486**, 75 (2010).
- [4] E. Estrada, N. Hatano, and M. Benzi, The physics of communicability in complex networks, *Phys. Rep.* **514**, 89 (2012).
- [5] T. P. Peixoto, Hierarchical block structures and high-resolution model selection in large networks, *Phys. Rev. X* **4**, 011047 (2014).
- [6] S. Fortunato and D. Hric, Community detection in networks: A user guide, *Phys. Rep.* **659**, 1 (2016).
- [7] S. Fortunato and M. E. J. Newman, 20 years of network community detection, *Nature Phys.* **18**, 848 (2022).
- [8] L. H. Harper, Optimal assignments of numbers to vertices, *J. Soc. Ind. Appl. Math.* **12**, 131 (1964).
- [9] M. Juvan and B. Mohar, Optimal linear labelings and eigenvalues of graphs, *Discrete Appl. Math.* **36**, 153 (1992).
- [10] J. Díaz, J. Petit, and M. Serna, A survey of graph layout problems, *ACM Comput. Surv.* **34**, 313 (2002).
- [11] J. Petit, Experiments on the minimum linear arrangement problem, *J. Exp. Algorithmics* **8** (2003).
- [12] S. Rao and A. W. Richa, New approximation techniques for some linear ordering problems, *SIAM J. Comput.* **34**, 388 (2005).
- [13] N. R. Devanur, S. A. Khot, R. Saket, and N. K. Vishnoi, Integrality gaps for sparsest cut and minimum linear arrangement problems, in *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing* (2006) pp. 537–546.
- [14] A. Caprara, A. N. Letchford, and J.-J. Salazar-González, Decorous lower bounds for minimum linear arrangement, *INFORMS J. Comput.* **23**, 26 (2011).
- [15] H. Seitz, *Contributions to the minimum linear arrangement problem*, Ph.D. thesis (2010).
- [16] S. T. Barnard, A. Pothen, and H. Simon, A spectral algorithm for envelope reduction of sparse matrices, *Numer. Linear Algebra Appl.* **2**, 317 (1995).
- [17] A. George and A. Pothen, An analysis of spectral envelope reduction via quadratic assignment problems, *SIAM J. Matrix Anal. Appl.* **18**, 706 (1997).
- [18] C. Ding and X. He, Linearized cluster assignment via spectral ordering, in *Proceedings of the Twenty-First International Conference on Machine Learning, ICML '04* (Association for Computing Machinery, New York, NY, USA, 2004) p. 30.
- [19] L. G. S. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney, Think locally, act locally: Detection of small, medium-sized, and large communities in large networks, *Phys. Rev. E* **91**, 012821 (2015).
- [20] M. Sales-Pardo, R. Guimera, A. A. Moreira, and L. A. N. Amaral, Extracting the hierarchical organization of complex systems, *Proc. Natl. Acad. of Sci.* **104**, 15224 (2007).
- [21] T. Kawamoto and T. Kobayashi, Sequential locality of graphs and its hypothesis testing, *arXiv preprint arXiv:2111.11267* (2021).
- [22] B. E. Sagan, *The symmetric group: representations, combinatorial algorithms, and symmetric functions*, Vol. 203 (Springer Science & Business Media, 2013).
- [23] https://en.wikipedia.org/wiki/Permutation#One-line_notation.
- [24] M. Garey, D. Johnson, and L. Stockmeyer, Some simplified np-complete graph problems, *Theor. Comput. Sci.* **1**, 237 (1976).
- [25] E. Cuthill and J. McKee, Reducing the bandwidth of sparse symmetric matrices, in *Proceedings of the 1969 24th national conference* (1969) pp. 157–172.
- [26] A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite* (Prentice Hall Professional Technical Reference, 1981).
- [27] W. S. Robinson, A method for chronologically ordering archaeological deposits, *Am. Antiq.* **16**, 293 (1951).
- [28] I. Liiv, Seriation and matrix reordering methods: An historical overview, *Stat. Anal. Data Min.: The ASA Data Science Journal* **3**, 70 (2010).
