

Extracting fuzzy clusters from massive attributed graphs using Markov lumpability optimization

Kai-Yue Jiang(蒋凯悦), Li-Heng Xu(徐力恒), Shi-Pei Lin(林诗佩), Li-Yang Zhou(周李阳), Hui-Jia Li(李慧嘉), and Ge Gao(高歌)

Citation: Chin. Phys. B, 2025, 34 (**10**): 108903. DOI: 10.1088/1674-1056/addaa2
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Influence of negative information dissemination and vaccination behavioral decision-making on epidemic spreading in a three-layer network

Liang'an Huo(霍良安) and Leyao Yin(尹乐瑶)
Chin. Phys. B, 2025, 34 (**6**): 068902. DOI: 10.1088/1674-1056/adc402

Impact of environmental factors on the coevolution of information-emotions-epidemic dynamics in activity-driven multiplex networks

Liang'an Huo(霍良安), Bingjie Liu(刘炳杰), and Xiaomin Zhao(赵晓敏)
Chin. Phys. B, 2024, 33 (**12**): 128903. DOI: 10.1088/1674-1056/ad7df5

Source localization in signed networks with effective distance

Zhi-Wei Ma(马志伟), Lei Sun(孙蕾), Zhi-Guo Ding(丁智国), Yi-Zhen Huang(黄宜真), and Zhao-Long Hu(胡兆龙)
Chin. Phys. B, 2024, 33 (**2**): 028902. DOI: 10.1088/1674-1056/ad1482

Identifying influential spreaders in social networks: A two-stage quantum-behaved particle swarm optimization with Lévy flight

Pengli Lu(卢鹏丽), Jimao Lan(揽继茂), Jianxin Tang(唐建新), Li Zhang(张莉), Shihui Song(宋仕辉), and Hongyu Zhu(朱虹羽)
Chin. Phys. B, 2024, 33 (**1**): 018901. DOI: 10.1088/1674-1056/acd3e0

Contagion dynamics on adaptive multiplex networks with awareness-dependent rewiring

Xiao-Long Peng(彭小龙) and Yi-Dan Zhang(张译丹)
Chin. Phys. B, 2021, 30 (**5**): 058901. DOI: 10.1088/1674-1056/abe1ab

---

Extracting fuzzy clusters from massive attributed graphs using Markov lumpability optimization

Kai-Yue Jiang(蒋凯悦)[1,†], Li-Heng Xu(徐力恒)[1,†], Shi-Pei Lin(林诗佩)[1,†], Li-Yang Zhou(周李阳)[1,†], Hui-Jia Li(李慧嘉)[1,‡], and Ge Gao(高歌)[2,§]

[1] *School of Statistics and Data Science, LPMC and KLMDASR, Nankai University, Tianjin* 300074, *China*
[2] *Management School, Beijing Sport University, Beijing* 100084, *China*

(Received 7 March 2025; revised manuscript received 11 May 2025; accepted manuscript online 20 May 2025)

Attributed graph clustering plays a vital role in uncovering hidden network structures, but it presents significant challenges. In recent years, various models have been proposed to identify meaningful clusters by integrating both structural and attribute-based information. However, these models often emphasize node proximities without adequately balancing the efficiency of clustering based on both structural and attribute data. Furthermore, they tend to neglect the critical fuzzy information inherent in attributed graph clusters. To address these issues, we introduce a new framework, Markov lumpability optimization, for efficient clustering of large-scale attributed graphs. Specifically, we define a lumped Markov chain on an attribute-augmented graph and introduce a new metric, *Markov lumpability*, to quantify the differences between the original and lumped Markov transition probability matrices. To minimize this measure, we propose a conjugate gradient projection-based approach that ensures the partitioning closely aligns with the intrinsic structure of fuzzy clusters through conditional optimization. Extensive experiments on both synthetic and real-world datasets demonstrate the superior performance of the proposed framework compared to existing clustering algorithms. This framework has many potential applications, including dynamic community analysis of social networks, user profiling in recommendation systems, functional module identification in biological molecular networks, and financial risk control, offering a new paradigm for mining complex patterns in high-dimensional attributed graph data.

Keywords: attributed clustering, Markov chain, lumped random walk, fuzzy clusters, optimization

PACS: 89.75.Da, 02.50.Ey, 07.05.Tp, 05.40.–a

DOI: 10.1088/1674-1056/addaa2 **CSTR:** 32038.14.CPB.addaa2

1. Introduction

Attributed graphs,[1,2] in which nodes represent users, attributes capture user profiles, and edges denote interactions, have become increasingly prevalent in various domains, particularly in social networks, citation networks, and recommendation systems.[3–5] These graphs provide a powerful means to represent network-structured data, revealing complex relationships and interactions within the system. A fundamental and challenging problem in attributed graph analysis is clustering,[6,7] which facilitates a deeper understanding of the graph's structure and helps uncover cohesive topological groupings.[8,9] Effective clustering in attributed graphs requires balancing both structural cohesion and attributed similarity within clusters while simultaneously maximizing structural and attributed dissimilarity between clusters.[10,11]

In recent years, numerous methods have been proposed for deriving optimal partitions from attributed graphs. For example, the PHG algorithm[12] is a three-phase approach for influence maximization that leverages the attributed graph structure. The DFCN model[13] combines learnable parameters from FCN and DRN to achieve effective clustering. Additionally, the general Bayesian framework for attributed graph clustering (GBAGC)[14] defines a probabilistic model that naturally integrates both structural and attributed information. The FDAGC and FCAN-MOPSO algorithms[15,16] are capable of achieving the task in a purely unsupervised and end-to-end manner without additionally incorporating conventional clustering techniques, while other methods include density-based spatial clustering,[17] spectral clustering,[18] K-means,[19] and deep learning-based techniques,[20,21] among others. Specifically, Shang *et al.*[22] generalized the related crisp graphs and fuzzy graphs by applying graph theory to single-valued neutrosophic sets and explored a new type of graph structure known as single-valued neutrosophic graphs.

Despite this progress, many existing methods rely heavily on parameter estimation for structural proximity, often neglecting the intrinsic characteristics of attributed clusters.[23] Two primary issues persist in current clustering approaches. First, many methods treat attributed features as mere derivatives of topological structures, failing to adequately balance the contributions of structural and attributed information. Second, most methods overlook the issue of fuzzy clustering.

[†]Kaiyue Jiang, Liheng Xu, Shipei Lin and Li-Yang Zhou made equal contributions to this paper.
[‡]Corresponding author. E-mail: hjli@nankai.edu.cn
[§]Corresponding author. E-mail: gaoge@bsu.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

Overlapping graph partitioning is already a fundamental challenge in graph analysis, reflecting the inherent complexity and ambiguity present in real-world data. Existing techniques, however, struggle to scale to large or high-dimensional attributed graphs, and the limited number of parameters often hampers the efficiency of these methods.

