Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

PHYSICA A
STATISTICAL MECHANICS AND ITS APPLICATIONS

Visualizing complex networks by leveraging community structures

Zhenhua Huang [a,b], Junxian Wu [a], Wentao Zhu [b], Zhenyu Wang [a,*], Sharad Mehrotra [b,*], Yangyang Zhao [a]

[a] *School of Software Engineering, South China University of Technology, Guangzhou 510006, China*
[b] *School of Information and Computer Science, University of California Irvine, Irvine 92617, USA*

ARTICLE INFO

Article history:
Received 19 March 2019
Received in revised form 25 September 2020
Available online 18 November 2020

Keywords:
Network visualization
Community detection
Modularity
Graph compression
Visualization metric
Force directed

ABSTRACT

Layout algorithms provide an intuitive way of visualizing and understanding complex networks. Complex networks such as social networks, coauthorship networks, and protein interaction networks often display community structures. Existing network visualization methods that are mostly based on force-directed algorithms do not fully exploit community structures, leading to layouts with intertwined nodes/edges or "hairball" issues, especially when the size and complexity of networks increase. This paper generalizes the force-directed framework and proposes a new method for network visualization exploiting community structures. The approach, entitled GRA (**G**eneralized **R**epulsive and **A**ttractive algorithm), first discovers communities using community detection mechanisms and then computes weighted repulsive and attractive forces between intra- and inter-community nodes. GRA simulates the nodes in a network as particles and moves them based on repulsive and attractive forces until convergence. The method is also extended to visualize larger-scale graphs by using detected communities to compress the original graph. To quantify the effectiveness of network visualization, an area estimation method based on a multivariate Gaussian distribution with noise tolerance is introduced. A layout with a high metric prevents the visualization from entanglement while making as much full use of the canvas space as possible. Case studies on complex networks of various types and sizes demonstrate that GRA achieves state-of-the-art performance and facilitates complex network analysis.

1. Introduction

Many systems can be represented as complex networks, such as social networks, biological networks, and citation networks, in which the nodes are entities or objects and the edges are relationships between nodes [1–3]. To understand the structure of complex networks, layout algorithms are proposed to displace nodes of a network in proper positions in a canvas. A visualization algorithm enables us to have an intuitive understanding of the graph structure in complex networks. The existing classical visualization methods are mostly based on the force-directed algorithm, such as the Kamada–Kawai spring model (KK) [4], and Fruchterman–Reingold spring model (FR) [5]. These algorithms work well in small and sparse graphs. However, they cannot visualize the structure of networks clearly and may suffer from issues of entanglement or "hairballs" [6], in some networks as illustrated in Fig. 1(a) and (b).

[*] Corresponding authors.
E-mail addresses: wangzy@scut.edu.cn (Z. Wang), sharad@ics.uci.edu (S. Mehrotra).

https://doi.org/10.1016/j.physa.2020.125506
0378-4371/© 2020 Elsevier B.V. All rights reserved.

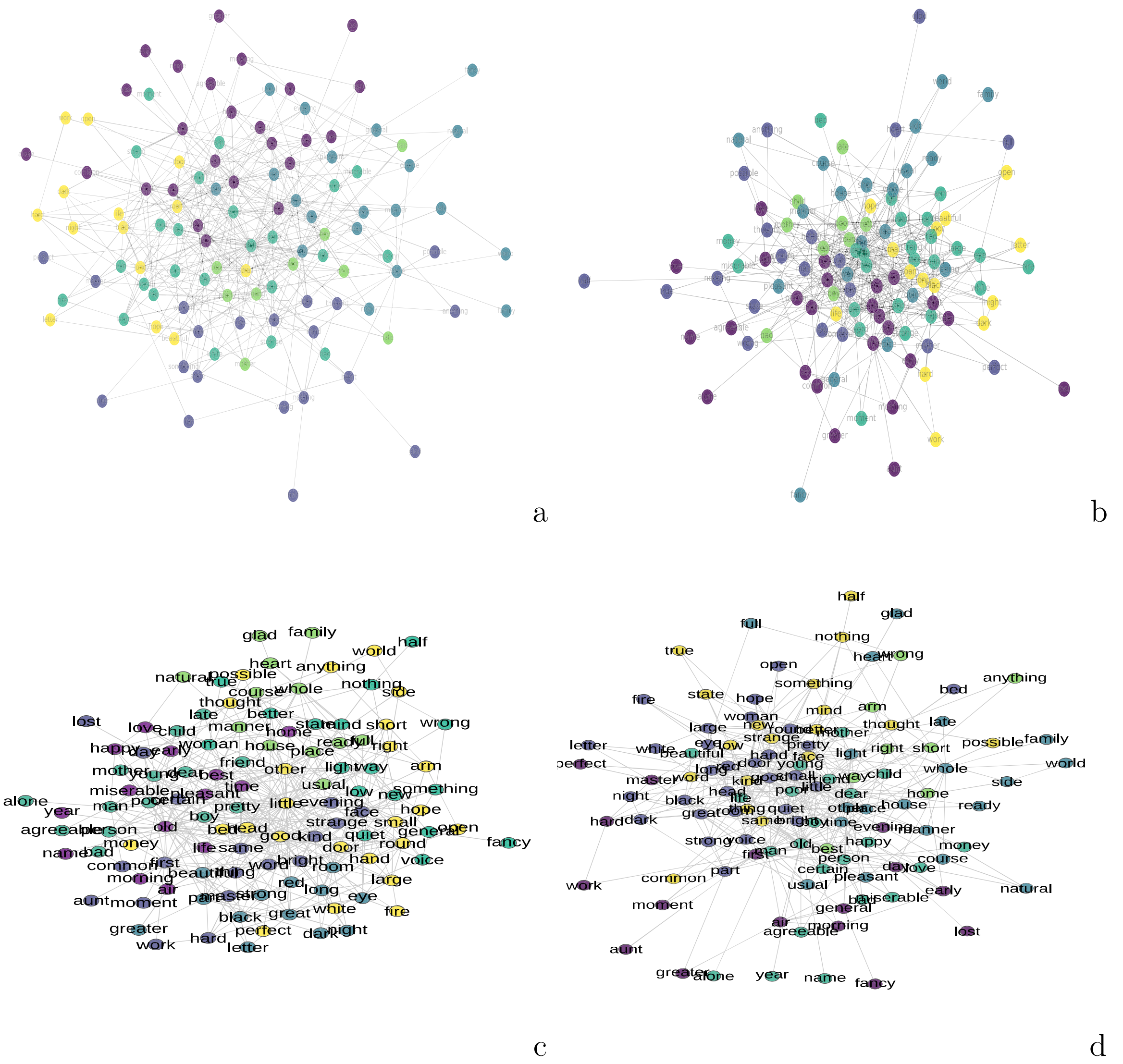

Fig. 1. Visualization results on the Adjnoun Network [7]. (a) Kamada–Kawai spring layout [4]. (b) Fruchterman–Reingold spring layout [5]. (c) Force Atlas2 layout [8]. (d) HQFD layout [9].