- [29] D. Kendall, Incidence matrices, interval graphs and seriation in archeology, *Pac. J. Math* **28**, 565 (1969).
- [30] J. E. Atkins, E. G. Boman, and B. Hendrickson, A spectral algorithm for seriation and the consecutive ones problem, *SIAM J. Sci. Comput.* **28**, 297 (1998).
- [31] F. Fogel, R. Jenatton, F. Bach, and A. d’Aspremont, Convex relaxations for permutation problems, in *Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS'13* (Curran Associates Inc., Red Hook, NY, USA, 2013) pp. 1016–1024.

- [32] N. Vuokko, Consecutive ones property and spectral ordering, in *Proceedings of the SIAM International Conference on Data Mining, SDM* (SIAM, 2010) pp. 350–360.
- [33] S. Brin and L. Page, The anatomy of a large-scale hypertextual web search engine, *Comput. netw. ISDN syst.* **30**, 107 (1998).
- [34] C. De Bacco, D. B. Larremore, and C. Moore, A physical model for efficient ranking in networks, *Sci. Adv.* **4**, eaar8260 (2018).
- [35] T. P. Peixoto, Ordered community detection in directed networks, *Phys. Rev. E* **106**, 024305 (2022).
- [36] E. Letizia, P. Barucca, and F. Lillo, Resolution of ranking hierarchies in directed networks, *PloS one* **13**, e0191604 (2018).
- [37] L. Iacovissi and C. De Bacco, The interplay between ranking and communities in networks, *Sci. Rep.* **12**, 1 (2022).
- [38] C. Watanabe and T. Suzuki, Autoll: Automatic linear layout of graphs based on deep neural network, in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)* (IEEE, 2021) pp. 1–10.
- [39] O.-H. Kwon, C.-H. Kao, C.-H. Chen, and K.-L. Ma, A deep generative model for reordering adjacency matrices, *IEEE Trans. Vis. Comput. Graph.* (2022).
- [40] M. E. J. Newman, *Networks* (Oxford university press, 2018).
- [41] G. H. Golub and C. F. Van Loan, *Matrix computations* (JHU press, 2013).
- [42] P. W. Holland, K. B. Laskey, and S. Leinhardt, Stochastic blockmodels: First steps, *Social networks* **5**, 109 (1983).
- [43] T. Kawamoto, M. Ochi, and T. Kobayashi, Consistency between ordering and clustering methods for graphs, *arXiv preprint arXiv:2208.12933* (2022).
- [44] M. E. J. Newman, Modularity and community structure in networks, *Proc. Natl. Acad. Sci. U.S.A.* **103**, 8577 (2006).
- [45] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*, Vol. 1 (ACM Press: New York, 1993).
- [46] T. S. Evans, Clique graphs and overlapping communities, *J. Stat. Mech.: Theory Exp.* **2010** (12), P12037.
- [47] <https://graph-tool.skewed.de>.
- [48] T. P. Peixoto, Efficient monte carlo and greedy heuristic for the inference of stochastic block models, *Phys. Rev. E* **89**, 012804 (2014).
- [49] T. P. Peixoto, Model selection and hypothesis testing for large-scale network models with overlapping groups, *Phys. Rev. X* **5**, 011033 (2015).
- [50] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, Link communities reveal multiscale complexity in networks, *Nature* **466**, 761 (2010).
- [51] J. Xie, S. Kelley, and B. K. Szymanski, Overlapping community detection in networks: The state-of-the-art and comparative study, *ACM comput. surv.* **45**, 1 (2013).
- [52] M. E. J. Newman and G. Reinert, Estimating the number of communities in a network, *Phys. Rev. Lett.* **117**, 078301 (2016).
- [53] B. Karrer and M. E. J. Newman, Stochastic blockmodels and community structure in networks, *Phys. Rev. E* **83**, 016107 (2011).
- [54] U. Von Luxburg, A tutorial on spectral clustering, *Statistics and computing* **17**, 395 (2007).
- [55] https://github.com/MasakiOchi/MLE_ORGM.