To address these challenges, we propose a new framework, named Markov lumpability optimization (MLO), for efficient clustering on large-scale attributed graphs. Specifically, we define the lumped Markov chain on an attributed augmented graph and introduce a new metric, *Markov lumpability*, which quantifies the discrepancy between the original and lumped Markov transition probability matrices. To minimize Markov lumpability, we develop a conjugate gradient projection-based method that ensures the partition aligns with the intrinsic structure of the clusters through conditional optimization. Notably, the MLO framework reveals fuzzy cluster configurations in attributed graphs probabilistically. Through extensive experiments on both artificial and real datasets, we demonstrate that the MLO framework outperforms existing algorithms, effectively verifying its efficacy and reliability while achieving superior fuzzy clustering results.

2. The framework

2.1. Attributed augmented graph

We represent an attributed network as $G = (S, E, \Lambda)$, where S is the set of n nodes, E is the set of m edges, and $\Lambda = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ is the collection of attributed vectors associated with each node, characterizing their properties. Each node $x_i, x_j \in S$ is associated with an η-dimensional attributed vector, $\boldsymbol{x}_i = [x_{i1}, \ldots, x_{i\eta}]$ and $\boldsymbol{x}_j = [x_{j1}, \ldots, x_{j\eta}]$, where $|S| = n$, and the edges $w(i, j) \in E$ satisfy $w(i, j) = 1$ if nodes i and j are connected, and $w(i, j) = 0$ otherwise.

Utilizing random walk theory to uncover attributed cluster configurations is both efficient and challenging. The random walker is confined within densely connected clusters that exhibit high structural similarity. To apply random walk theory in attributed graph clustering, we define a new attributed augmented graph that incorporates Markov properties. In this graph, the edge weight between adjacent nodes i and j represents their attributed similarity, which is quantified by the Euclidean distance between their attributed vectors.

The similarity between nodes i and j is given by

$$\mathrm{sim}(i, j) = \mathrm{e}^{-\frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}, \tag{1}$$

where $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ denotes the Euclidean distance, defined as $\sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j) \cdot (\boldsymbol{x}_i - \boldsymbol{x}_j)}$. The similarity measure $\mathrm{sim}(i, j)$ ranges from zero to one, increasing with the similarity between the two attributed vectors. Compared with traditional linear or discrete similarity measures, Eq. (1) is a normalized quality function which lies between [0, 1], which is very suitable for computations involving high-dimensional data. Furthermore, since x_i is vector attributed data, we can extend it to a high-dimensional form. So, Eq. (1) is more flexible than one-hot attributes and is more suitable for complex situations.

Based on Markov theory, we define the transition probability $P(i, j)$ between two adjacent nodes using this attributed similarity. Specifically, for the edges $e(i, j)$, we have

$$P(i, j) = \frac{\mathrm{sim}(i, j)}{d(i)}, \quad d(i) = \sum_{h \in S} \mathrm{sim}(i, h), \tag{2}$$

where $d(i)$ represents the attributed degree of node i. Notably, as the similarity between two attributed vectors increases, so does the transition probability $P(i, j)$. Therefore, when considering a random walker on graph G, governed by P, the walker tends to move to nodes with more similar attributed values in the next step, following the structure of the graph.

2.2. The lumped Markov chain

For a given Markov chain on the attributed augmented graph with transition matrix P, the stationary distribution is defined as

$$\pi(i) = \frac{d(i)}{\sum_{h \in S} d(h)}, \tag{3}$$

where $d(i)$ represents the degree of attribution of of node i, and we assume that it satisfies the detailed balance condition

$$\pi(i)P(i, j) = \pi(j)P(j, i). \tag{4}$$

We now consider a partition of S in the form $S = \cup_{k=1}^{r} S_k$, where $S_k \cap S_l = \emptyset$ for $k \neq l$. Our objective is to aggregate the nodes within each cluster in order to derive the lumped Markov chain formulation. To achieve this, we first map each set S_k in the state space $\mathbb{S} = \{S_1, \ldots, S_r\}$ to the nodes of an r-node network $\hat{G}(\mathbb{S}, \mathbb{E})$. Here, $\mathbb{E} = \{w(S_k, S_l)\}_{S_k, S_l \in \mathbb{S}}$, where the weight $w(S_k, S_l)$ of the edge between S_k and S_l is defined as

$$w(S_k, S_l) = \sum_{i \in S_k, j \in S_l} \pi(i)P(i, j). \tag{5}$$

This sum encompasses all transition probabilities $P(i, j)$, where $i \in S_k$ and $j \in S_l$. Based on the detailed balance condition in Eq. (4), we can confirm that $w(S_k, S_l) = w(S_l, S_k)$.

Next, we define $\hat{\pi}_k = \sum_{z \in S_k} \pi(z)$ for $k = 1, \ldots, r$, which represents the lumped stationary distribution for the aggregated nodes in $\hat{G}(\mathbb{S}, \mathbb{E})$. The lumped Markov chain on $\hat{G}(\mathbb{S}, \mathbb{E})$ has a stationary distribution $\hat{\pi}$ and lumped transition probabilities given by

$$Q_{kl} = \frac{w(S_k, S_l)}{\sum_{b=1}^{r} w(S_k, S_b)} = \frac{1}{\hat{\pi}_k} \sum_{i \in S_k, j \in S_l} \pi(i)P(i, j). \tag{6}$$

It is evident that Q forms a transition matrix within the state space $\mathbb{S}$ and satisfies the detailed balance condition with respect to $\hat{\pi}$, i.e.,

$$\hat{\pi}_k \cdot Q_{kl} = \hat{\pi}_l \cdot Q_{lk}. \tag{7}$$

To obtain a more distinct probability expression, we introduce a probability function $\rho_k(i)$, which represents the probability that node i belongs to the kth cluster. We assume the following conditions for $\rho_k(i)$:

$$\rho_k(i) \geq 0 \quad \text{and} \quad \sum_{k=1}^{r} \rho_k(i) = 1 \quad \text{for all} \quad i \in S. \tag{8}$$

Thus, the transition matrix Q can be reformulated in terms of π as

$$\begin{aligned} Q_{kl} &= \frac{1}{\hat{\pi}_k} \sum_{i,j \in S} \pi(i)\rho_k(i)P(i,j)\rho_l(j) \\ &= \sum_{i,j \in S} \pi_k(i)P(i,j)\rho_l(j), \end{aligned} \tag{9}$$

where $\pi_k(i)$ denotes the probability distribution of node i within cluster S_k, and $\rho_l(j)$ represents the probability that node j belongs to cluster S_l.

2.3. Markov lumpability

To evaluate the quality of the partition, we define the difference between the original Markov transition probability matrix P and the lumped Markov transition probability matrix Q as a new metric, Markov lumpability. Its definition is as follows:

Definition 1 (Markov lumpability) Given the original Markov transition probability matrix P and the lumped Markov transition probability matrix Q, we define the difference $L = \|P - Q\|_\pi^2$ in the Hilbert–Schmidt norm[24] as Markov lumpability.