How should limitations of force-directed visualization algorithms be solved? Yi et al. [9] applied a quadtree to accelerate the algorithm in the HQFD (high-quality force-directed) algorithm, preventing the physical model from falling into local optimal minimums. Force Atlas2 [8] was proposed by utilizing the degree information of networks to adjust the attractive and repulsive forces. However, these methods still fail to produce a reasonable layout, as shown in Fig. 1(c) and (d). There is still a need for more effective visualization algorithms.

Notably, according to previous research, community structure is a common property in complex networks [1,2]. A community is a group of nodes that are densely connected or highly related and have sparse connections or are less associated with the rest of the groups [2,10]. Typical community structures include the peer community, the core–periphery community, and the multicore community, as shown in Fig. 2. A complex network can be divided into several basic communities as shown in Fig. 3. Since a network can be decomposed into communities, as long as structures that are within and between communities are clearly visualized, the network will be well displayed.

Generally, nodes in the same community tend to be clustered together and should be placed closer in a layout. The nodes in different communities are far away, and they should be far apart in visualization. A good visualization method for community structured complex networks can uncover how nodes are organized inside a community and how different communities are communicated. Moreover, fewer cross-links are expected in layouts [9].

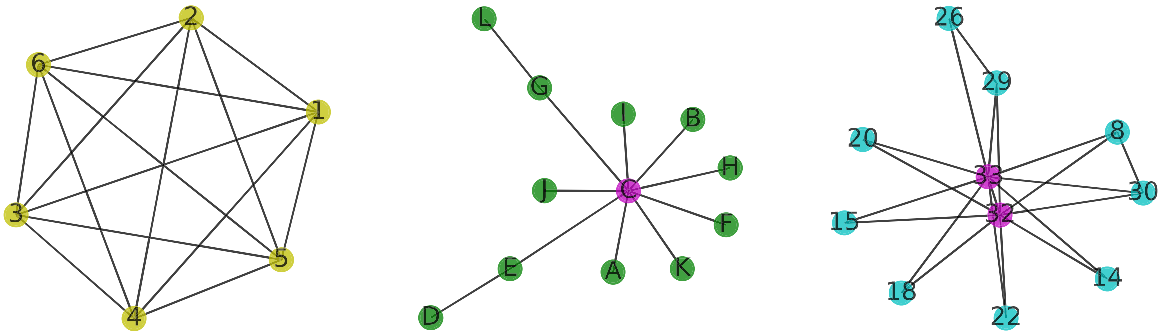

Fig. 2. Three typical community structures. Left: a peer community from an empirical network [11]. Middle: a core–periphery community from a social information propagation network [12]. Right: a multicore community from the Karate Club Network [13].

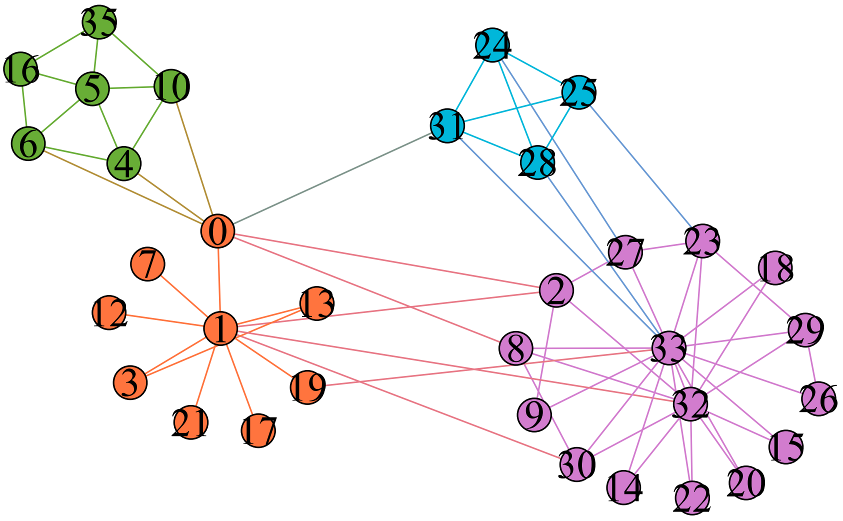

Fig. 3. An example social network that can be decomposed into several communities of different types.

Leveraging the concept of community structures of complex networks, we can display the network structures more clearly, as illustrated in Fig. 4. From the visualization in Fig. 4, we can find that the word “child” is closely related to “mother” and the “bad” is close to “boy” in the green-colored community.

To achieve a better and effective complex network visualization, we explore community structures in visualization and apply it in a generalized repulsive and attractive layout framework. Adaptive weighted attractive and repulsive forces are applied to calculate forces within and between communities. Rather than visualization without using a particular community detection method (e.g., force-directed layout algorithms), we target the combination of the force-directed algorithm and community detection algorithm to improve the quality of graph visualization. Moreover, we extend our method to visualize larger graphs utilizing a coarse graph based on detected hierarchical communities. A novel metric is proposed based on the multivariate Gaussian distribution model to measure the overlapping areas of layouts and the quality of complex network visualization. The metrics take both intra- and inter-community structure visualization into consideration. The metric reflects how good a layout is to a certain extent and can be used as a reference for selecting parameters in the GRA method. The experimental results in various types of complex networks demonstrate our method performs better and yields more insights in visualizing the complex network structures than strong baselines.[1]

2. Related work

Most of the existing effective network visualization algorithms are based on the force-directed model, such as the Kamada–Kawai spring model [4] and Fruchterman–Reingold spring model [5], HQFD [9] and Force Atlas2 [8]. The force-directed model regards the nodes in the network as particles in a physical system with attractive and repulsive forces

[1] The source code of the Java version of the GRA algorithm can be found at https://github.com/packagewjx/gephi/tree/gra, which is built on the basis of Gephi. The Python version will also be released in the link.

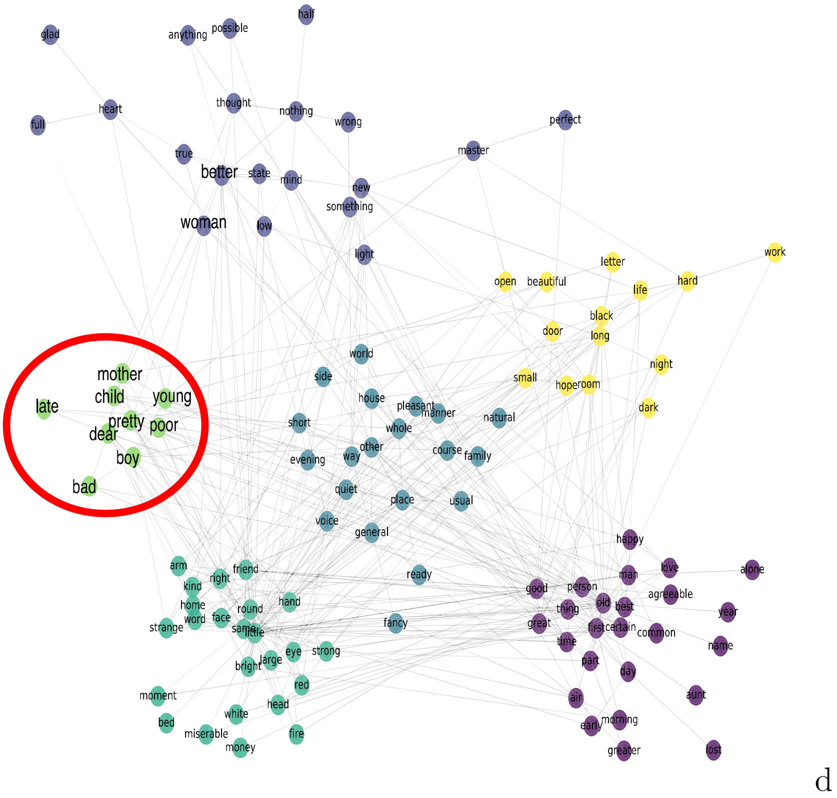

Fig. 4. Visualization on the Adjnoun network [7] by GRA layout.

that work on them. The algorithm produces a desirable placement of the nodes by minimizing the energy of the overall system. The algorithm works well in small and sparse networks. However, it suffers from edge entanglement or the "hairball" issue in large-sized complex networks of high complexity because of the complexity and small-world phenomenon in complex networks [6]. Furthermore, a study found that organized structures facilitate the understanding of the network structure [14]. However, only a few works focused on how to apply community structure into complex network visualization.

There are some related works on visualization that are related to communities and are close to the works in our paper; however, they aim at solving different problems or at different perspectives. One kind of method takes communities or clusters as points or modules and then applies a force-directed model based on them. The LinLog model [15] is first designed for producing a layout that separates different graph clusters, but the internal structure inside a cluster is ignored. Bourqui et al. [16] extended the method of GRIP [17] and computed multilevel layouts using the force-directed method in a top-down manner. This method considers the weighted graph distance to cluster nodes. To overcome the drawback of the classical radar layout in representing group level visualization, multicircular sifting and edge crossings are proposed by Baur et al. [18] to optimize both the vertex order and edge winding. Vehlow et al. [19] targeted visualizing fuzzy overlapping communities. The overlapping nodes are placed between communities. The method visualizes the communities by a circular layout, illustrating a simple solution that produces a clear sketch of entire communities; however, the approach ignores the natural structure inside communities. This kind of method works well in certain cases but fails to provide structures inside communities.

Another kind of method focuses on the local structure of networks or changes the structure of networks to produce a meaningful visualization. Arlind et al. [6] tried to solve the "hairball" issue utilizing edge deletion in small-world graph visualization and produced a clearer clustering structure in the coarse network; nevertheless, this approach caused network information loss by removing parts of edges. The CFinder [20] visualizes small clique communities in biology or social networks by the CPM community detection method [3], but it fails to provide a global view of the whole network. Parveen et al. [21] applied a metric-based method that maps the network structure into a similarity space, which enabled an efficient hierarchical graphical representation of large graphs while failing to produce an intuitive and clear visualization of graphs.

The multiview method is also a popular method to provide visualizations at different levels of complex networks. Auber et al. [22] found that many social networks are small-world networks with a multiscale nature and proposed a method to identify the weakest edges in small-world networks and divide the network into smaller components. The method only works on relatively small networks. The HQFD [9] improves the calculation speed of the Fruchterman–Reingold spring model [5] by the Barnes Hut tree [23] and is embedded into a multiview framework. However, this approach still suffers from similar problems encountered in the Fruchterman–Reingold spring model [5] in some cases as shown in Fig. 1(d).

Several visualization works focus on practical implementation tricks rather than on the design of algorithms. The Force Atlas2 is a fast force-directed algorithm that visualizes large graphs at a faster speed but still requires improvements in terms of quality and structure details [8]. Spinelli et al. [24] developed a plugin to support cluster visualization in Cytoscape. It represents a community as a clicking box that represents a community but loses the structure of the global networks.

Our works focus on visualizing both local and global structures of networks by applying community detection in a generalized repulsive and attractive layout framework. In regard to community detection, various algorithms have been proposed to detect communities in networks, such as GN [1], CPM [3], Louvain [25], EAGLE [11] and MOHCC [26]. The recent work [27] proposed a method for detecting building blocks, namely, groups of network nodes that are usually found together in the same community, which can also be applied with our method in the future. Modularity is used to measure and analyze how well a community division is represented [2]. We propose a metric to analyze the quality of network visualization. Our visualization method, GRA, is based on Louvain [25], which is one of the fastest community detection algorithms. However, GRA is compatible with other community detection algorithms. The experiments in the paper demonstrated that by cooperating community influences into network visualization, the performance can be highly improved.