It is evident that Q should capture the intrinsic characteristics of P and that Markov lumpability L should be minimal for a reasonable network partition. However, minimizing L directly is challenging, as P and Q exist in different state spaces with different dimensions. To overcome this, we employ an alternative approach by deriving the lifted lumped Markov transition probability matrix $\tilde{Q}$, based on Q. Every Q can be lifted to the space of transition probability matrices in the original state space S via the following transformation:

$$\tilde{Q}(i,j) = \sum_{k,l=1}^{r} \rho_k(i)Q_{kl}\pi_l(j), \quad i,j \in S, \tag{10}$$

where

$$\pi_k(i) = \frac{\rho_k(i)\pi(i)}{\hat{\pi}_k}, \quad \hat{\pi}_k = \sum_{z \in S} \rho_k(z)\pi(z). \tag{11}$$

As shown in Fig. 1, $\tilde{Q}(i,j)$ represents the likelihood that two nodes i and j share at least one common community k. Since $\tilde{Q}$ contains the partition information and has the same size as P, we can naturally use it to reveal the optimal clusters.

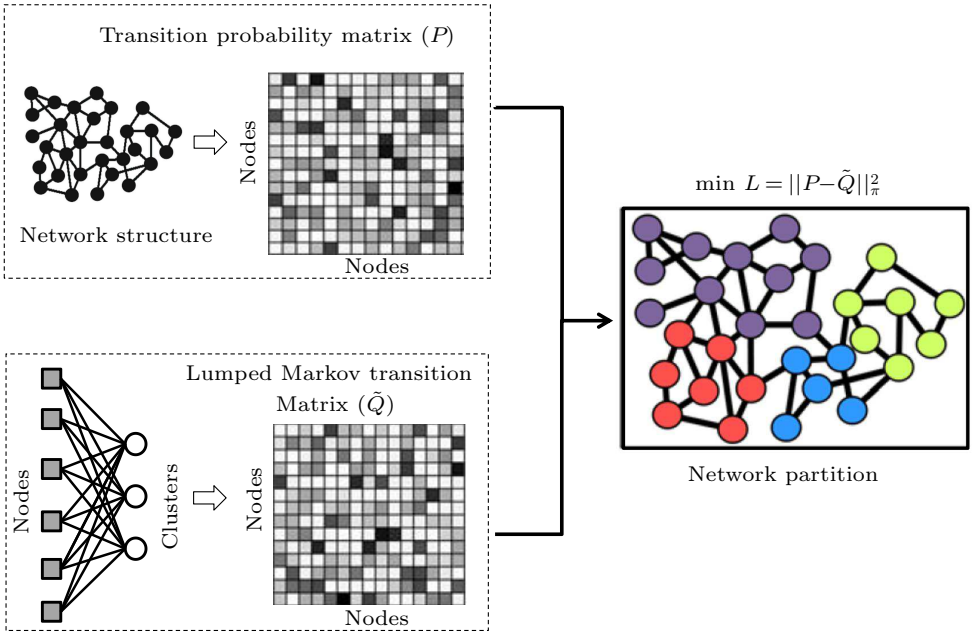


Fig. 1. Illustration of Markov lumpability optimization (MLO). Every component of the transition probability matrix P accounts for the likelihood that two nodes are connected to each other, and each element of the lumped Markov transition probability matrix $\tilde{Q}$ represents the probability that two nodes are in at least one common cluster. We define the difference $L = \|P - Q\|_\pi^2$ in the Hilbert–Schmidt norm as the Markov lumpability.

Utilizing $\tilde{Q}$ from Eq. (10), we can provide a detailed formulation for Markov lumpability L as follows:

$$L = \|P - \tilde{Q}\|_\pi^2 = \sum_{i,j \in S} \frac{\pi(i)}{\pi(j)} \left|P(i,j) - \tilde{Q}(i,j)\right|^2 = \sum_{i,j \in S} \pi(i)\pi(j) \left(\sum_{k,l=1}^{r} \rho_k(i)\rho_l(j)\frac{Q_{kl}}{\hat{\pi}_l} - \frac{P(i,j)}{\pi(j)} \right)^2, \tag{12}$$

subject to the constraint in Eq. (8).

By minimizing Eq. (13), the optimal partition and the corresponding $\tilde{Q}$ can be obtained by

$$\min_{\rho_k(i),Q} L = \min_{\rho_k(i),Q} \|P-\tilde{Q}\|_\pi^2. \tag{13}$$

Furthermore, we define Q^*_{kl}, which aims to minimize L in Eq. (12):

$$\begin{aligned} Q^*_{kl} &= \sum_{i,j\in S} \pi_k(i)P(i,j)\rho_l(j) \\ &= \frac{1}{\hat{\pi}_k}\sum_{i,j\in S}\pi(i)\rho_k(i)P(i,j)\rho_l(j), \end{aligned} \tag{14}$$

which is similar to Q_{kl} in hard clustering. Since $\sum_{z\in S}\pi_k(z)=1$ for all k, Q^*_{kl} is a valid transition probability matrix that satisfies the detailed balance condition with respect to $\hat{\pi}$:

$$\hat{\pi}_k\cdot\hat{P}^*_{kl} = \hat{\pi}_l\cdot\hat{P}^*_{lk}. \tag{15}$$

To find the optimal solution for L, we derive the following fundamental theorem.

Theorem 1. The expressions for $\partial L/\partial Q$ and $\partial L/\partial\rho$ with respect to Eq. (12) are given by

$$\frac{\partial L}{\partial Q} = 2\big(\hat{\pi}QI_{\hat{\pi}}^{-1}\hat{\pi}I_{\hat{\pi}}^{-1} - I_{\hat{\pi}}Q^*I_{\hat{\pi}}^{-1}\big), \tag{16a}$$

$$\begin{aligned} \frac{\partial L}{\partial\rho} = 2\big[&(QI_{\hat{\pi}}^{-1}\hat{\pi}I_{\hat{\pi}}^{-1}Q^{\mathrm{T}} + I_{\hat{\pi}}^{-1}Q^{\mathrm{T}}\hat{\pi}QI_{\hat{\pi}}^{-1}) \\ &\times\rho I_\pi - \big(QI_{\hat{\pi}}^{-1}+I_{\hat{\pi}}^{-1}Q^{\mathrm{T}}\big)\cdot\rho P^{\mathrm{T}}I_\pi\big], \end{aligned} \tag{16b}$$

where $\rho=(\rho_k(i))_{k=1,\dots,r,i\in S}$ is a $r\times n$ matrix, and $\hat{\pi}$ is an $r\times r$ matrix with entries

$$\hat{\pi}_{kl} = \sum_{z\in S}\pi(z)\rho_k(z)\rho_l(z) = (\rho\cdot I_\pi\cdot\rho^{\mathrm{T}})_{kl}. \tag{17}$$

The diagonal matrices I_π and $I_{\hat{\pi}}$ are $n\times n$ and $r\times r$ matrices, respectively, with entries

$$I_\pi(i,j) = \pi(i)\delta(i,j), \quad i,j\in S, \tag{18a}$$

$$(I_{\hat{\pi}})_{kl} = \hat{\pi}_k\delta_{kl}, \quad k,l=1,\dots,r, \tag{18b}$$

where $\delta(i,j)$ and δ_{kl} are Kronecker delta symbols.