3. Preliminaries

3.1. Network visualization

Before introducing the algorithm of GRA, we briefly introduce several expectations and objectives in the visualization algorithm as follows:

- Minimizing the number of cross-linked edges.
- Nodes that are close in networks are placed close in the visualization.

To uncover the structure of community structured networks, we consider:

- The nodes inside a community are distinguished.
- Communities should maintain a distance from each other and not highly overlap with each other.
- The node connected with the other community is placed close to the connected community.

A network is noted as $G = (V, E)$, where the V and E are the vertex and edge sets, respectively. The problem of visualizing a network is cast as finding a position $p_i = (x, y)_i$ or $p_i = (x, y, z)_i$ for each node so that the structure of the network can be clearly visualized. To further conduct community structured visualization, the structures within and between communities are expected to be well displayed.

3.2. Community detection

In contrast to previous visualization algorithms that equalize each node, different weights are applied on the repulsive and attractive forces between nodes based on certain strategies to produce favorable visualization. To calculate weights on these forces, the community labels C_i of the nodes are first prepared. The community division is expected to have a high modularity $Q = \frac{1}{2m}\sum_{v,w}(A_{vw} - k_v * k_w/2m)\delta(C_v, C_w)$ [2], where A_{vw} denotes whether there is a link between node v and w, and C_v is the community label of v. The simple version of Q is $\sum e_{ii} - a_i^2$, where e_{ij} is the fraction of edges between the community i and j, and $a_i^2 = \sum e_{ji}$ is the probability that a random edge would fall into community i as described in [28]. The community detection algorithm is based on Louvain [25]. Louvain first initializes small communities and then merges the communities by maximizing the positive gain of modularity $\triangle Q$. The $\triangle Q$ of moving a node i into a community C employs the following formula.

$$\triangle Q = [\frac{\sum_{in} + 2 * k_{i,in}}{2m} - (\frac{\sum_{tot} + k_i - 2 * k_{i,in}}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2], \tag{1}$$

where $\sum_{in}$ is the sum of the edge weights inside community C, $\sum_{tot}$ is the sum of the edge weights incident to nodes in C, k_i is the sum of the edge weights incident to node i, $k_{i,in}$ is the sum of the edge weights from node i to nodes in C and m is the sum of all the edge weights in the network.

4. GRA: Generalized repulsive and attractive model

Our algorithm takes each node as a particle, which is the same in force-directed algorithms. There is a repulsive force between every two nodes and an attractive force within two nodes that have a connection or stay in the same community. The repulsive force is calculated by: $\vec{F}_r(p_i, p_j) = CK^2/d(p_i, p_j)$, where the C regulates the relative strength of the repulsive and attractive forces, and $d(p_i, p_j)$ is visualization distance between node i and j. C is set to 1.0 in this paper. K is the optimal distance [5,29] and set to $\sqrt{S/N}$, where S is the area of the canvas and N is the number of vertexes. The attractive force is measured by: $\vec{F}_a(p_i, p_j) = \lambda * K/d(p_i, p_j)$, where λ is the weight of the edge between node i and j. If the graph is not weighted, the λ is set to 1.0. Then, the forces move the nodes and minimize the energy of the whole network until a stable state is reached. According to previous works [5], a layout with a lower energy has a good visualization. The total energy in the tth iteration is:

$$E(t) = \sum_{v\in V-\{u\}} \frac{\vec{F}_a(u, v)(u - v)w_a^t(u, v)}{d(u, v)} + \sum_{uv\in E} \frac{\vec{F}_r(u, v)(u - v)w_r^t(u, v)}{d(u, v)}, \tag{2}$$

where w_a^t and w_r^t are the weights of the attractive and repulsive force between node u and node v, respectively, in t_{th} iteration. Giving different w_a^t and w_r^t values for forces requires a high computational complexity. To simplify the model, forces between and within communities are given different weights. Therefore, w_a^t is either set to 1.0 if two nodes are inside a community or α when a connection is between communities. To prevent nodes from moving away from the network core area, w_r^t is set to $0.99 \times w_r^{t-1}$ in practice. In addition, there are some constraints such as $d(u, v) \geq \tau$ and $-L/2 \leq v.x \leq L/2$, $-W/2 \leq v.y \leq W/2$ in the canvas, where τ is the minimum distance between u and v, $v.x$ and $v.y$ are the x-axis and y-axis location of node v, and L and W are width and height of the canvas. In practice, for a canvas with size of 1.0 × 1.0, the optimal number of nodes is approximately 150. Thus, the width and height of the canvas is set to $\sqrt{N/150}$. The learning rate η also has a significant impact on graph visualization. We use a linear decay strategy, $\eta^{t+1} = 0.95\eta^t$, where η^t is the learning rate at tth iteration. A decayed learning rate is demonstrated to benefit the model learning, as described in [30]. The model developed on the matrix operation is summarized in Algorithm 1.

Algorithm 1 GRA Algorithm

```
procedure GRA($alpha$, $iters$, $tol$)
    Get community division by COMMUNITY_DETECTION.
    Lay out the network by REPULSE_ATTRACT.
end procedure
procedure REPULSE_ATTRACT
    iter = 0, converged = False
    Initial position matrix P or inherit from previous positions.
    Calculate repulsive and attractive weights W.
    while not converged and iter < iters do
        Calculate distance matrix D by broadcast and L2-norm from P.
        Calculate displacement Disp += ($\vec{F}_r(\mathbf{D}) + \vec{F}_a(\mathbf{W} * \mathbf{D})$)
        $Energy$ = L2-norm(Disp)
        Calculate $\triangle Energy$ by $Energy$ and the value in the last iteration.
        P += Disp * $\eta$ / $Energy$
        Adjust learning rate $\eta$
        if $\triangle Energy/Energy < tol$ then:
            Converged = True
        end if
        iter +=1
    end while
end procedure
procedure COMMUNITY_DETECTION
    1) Each node is given an initial unique community id.
    2) Calculate $\triangle Q$ by Formula (1) with neighbor communities for each community and merge the communities into
the neighbor community with the largest and positive $\triangle Q$.
    3) Build a new coarser graph. Repeat step 2) until no communities are further merged, ending with a dendrogram.
Choose the best division according to modularity $Q$.
end procedure
```

4.1. Graph compression and multilevel visualization

The advantage of applying a community structure in visualization is that it naturally provides a way of graph compression (i.e. graph coarsening [29]) and multilevel visualization [29]. If a complex network grows large, it is difficult to be well visualized and is easily trapped at the local maximum energy. Therefore, we need to compress the networks and give the multilevel visualizations of the entire complex network.