Proof First, we calculate the variation of L in Eq. (12) with respect to Q_{kl}:

$$\begin{aligned} \frac{\partial L}{\partial Q_{kl}} &= 2\sum_{i,j\in S}\pi(i)\pi(j)\bigg(\sum_{m,n=1}^{r}\rho_m(i)\rho_n(j)\frac{Q_{mn}}{\hat{\pi}_n} - \frac{P(i,j)}{\pi(j)}\bigg) \\ &\quad\times\sum_{s,t=1}^{r}\rho_s(i)\rho_t(j)\frac{1}{\hat{\pi}_t}\delta_{sk}\delta_{tl} \\ &= 2\bigg(\frac{1}{\hat{\pi}_l}\sum_{i,j\in S}\sum_{m,n=1}^{r}\pi(i)\pi(j)\rho_k(i)\rho_l(j)\rho_m(i)\rho_n(j)\frac{Q_{mn}}{\hat{\pi}_n} \\ &\quad - \frac{1}{\hat{\pi}_l}\sum_{i,j\in S}\pi(i)P(i,j)\rho_k(i)\rho_l(j)\bigg) \\ &= 2\bigg(\frac{1}{\hat{\pi}_l}\sum_{m,n=1}^{r}\hat{\pi}_{km}\frac{Q_{mn}}{\hat{\pi}_n}\hat{\pi}_{nl} - \frac{\hat{\pi}_k}{\hat{\pi}_l}Q^*_{kl}\bigg). \end{aligned} \tag{19}$$

The expression for $\partial L/\partial Q$ in Eq. (16a) is derived from the matrix form of the above result. By varying L with respect to $\rho_r(z)$, we obtain the following:

$$\begin{aligned} \frac{\partial L}{\partial\rho_r(z)} &= 2\sum_{i,j\in S}\pi(i)\pi(j)\bigg(\sum_{m,n=1}^{r}\rho_m(i)\rho_n(j)\frac{Q_{mn}}{\hat{\pi}_n} - \frac{P(i,j)}{\pi(j)}\bigg) \\ &\quad\times\sum_{k,l=1}^{r}\bigg[\delta_{kr}\delta(i,z)\rho_l(j)\frac{Q_{kl}}{\hat{\pi}_l} + \delta_{lr}\delta(j,z)\rho_k(i)\frac{Q_{kl}}{\hat{\pi}_l} \\ &\quad - \rho_k(i)\rho_l(j)\frac{Q_{kl}}{\hat{\pi}_l^2}\sum_{v\in S}\delta_{lr}\delta(v,z)\pi(v)\bigg]. \end{aligned} \tag{20}$$

As expected, due to the detailed balance condition in Eq. (4) and the definition of Q^* in Eq. (14), we obtain the following equation:

$$\begin{aligned} \frac{\partial L}{\partial\rho_c(z)} = 2\bigg[&\sum_{y\in S}\pi(z)\pi(y)\bigg(\sum_{m,n=1}^{r}\rho_m(z)\rho_n(j)\frac{Q_{mn}}{\hat{\pi}_n} - \frac{P(z,j)}{\pi(j)}\bigg) \\ &\times\sum_{k=1}^{r}\rho_l(j)\frac{Q_{cl}}{\hat{\pi}_l} \\ &+\sum_{x\in S}\pi(z)\pi(i)\bigg(\sum_{m,n=1}^{r}\rho_m(i)\rho_n(z)\frac{Q_{mn}}{\hat{\pi}_n} - \frac{P(i,z)}{\pi(z)}\bigg) \\ &\times\sum_{k=1}^{r}\rho_k(i)\frac{Q_{kc}}{\hat{\pi}_c} \\ &-\sum_{i,j\in S}\pi(i)\pi(j)\bigg(\sum_{m,n=1}^{r}\rho_m(i)\rho_n(j)\frac{Q_{mn}}{\hat{\pi}_n} - \frac{P(i,j)}{\pi(j)}\bigg) \\ &\times\sum_{k=1}^{r}\pi(z)\rho_k(i)\rho_c(j)\frac{Q_{kc}}{\hat{\pi}_c^2}\bigg]. \end{aligned} \tag{21}$$

Eventually, Eq. (16b) is obtained through a series of appropriate manipulations, completing the proof.

In Theorem 1, we present the partial derivatives of the objective function L with respect to Q and ρ (Eqs. (16a) and (16b)), which are essential for designing gradient-based optimization methods. The optimal Q_{kl} for any fixed partition $\{S_1,\dots,S_r\}$ can be found using this formulation. Once the partition $\{S_1,\dots,S_r\}$ is known, the minimizer of Eq. (13) is determined by Eq. (9). Moreover, the corresponding $\tilde{Q}$ is the stochastic matrix defined in Eq. (10), offering the optimal rank-r approximation of the original matrix with respect to the metric in Eq. (13).

3. The Markov lumpability optimization algorithm

Based on the results of Theorem 1, we propose an efficient method for uncovering optimal partitions using MLO. To address the constrained optimization problem in Eq. (13), MLO utilizes a variant of the conventional conjugate gradient method, commonly applied in machine learning for minimizing objective functions. The key idea behind this optimization is the steepest descent procedure with a nonzero momentum

term, leading to a reduced conjugate gradient framework with projections.[25,26]

Before introducing the MLO algorithm, we define a projection operator $\mathcal{P}$. The operator $\mathcal{P}$ maps a real vector onto a probability vector. Suppose $\boldsymbol{v} = (v_1, v_2, \ldots, v_N) \in \mathbb{R}^N$, and let $v_i < 0$ for $i \in \Lambda$. When $i \in \Lambda$, we set $\mathcal{P}v_i = 0$; otherwise, we set $\mathcal{P}v_i = v_i / \sum_{j \notin \Lambda} v_j$. The detailed steps of the MLO algorithm are given in Algorithm 1.

Algorithm 1 Markov lumpability optimization (MLO)

Input: The initial state $\rho^{(0)}$ as the indicator matrix with $n = 0$, learning rates $\alpha, \beta > 0$, and the tolerance TOL .

Output: The optimal partition $\rho^{(n)}$.