The procedure COMMUNITY_DETECTION in the Algorithm 1 produces a series of coarser or compressed graphs $\{G_0, G_1, \ldots, G_k\}$. In practice, k is a small positive number. The coarse step continues until a graph with a small number

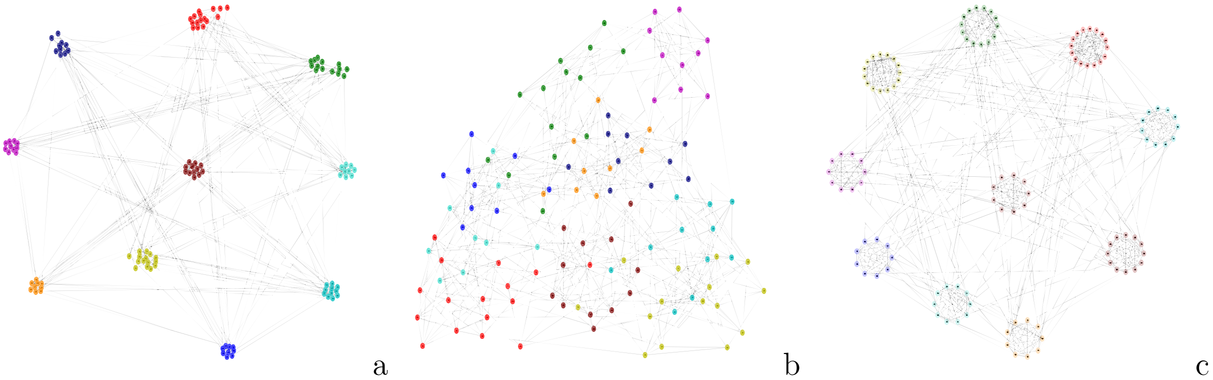

Fig. 5. Results in the Football Club Network [10]. (a) Clustering Visualization, $\alpha = 0.0$. (b) The Fruchterman–Reingold spring layout, $\alpha = 1.0$. (c) The community or clustering visualization in related works [16,19,22]. Groups in color represent different communities.

of nodes remains or $|G_i|/|G_{i-1}|$ is below a threshold ($\rho < 0.75$), which is similar to the work in [29]. The optimal layout for the coarsest graph G_k can be calculated easily. When the layout of the coarser graph G_i is obtained, the graph G_{i-1} is restored by a certain strategy, named the Prolongation_Strategy in Algorithm 2. For larger graphs, the matrix in Algorithm 1 is replaced by the sparse matrix to substantially save memory and calculation costs.

The multilevel layout algorithm with graph coarse is as Algorithm 2:

Algorithm 2 The Multilevel Layout Algorithm

```
procedure MULTILEVEL_ALGORITHM($\rho$, $r$)
    Get the coarsest graph $G^n$ and graph series $\{G^i\}$ by Algorithm 1.
    Prepare prolongation mappings $\{M_i\}$.
    while not $G_0$ do:
        $P^i$ = Layout $G_i$ according to Repulse_Attract in Algorithm 1.
        $P^{i-1}$ = PROLONGATION_STRATEGY($P^i$, $M_i$, $G_i$)
    end while
    Get final layout $P^0$ = Layout $G_0$ according to Repulse_Attract in Algorithm 1.
end procedure
procedure PROLONGATION_STRATEGY($P^i$, $M_i$, $G_i$)
    for node $n$ in $G_i$ do
        Get mapped nodes $M_i[n]$ in $G_i$.
        if len($M_i[n]$) $> 1$ then
            Place $M_i[n]$ around $P_i[n]$ by radius $r$.
        else
            $P_{i-1}[n] = P_i[n]$
        end if
    end for
end procedure
```

4.2. Visualization evaluation

The parameter α has a significant influence on the visualization results. When α is set to 0.0, the attractive force weights between communities are small and the network visualization resembles the clusters in Fig. 5(a). The connections between communities are easily perceived, while the local network structure within a community is not well presented and the space of the canvas is not fully used. When α is set to 1.0, connections between some communities are not intuitively visualized due to highly overlapped areas, as shown in Fig. 5(b). When the community is visualized as shown in Fig. 5(c), the local structure within a community is not visualized. Thus, the choice of a good parameter of α that is able to balance the two optimal objectives remains to be discovered.

From the view of community structured visualization, the objectives of community structured visualization are maximizing the total areas of all the communities while minimizing the overlapping area between different communities. It does not mean that the overlapping nodes are avoided since overlapping nodes or "noisy" nodes commonly occur in complex networks [3], as long as the overlapping areas do not destroy the visualization of the network. Since our model is based on physical force movements, the overlapping nodes are expected to be naturally placed between communities, and the evaluation should be able to tolerate noise.

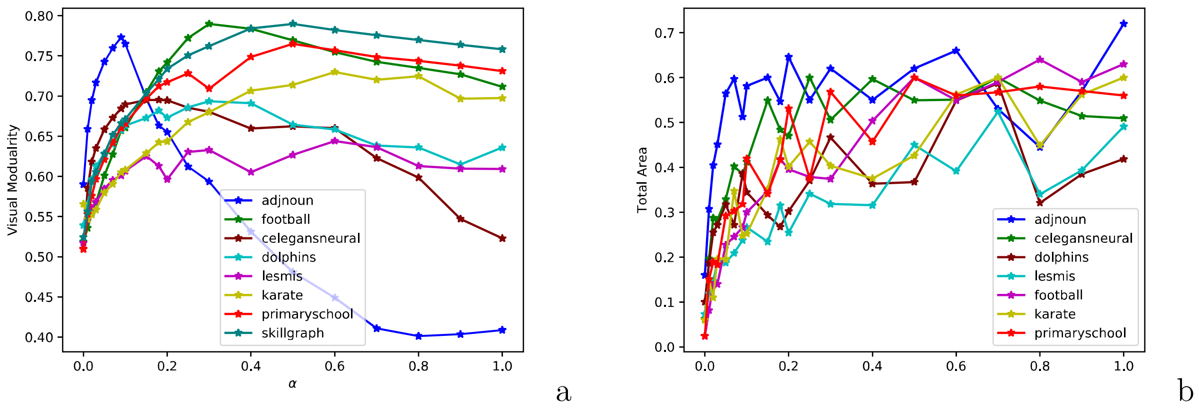

Fig. 6. How the parameter α affects the total area S_t and VQ.