Repeat:

1. $I_\pi(i,j) = \pi(i)\delta(i,j)$
2. $\hat{\pi}_{kl} = (\rho \cdot I_\pi \cdot \rho^{\mathrm{T}})_{kl}$
3. $\frac{\partial L}{\partial Q} = 2\left(\hat{\pi} Q I_{\hat{\pi}}^{-1} \hat{\pi} I_{\hat{\pi}}^{-1} - I_{\hat{\pi}} Q^* I_{\hat{\pi}}^{-1}\right)$
4. $\frac{\partial L}{\partial \rho} = 2\left[\left(Q I_{\hat{\pi}}^{-1} \hat{\pi} I_{\hat{\pi}}^{-1} Q^{\mathrm{T}} + I_{\hat{\pi}}^{-1} Q^{\mathrm{T}} \hat{\pi} Q I_{\hat{\pi}}^{-1}\right) \cdot \rho I_\pi - (Q I_{\hat{\pi}}^{-1} + I_{\hat{\pi}}^{-1} Q^{\mathrm{T}}) \cdot \rho P^{\mathrm{T}} I_\pi - \left((I_{\hat{\pi}}^{-2} Q^{\mathrm{T}}) * \left(\hat{\pi} I_{\hat{\pi}}^{-1} Q^{\mathrm{T}} \hat{\pi}\right)\right) \cdot \mathbf{1}_{r\times 1} \pi^{\mathrm{T}} + \left((I_{\hat{\pi}}^{-2} Q^{\mathrm{T}}) * (Q^*)^{\mathrm{T}}\right) \cdot \hat{\pi} \pi^{\mathrm{T}}\right]$
5. $Q^{(n+1)} = \mathcal{P}\left(Q^{(n)} - \alpha \frac{\partial L}{\partial Q}(Q^{(n)}, \rho^{(n)}) + \beta\left(Q^{(n)} - Q^{(n-1)}\right)\right)$
6. $\rho^{(n+1)} = \mathcal{P}\left(\rho^{(n)} - \alpha \frac{\partial L}{\partial \rho}(Q^{(n)}, \rho^{(n)}) + \beta\left(\rho^{(n)} - \rho^{(n-1)}\right)\right)$

Until: $\|\rho^{(n+1)} - \rho^{(n)}\|$ converges under the given TOL.

The quantification of the iteration steps is nontrivial, as it depends on factors such as the inherent structure of the network and the selection of initial values. In networks with well-defined clusters, convergence typically occurs quickly, whereas for diffusive networks, the convergence process may be slower.

Next, we estimate the computational complexity per iteration. In the computation of matrix P, the complexity is $O(m)$. In the iteration step related to Q, all matrices are of size $r \times r$ and full rank, as indicated by Eq. (16a). The computations of $\hat{\pi}$ and $\hat{\hat{\pi}}$ cost $O(rn)$ and $O(r^2 n)$, respectively. Since P is sparse with m entries, the cost for computing Q^* is $O(r^2 m)$, where m is the number of edges, typically assumed to be $O(n)$ in practical networks. Therefore, the cost for the Q-related step is $O(r^2(m+n))$. The cost for the ρ-related step is also $O(r^2(m+n))$, as per Eq. (16b), due to the involvement of Q^* in the equations.

Finally, given the significant applications of modularity optimization as a matrix-based partitioning method in graph clustering and the fact that MLO also belongs to the category of matrix-based optimization approaches, it is natural to compare the two frameworks to highlight the advantages of MLO. Both methods share fundamental similarities: they optimize community partitions through quantitative metrics and rely on matrix analysis (Markov chains vs. adjacency matrices) to reveal network structures. However, MLO distinguishes itself by integrating structural and attribute information through an attributed augmented graph, defining a lumped Markov chain, and introducing the Markov lumpability metric. While preserving the significance of community structure emphasized by modularity optimization, MLO naturally supports fuzzy clustering via the aggregation of transition probability matrices, effectively balancing the contributions of multisource data.

4. Experimental details

In this section, we present comprehensive experimental evaluations conducted across various multicategory datasets to assess the performance of the proposed method. All experiments were executed on a system equipped with dual socket P+ (LGA-4189) 3rd Gen Intel Scalable Processors, an Intel C621A chipset, and up to 8 TB of Intel Optane persistent memory.

4.1. Evaluation methods and metrics

To demonstrate the superiority of the MLO method, we selected five representative clustering algorithms for comparison. Specifically, K-means[19] is a widely used clustering algorithm known for its efficiency in solving clustering tasks, including fuzzy clustering. PHG[12] is an influence maximization model based on the linear threshold model, incorporating three phases: partition, heuristic, and greedy. DFCN[13] is an approach that automatically learns high-order explicit and implicit feature interactions. GBAGC[14] is a new Bayesian framework specifically designed for attributed graph clustering, and FDAGC and FCAN-MOPSO[15,16] are established fuzzy clustering algorithms that optimize node memberships based on dense structures and content relevance.

The clustering performance of these methods is evaluated using three commonly used metrics: accuracy (ACC), normalized mutual information (NMI), and adjusted Rand index (ARI). The ACC metric evaluates the ratio of correctly predicted instances, defined as

$$\mathrm{ACC}(r,s) = \frac{\sum_{i=1}^{N} \delta(r_i, \mathrm{map}(s_i))}{n}, \tag{22}$$

where r_i represents the true class label of the ith sample, s_i represents the predicted class label of the ith sample, and $\mathrm{map}(\cdot)$ is a mapping function used to match predicted labels with true labels. The function $\delta(x,y)$ is a binary classification function

that outputs one if $x = y$ and zero otherwise. Higher ACC values indicate better clustering performance.

Before introducing NMI, we first define mutual information (MI), which measures the amount of information shared between two variables. The MI between two variables X and Y is defined as

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{23}$$

where $H(X)$ represents the entropy of X, and $H(X,Y)$ is the joint entropy of X and Y. The NMI is then computed as

$$\mathrm{NMI}(r,s) = \frac{2I(r;s)}{H(r)+H(s)}, \tag{24}$$

where $H(r)$ and $H(s)$ are the entropies of the true and predicted class labels, respectively. A higher NMI value indicates better alignment between the true and predicted labels.

The adjusted Rand index (ARI) improves upon the traditional Rand index (RI) by adjusting for chance. The ARI is defined as

$$\mathrm{ARI} = \frac{RI - E[RI]}{\max(RI) - E[RI]}, \tag{25}$$

where $RI = \frac{a+d}{a+b+c+d}$. In this equation, a is the number of sample pairs in the same class, b is the number of pairs that are in the same class in one partition but different in another, c is the number of pairs that are in different classes in one partition but the same in the other, and d is the number of pairs in different classes in both partitions. A higher ARI indicates better clustering quality.

4.2. Evaluation of synthetic networks

In practice, we evaluate the performance of the proposed MLO method using the well-known LFR benchmark,[31,32] which generates synthetic graphs for community detection tasks. The LFR benchmark allows control over network structure and attributes using the mixing parameter μ, which ranges from 0.1 to 0.8. The parameter μ determines the balance between intra- and intercommunity connections. A low μ value results in a clear community structure with many intracommunity connections and fewer intercommunity connections.

We construct different network instances using the following parameters: N (number of nodes), k (average degree), max-k (maximum degree), min-c and max-c (minimum and maximum community sizes), n_{attr} (number of attributes), a_{inf} (attribute domain cluster assignment), and R (attributed range). For our experiments, we set $K = 5000$, $k = 150$, max-$k = 200$, min-$c = 50$, max-$c = 250$, $n_{\mathrm{attr}} = 5$, and $\mu \in [0.1, 0.9]$. The results are shown in Fig. 2, where we observe that when $\mu \in [0.1, 0.4]$, many methods, including FDAGC, FCAN-MOPSO, and MLO, achieve high performance, with NMI values and ARI values close to one. It is worth noting that MLO obtains the highest value among the three mentioned algorithms with a slight advantage. As μ increases, the NMI and ARI values of all methods decline, with performance dropping below NMI = 0.8 and ARI = 0.6 for $\mu \in [0.7, 0.9]$. Despite this general decline, MLO consistently outperforms other methods across most values of μ. This demonstrates the superiority of MLO, as it consistently provides the highest ARI values across different μ values.

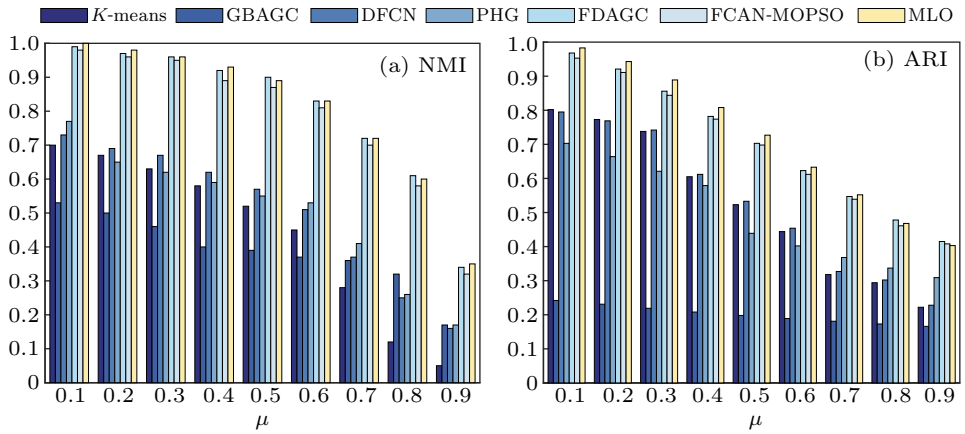


Fig. 2. Experimental results for the LFR benchmark: (a) method performances evaluated in terms of NMI on synthetic networks; (b) method performances evaluated in terms of ARI on synthetic networks.

4.3. Evaluation of real-world networks

We evaluate the performance of MLO and the comparison methods on eight real-world networks that contain high-dimensional attributes. These datasets include Cora, Citeseer, Facebook, Photo, ACM, Eat, Uat, and Amap. The Cora dataset[33] is a widely used benchmark in machine learning, containing citation data from academic papers. The Citeseer dataset[34] consists of 3312 scientific publications. The Facebook dataset[35] contains citations and abstracts from biomedical literature. The Photo and Amap datasets[36,37] come from Amazon product reviews. The ACM dataset[38] contains papers published in KDD, SIGMOD, and other conferences. The Eat and Uat datasets[39,40] are derived from transporta-

tion statistics. The statistics of these datasets are presented in Table 1.

Table 1. Statistics of real-world data sets.

Data set	Nodes	Edges	Features	Categories
Cora	2708	10556	1433	7
Citeseer	3327	9104	3703	6
Facebook	4089	170714	1406	193
Photo	7650	238162	745	8
ACM	3025	29281	1870	3
Eat	399	5994	203	4
Uat	1190	135999	239	4
Amap	7650	119081	745	8

We evaluate the various methods using the following metrics: ACC, NMI, and ARI. Figures 3(a)–3(c) compare the performance of MLO against those of other algorithms on these real-world networks. In Fig. 3(a), MLO outperforms all other methods in terms of NMI on the Cora, Citeseer, Facebook, Photo, ACM, and Amap datasets. Figures 3(b) and 3(c) show that MLO also performs the best in terms of ARI and ACC across most real-world networks, with only FDAGC outperforming it on the ACM dataset in terms of ACC. The experimental data indicate that the MLO algorithm achieves a combined improvement of approximately 15% compared to other algorithms. Specifically, it demonstrates an average improvement of about 14.3% in terms of the ACC metric, 10.5% in the ARI metric, and 20.2% in the NMI metric, calculated using the arithmetic mean across these metrics.

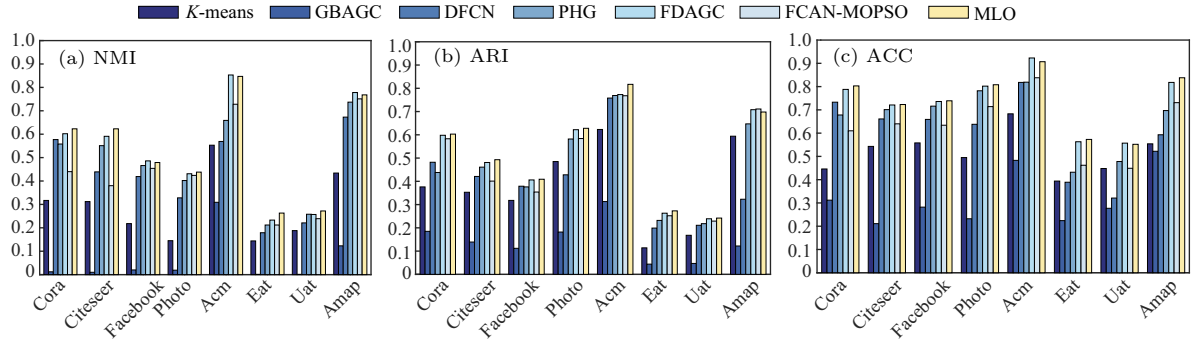


Fig. 3. The result of the eight real networks in terms of six different algorithms, evaluated using NMI, ARI, and ACC. This figure group includes (a) the value of NMI, (b) the value of ARI and (c) the value of ACC.

From these results, it is evident that MLO provides superior performance compared to other methods, particularly in networks with high-dimensional attributes. Notably, the performance of all methods on the ACM dataset is relatively high, which can be attributed to two factors: the dense structural and attribute information in ACM, and the relatively small number of real clusters, which reduces the impact of misclassifications. The real-world networks also perform well with the Photo and Amap datasets due to their clear data label structures and low noise levels. On the other hand, the *K*-means algorithm struggles to process datasets with undefined structures. Finally, the clustering results for the Cora dataset are generally worse due to its unsuitable label settings for the clustering task.