Note that because the process of layout is not conductive to mathematical derivation, we cannot calculate the partial derivatives of parameters and find the best parameters automatically. However, we can design metrics and provide references for parameter selection to distinguish good and poor visualizations.

The total area and overlapping area of all the communities are marked as S_t and S_o, respectively. One candidate method to evaluate the area of S_t and S_o is using a circle or ellipse to fit the nodes in each community. Then, we can easily calculate the major/minor axis of an ellipse, as shown in Fig. 7(a). However, the layout algorithms can produce different shapes of ellipses in various angles and cause deviations when estimating appropriate circles or ellipses for the communities, as shown in Fig. 7(b). A proper metric with noise tolerated is expected. The Gaussian mixture model is a popular probabilistic model for fitting normally distributed subpopulations in a whole population. A multivariate Gaussian mixture model is parameterized by its means and covariances, and the mixture component weights. We applied the multivariate Gaussian model to fit the position distribution of nodes in a community. Then, an ellipse is estimated by limiting nodes within the belief area of $\mu \pm 2 * \delta$. Leveraging the distribution and belief area, the "noisy" overlapping nodes are allowed and do not have a significant influence on core area estimation of communities, as shown in Fig. 7(c). Compared with the method without considering angles or distributions, our fitting method has the least overlapping area but still covers the core area of a community.

$$p(\vec{x}) = \sum_{i=1}^{KC} \phi_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i), \ wherein \sum_{i=1}^{KC} \phi_i = 1 \tag{3}$$

$$\mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^{KC} |\Sigma_i|}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^{\mathrm{T}} \Sigma_i^{-1} (\vec{x}-\vec{\mu}_i)\right) \tag{4}$$

In our method, a two-dimensional Gaussian model is built as in Eqs. (3) and (4). The model is learned by maximum likelihood estimation given the observed data points. The Expectation–Maximization optimization is applied for learning the Gaussian distribution as follows:

- In the Expectation-step, calculate the expectation of the component assignments for each data point given the model parameters μ, and δ.
- In the Maximization-step, maximize the expectations calculated in the Expectation-step for the model parameters and update the values μ and δ.
- Repeat the above steps until the algorithm converges, producing a maximum likelihood estimate.

In our case, $KC = 1$ and $\phi_i = 1.0$, and a multivariate Gaussian distribution is estimated for each community since the community division is known.

The metric is named Visualization Modularity (VQ) in the paper with respect to modularity [2].

$$VQ = \gamma * S_t + (1-\gamma) * (1 - S_o/S_t)^a \tag{5}$$

where S_t is the total area of the communities and S_o is the overlapping area between the communities. To produce a better visualization, we try to increase S_t while minimizing S_o. In this paper, $\gamma = 0.5$, which means the network structures inside and between communities are weighted equally, and a is set to 2.0. The relationship between α and VQ is shown in Fig. 6(a). As shown in Figs. 8(a) and 9(b), even though some overlapping nodes (marked in upper case letters) exist between communities, the metric is still able to estimate proper ellipses for the communities. Some examples are selected to demonstrate that VQ can reflect the quality of community-structured visualizations. An example that reflects how the

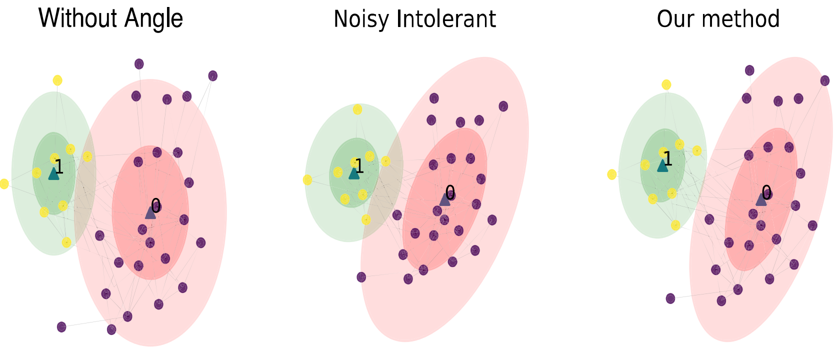

Fig. 7. How different methods influence the calculation of overlapping areas between communities.

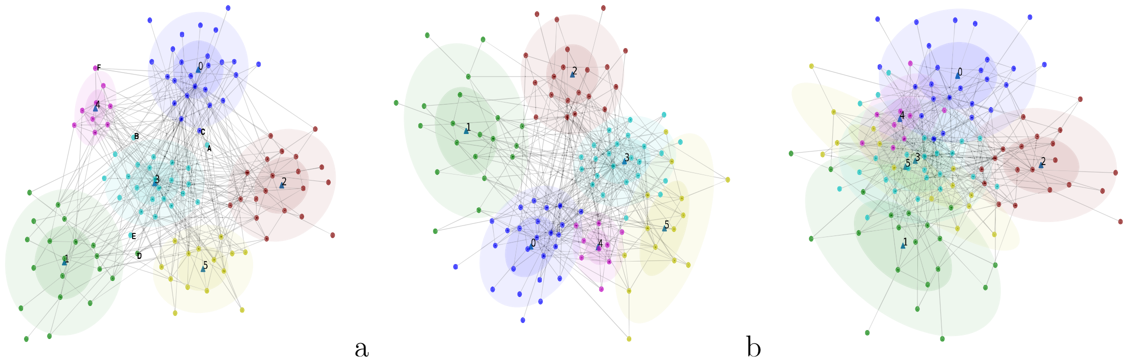

Fig. 8. (a) A visualization example of the Adjnoun network, where $\alpha = 0.05$ and VQ is 0.76. (b) $\alpha = 0.1$, VQ is 0.73. (c) $\alpha = 0.3$, VQ is 0.59.