Furthermore, we tested the speed of the proposed method and Fig. 4 compares the running times of the five clustering algorithms across the eight datasets. It is evident that the *K*-means algorithm is the fastest method on all datasets, as expected, since *K*-means is the simplest clustering method and thus exhibits the lowest computational cost. When *K*-means is excluded from the comparison, MLO emerges as the most efficient method among the remaining algorithms. This confirms that, despite its complexity, MLO achieves superior performance in terms of both clustering accuracy and computational efficiency compared to the other methods tested. The running time of MLO hardly changes when processing different datasets, while the running time of GBAGC is reduced significantly (by about 60%) when processing the ARI dataset compared to the NML dataset. We also note that DFCN takes the longest time to process every dataset among all the algorithms, demonstrating its high computational cost. As indicated by the experimental data, the MLO algorithm achieves an average percentage improvement of approximately 19.1% in running time compared to other algorithms, demonstrating its significant advantages in time efficiency.

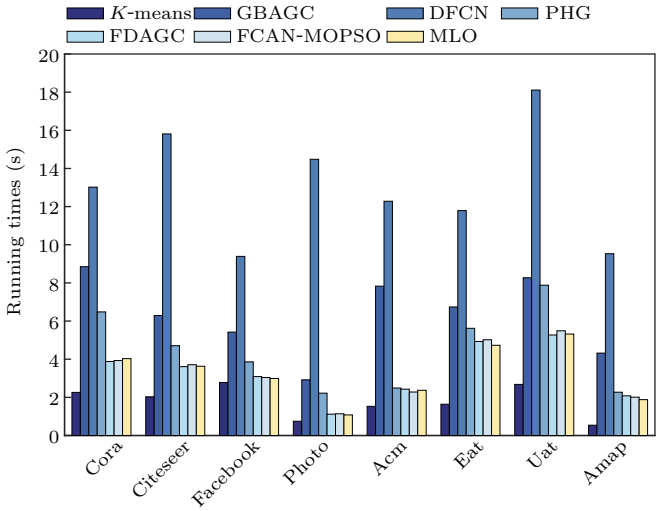


Fig. 4. Comparison of running times among the six methods.

4.4. Evaluation of large-scale network data sets

In order to further evaluate the effectiveness of the MLO algorithm, we conducted an experiment on three large datasets, including PubMed, Gplus, and Twitter. These three networks all have more than 10000 nodes. In addition to the statistics of these networks presented in Table 2, we chose NMI and ACC as evaluation metrics. To further demonstrate the performance optimization advantages of MLO over other algorithms of the same type, we selected GBAGC, PHG, GBAGC, and FCAN-MOPSO as the comparative algorithms, and the results are shown in Table 3. When observing the algorithmic performance in Table 3, it can be seen that MLO exhibits better performance than those of other algorithms. On average, MLO achieves a performance improvement of approximately 10.7% compared to other algorithms. These results are also consistent with the results presented in the previous sections, which shows that MLO also has good performance in complex networks.

Table 2. Statistics of large-scale network data sets.

Data set	Nodes	Edges	Features	Categories
PubMed	19717	88648	500	3
Twitter	125120	2248406	33569	3140
Gplus	107614	13673453	13634	468

Table 3. Results of large-scale network testing in terms of NMI and accuracy. Here, the bold numbers denote the best results across all different methods.

	PubMed		Gplus		Twitter	
	NMI	Accuracy	NMI	Accuracy	NMI	Accuracy
FCAN-MOPSO	0.55	0.49	0.30	0.66	0.43	0.69
FDAGC	0.603	**0.682**	0.303	0.744	0.452	0.761
GBAGC	0.13	0.36	0.07	0.6	0.09	0.59
MLO	**0.62**	0.651	**0.312**	**0.76**	**0.47**	**0.774**

4.5. Visualization of overlapping communities

To validate the effectiveness of fuzzy clustering on attributed graphs, we performed a visualization of the final partitions. Given the unique characteristics of each dataset, we selected four widely used benchmark datasets (http://www-personal.umich.edu/mejn/netdata),[41] namely, Dolphin, Karate, Football, and Polbooks, for visual analysis. Figure 5 presents the clustering results, providing a visual representation of the discovered communities. The overlapping regions between communities are highlighted with black dashed lines, and the detected overlapping communities are color-coded for clarity.

For example, in the Dolphin network (Fig. 5(a)), nodes 8, 20, 29, 31, 37, and 40 are shared by two distinct groups. Similarly, in the Karate network (Fig. 5(b)), nodes 3 and 9 belong to both groups identified by our algorithm. These results align with those obtained using classical methods, such as the Newman method. The visualizations confirm that the proposed MLO algorithm is effective in detecting overlapping communities in attributed graphs.

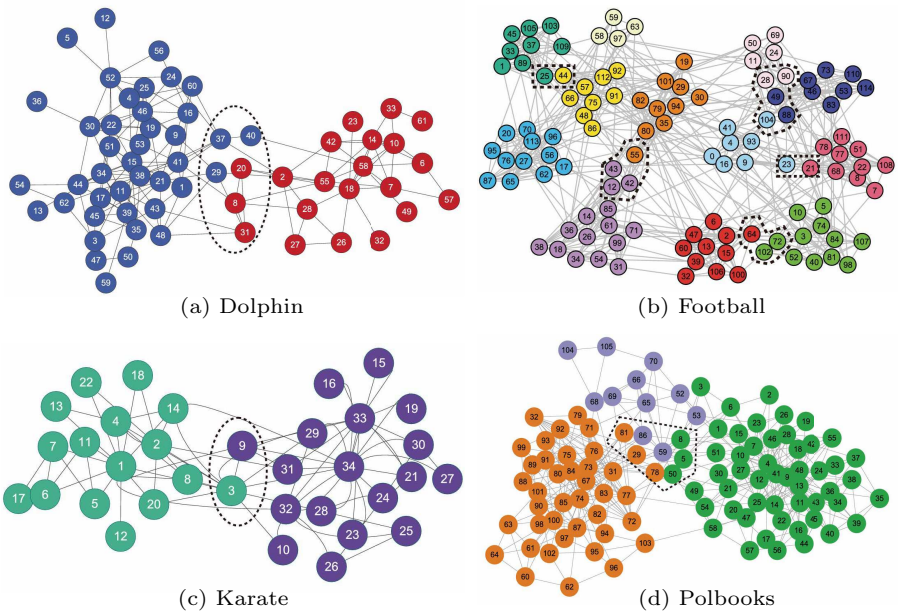


Fig. 5. Visualization of experimental results obtained using four real-world networks. These networks include (a) the Dolphin network, (b) the Football network, (c) the Karate network, and (d) the Polbooks network. The overlapping parts are highlighted by dashed lines.

5. Conclusions

Existing methods for attributed graph clustering often rely heavily on parameter estimation for structural proximity, which fails to balance the contributions of structural and attributed information. Furthermore, few approaches address the problem of fuzzy attributed clustering, where communities may overlap. To overcome these limitations, we have designed the MLO framework for the efficient clustering of large-scale attributed graphs. Specifically, we define a lumped Markov chain on an attributed augmented graph and introduce the new metric of Markov lumpability to measure the discrep-

ancy between the original and lumped Markov transition probability matrices. To minimize this metric, we propose a conjugate gradient projection-based method that aligns the partition with the intrinsic structure of the clusters through conditional optimization. We conducted extensive experiments on both synthetic and real-world datasets to evaluate the performance of MLO. The results demonstrate that MLO outperforms five state-of-the-art attributed graph clustering methods across three evaluation metrics. Furthermore, the run-time analysis confirms that MLO is more efficient than most competing algorithms. These findings highlight the effectiveness and efficiency of MLO for attributed graph clustering tasks.

This paper provides a solid foundation for future research into efficient clustering methods for large-scale attributed graphs. The proposed MLO framework, with its ability to balance structural and attributed information while addressing fuzzy clustering, opens new avenues for enhancing clustering techniques on complex, high-dimensional graphs. The results of this study demonstrate that MLO outperforms existing methods across several evaluation metrics, making it a promising approach for handling large, dynamic, and high-order graphs in real-world applications. In our future work, we will build on this framework by extending its scalability to high-order networks, incorporating dynamic and temporal aspects of graph data, and integrating uncertainty quantification into complex network analysis, recommendation systems, and biological network modeling.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 72571150) and Beijing Natural Science Foundation (Grant No. 9182015).

References

[1] Xiao C, Xu X, Lei Y, Zhang K, Liu S and Zhou F 2023 *IEEE Trans. Knowl. Data Eng.* **35** 10540
[2] Qin M, Zhang C, Bai B, Zhang G and Yeung D Y 2023 *ACM Trans. Knowl. Discov. Data* **17** 1
[3] Qiu R, Yin H, Huang Z and Chen T 2020 *Proc. 43rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.* 669
[4] Kargar M, Zihayat M and Szlichta J 2019 *Proc. 29th Annu. Int. Conf-Comput. Sci. Sofw. Eng.* **19** 397
[5] Albert R and Barabasi A L 2002 *Rev. Mod. Phys.* **74** 47
[6] Yang Y, Su X, Zhao B, Li G, Hu P, Zhang J and Hu L 2024 *IEEE Trans. Fuzzy Syst.* **32** 1951
[7] Li H J, Feng Y, Xia C and Cao J 2024 *ACM Trans. Knowl. Discov. Data* **18** 1
[8] Li H J, Cao H, Feng Y, Li X and Pei J 2024 *IEEE Trans. Knowl. Data Eng.* **36** 11
[9] He C, Liu S, Zhang L and Zheng J 2019 *J. Ambient Intell. Humaniz. Comput.* **10** 3399
[10] Xie X X, Huo L A, Dong Y F and Cheng Y Y 2024 *Chin. Phys. B* **33** 038704
[11] Hu X, Chen J X and Cheng Y 2024 *Chin. Phys. B* **33** 100202
[12] Qiu L, Jia W, Yu J, Fan X and Gao W 2019 *IEEE Access* **7** 62511
[13] Tu W, Liu X, Zhou S, Li M, Zhu E, Liu L, Zhang C and Yin J 2021 *Proc AAAI Conf. Artif. Intell.* **35** 9978
[14] Yang Y, Su X, Zhao B, Li G, Peng W, Zhang J and Hu L 2023 *IEEE Trans. Fuzzy Syst.* **32** 4
[15] Pons P and Latapy M 2006 *J. Graph Algorithms Appl.* **10** 191
[16] Hu L, Yang Y, Tang Z, He Y and Luo X 2023 *IEEE Trans. Fuzzy Syst.* **31** 10
[17] Ahmed M, Seraj R and Islam S M S 2020 *Electronics* **9** 1295
[18] Grohe M and Schweitzer P 2020 *Commun. ACM* **63** 128
[19] Ikotun A M, Ezugwu A E, Abualigah L, Abuhaija B and Heming J 2023 *Inf. Sci.* **622** 178
[20] Talaei Khoei T, Ould Slimane H and Kaabouch N 2023 *Neural Comput. Appl.* **35** 23103
[21] Mohammed A and Kora R 2023 *J. King Saud Univ. - Comput. Inf. Sci.* **35** 757
[22] Anwar S, Azeem M, Jamil M K, Almohsen B and Shang Y 2024 *J. Supercomput.* **80** 19976
[23] Huo L A and Yu Y 2023 *Chin. Phys. B* **32** 108703
[24] Gretton A, Bousquet O, Smola A and Scholkopf B 2005 *Int. Conf. Algorithmic Learn. Theory* 63
[25] Nazareth J L 2009 *Wiley Interdiscip. Rev.: Comput. Stat.* **80** 19976
[26] Hashem I, Alaba F, Jumare M, Ibrahim A and Abulfaraj A 2024 *IEEE Access* **12** 33757
[27] E W, Li T and Vanden-Eijnden E 2008 *Proc. Natl. Acad. Sci. USA* **105** 7907
[28] Zhou S, Liu X, Li M, Zhu E, Liu L, Zhang C and Yin J 2019 *IEEE Trans. Neural Netw. Learn. Syst.* **31** 1351
[29] Wang S, Liu X, Liu L, Zhou S and Zhu E 2023 *IEEE Trans. Neural Netw. Learn. Syst.* **34** 4359
[30] Wang S, Liu X, Zhu X, Zhang P, Zhang Y, Gao F and Zhu E 2021 *IEEE Trans. Image Process.* **31** 556
[31] Berahmand K, Mohammadi M, Saberi-Movahed F, Li Y and Xu Y 2023 *IEEE Trans. Netw. Sci. Eng.* **10** 372
[32] Rohan H R K and Sahadath M H 2023 *Prog. Nucl. Energy* **164** 104849
[33] Cui G, Zhou J, Yang C and Liu Z 2020 *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. & Data Mining* 976
[34] Li Y, Wang P, Li Z, Yu J X and Li J 2024 *Proc. 30th ACM SIGKDD Conf. Knowl. Discov. Data Min.* 1725
[35] Temsah M H, Altamimi I, Jamal A, Alhasan K and Al-Eyadhy 2023 *Cureus* **15** 9
[36] Peng Z, Liu H, Jia Y and Hou J 2022 *IEEE Trans. Circuits Syst. Video Technol.* **33** 3296
[37] Yang X, Liu Y, Zhou S, Wang S, Tu W, Zheng Q, Liu X, Fang L and Zhu E 2023 *Proc. AAAI Conf. Artif. Intell.* **37** 10834
[38] Cunha W, Viegas F, França C, Rosa T, Rocha L and Gonçalves M A 2023 *ACM Comput. Surv.* **55** 1
[39] Liu Y, Tu W, Zhou S, Liu X, Song L, Yang X and Zhu E 2022 *AAAI Conf. Artif. Intell.* 2022
[40] Wu X, Hu J, Zhang A, Quan Y, Miao Q and Sun P G 2024 arXiv:2408.09790
[41] Li H J, Song S, Tan W, Huang Z, Li X, Xu W and Cao J 2022 *Neurocomputing* **512** 482