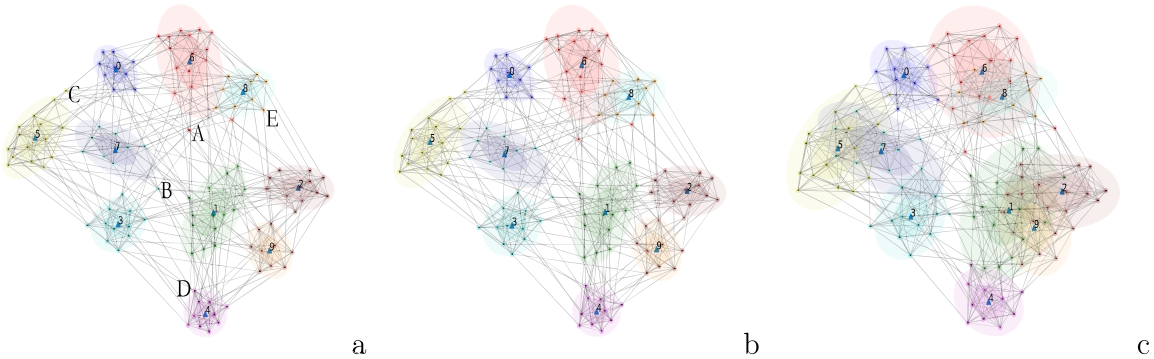

Fig. 9. (a) $\alpha = 0.2$, VQ is 0.74. (b) $\alpha = 0.3$, VQ is 0.78 in the Football Club Network. (c) $\alpha = 1.0$, VQ is 0.72.

parameter influences the visualization of the Adjnoun Network [7] is represented in Fig. 8. The layout with a higher VQ tends to have better performance in revealing the connections between the communities and maintains a clear structure within communities. In the Football Club Network [1], when the *alpha* is set to 0.3, some communities entangled together, e.g., community No.1 and community No.2 are highly overlapped in Fig. 8.

From Fig. 6(a), it can be seen that VQ increases relatively faster with α in the beginning and remains stable or decreases after a certain α value. When parameter α is 1.0, the visualization results of GRA are similar to the Fruchterman–Reingold spring [5]. Some small sparse networks, e.g., the Karate Club Network and the Lesmis Network [31], are easy to be visualized and do not decrease with α after VQ has reached the peak values. The Fruchterman–Reingold spring model performs good visualizations of those networks. However, for some complicated networks, including the Adjnoun Network [7], the Celegansneural Network [32], and the Football Club Network [1], VQ is particularly sensitive to α and decreases sharply after reaching the peaks (the detailed information about the networks is provided in Section 5). Our model produces better visualizations than the Fruchterman–Reingold model [5]. We can select proper parameters based on S_t and S_o since the curves are similar among different networks. However, the satisfactory level of visualization varies

Table 1
The statistical information of the networks used in the paper.

Networks	|V|	|E|	Density	Avg. shortest path
Adjnoun Network	112	425	0.07	2.53
Football Club Network	115	613	0.09	2.5
Karate Network	34	78	0.14	2.4
Dolphin Network	62	159	0.08	3.36
Lesmis Network	77	254	0.09	2.64
Celegansneural Network	297	2345	0.03	3.07
Skill Knowledge Graph	130	1210	0.14	2.19
Primary School Network	236	5899	0.21	1.86
Information Diffusion Network	191	514	0.03	2.61
Butterfly Similarity Network	832	6504	0.02	4.3
Retweet Network	255	376	0.01	3.8
LFR2500	2500	5377	1.7e−03	6.3
BA25000	25,000	24,999	8e−06	12.44
BA100000	100,000	99,999	2e−05	11.03

Table 2
The number of cross-links produced by visualization algorithms on the benchmark networks.

Networks	Circular	FR	KK	ForceAtlas2	HQFD	GRA
Adjnoun Network	45 070	16 974	13 208	**12 360**	14 998	12 830
Football Club Network	102 480	12 982	13 982	9524	10 666	**9044**
Karate Network	1216	142	138	138	154	**134**
Dolphin Network	6710	476	634	468	522	**440**
Lesmis Network	5672	1958	1812	1614	2398	**1376**
Celegansneural Network	1 307 922	473 990	287 990	**222 150**	358 294	248 296
Skill Knowledge Graph	473 548	51 248	55 824	49 370	75 316	**49 408**
Primary School Network	1 005 984	147 812	177 228	162 330	**136 322**	137 318

from person to person. For a network, the model can produce a good visualization at multiple alpha values, e.g., the layouts are acceptable when α is set from 0.05 or 0.1 in the Adjnoun Network, as shown in Fig. 8 and α is set from 0.2 or 0.3 in the Football Club Network, as shown in Fig. 9.

5. Experimental results

The sizes and density of the networks used are summarized in Table 1. The networks will be introduced in detail in later sections. The counts of cross-links of different visualization algorithms are calculated in Table 2. It can be seen that in most cases, our method performs the best or very close to the best performance in these networks. The Force Atlas2 has slightly lower cross-links in some networks. The visualization quality is not always increased when cross-links are decreased, as shown in Fig. 1. However, a terrible layout tends to have more cross-links.

5.1. Benchmark networks

`Adjnoun Network`: The Adjnoun Network [7] is an undirected network of common nouns and adjective adjacencies in the novel "David Copperfield" by the 19th century English writer Charles Dickens. A node represents either a noun or an adjective. An edge connects two words that occur in adjacent positions. The results in Fig. 1 show that our model produces much better visualization than state-of-the-art models.

`Football Club Network`: The Football Club Network includes the American club members in universities and their connections [1]. The visualization produced by our model is shown in Fig. 9(a). Nodes in different colors represent students who come from different clubs. The shallow colored circles represent the Gaussian distribution, and the blue triangles inside are the centers. Compared with the graph clustering visualization in Fig. 5(a), the layout of our model has a clearer network structure visualization inside a community. From the visualization, we have a better recognition of how those clubs are connected and who plays the role of connector between those clubs than that shown in previous visualization methods. In our visualization, if a node has connections with other communities, it is placed toward those communities or as an overlap outside the distribution circle; for example, the node A in the pink community. Even some nodes that are not in the core area in the community, such as node A, B, C, D, and E, they still do not lead to a substantial crash in visualization and have no large influence on understanding the data.

`Celegansneural`: This data set contains the graph of interconnections among the neurons in the Celegans nematode [32]. Our visualization reflects that the cyan community has more connections to the blue community than the green community. Compared with that of our method, the result of FR spring layout [5] does not clearly distinguish different communities and how those community modules are connected (see Fig. 10).

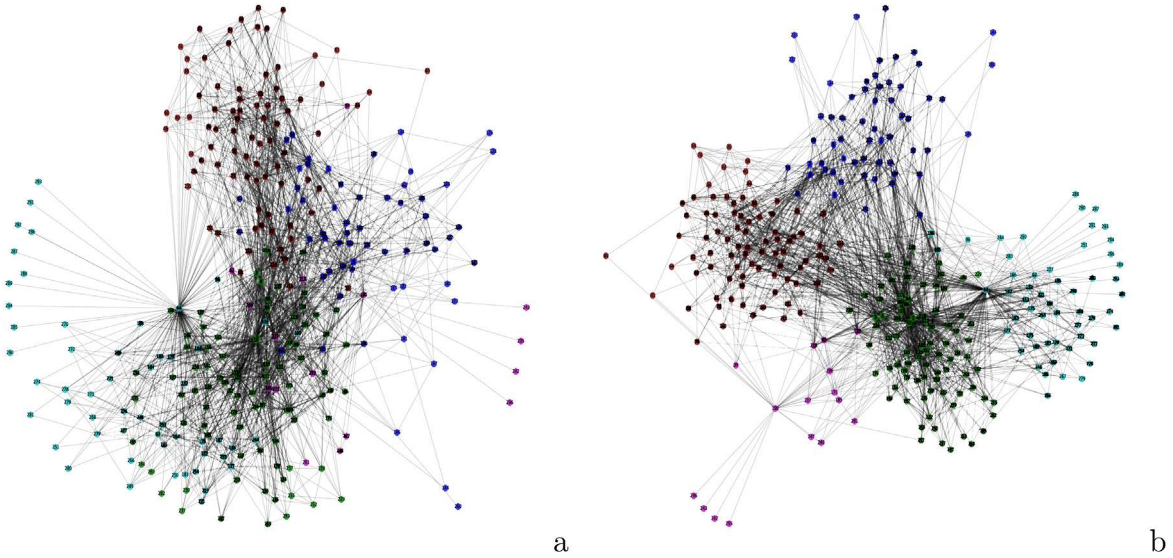

Fig. 10. Results on the Celegansneural neural network. (a) FR spring layout [5]. (b) GRA layout.

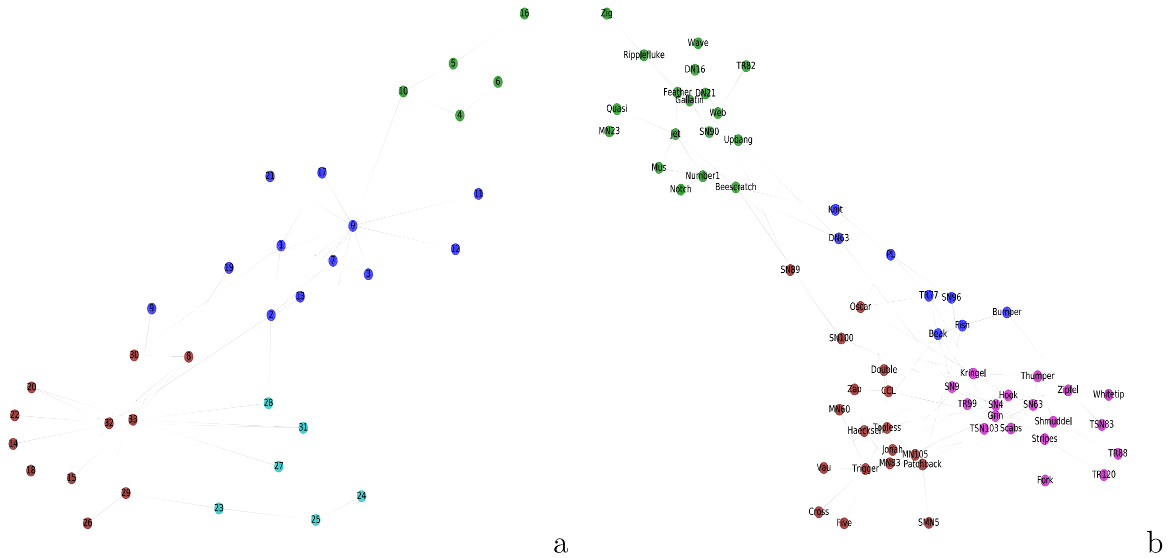

Fig. 11. (a) The Karate Network. (b) The Bottleneck Dolphin Network.

`Karate Network`: The Karate Club Network is a well-known network for testing community detection [13]. Since the network is small and sparse, different types of algorithms are able to produce clear visualizations. In addition, the visual modularity values between different algorithms are reasonably close. Our visualization result is shown in Fig. 11(a).

`Bottleneck Dolphin Network`: The Bottleneck Dolphin Network records the relationship and interactions between different dolphins in New Zealand [33]. From the visualization in Fig. 11(b), we can clearly see that there are two large communities. However, in one community, three small groups (blue, pink and brown communities) are formed, which reveals the hierarchical structures inside some communities.

`Lesmis Network`: Les Misérables is a French historical novel by Victor Hugo, first published in 1862, which is considered one of the greatest novels of the 19th century. The Lesmis Network is built in the paper [31]. A link is made between two nodes (characters) if they co-appear in the novel. The node's color represents the various subgroups to which each character belongs. From the visualization, we can determine that there is a peer community in the brown community and the yellow community in Fig. 12(a). The blue community is a core–periphery community.

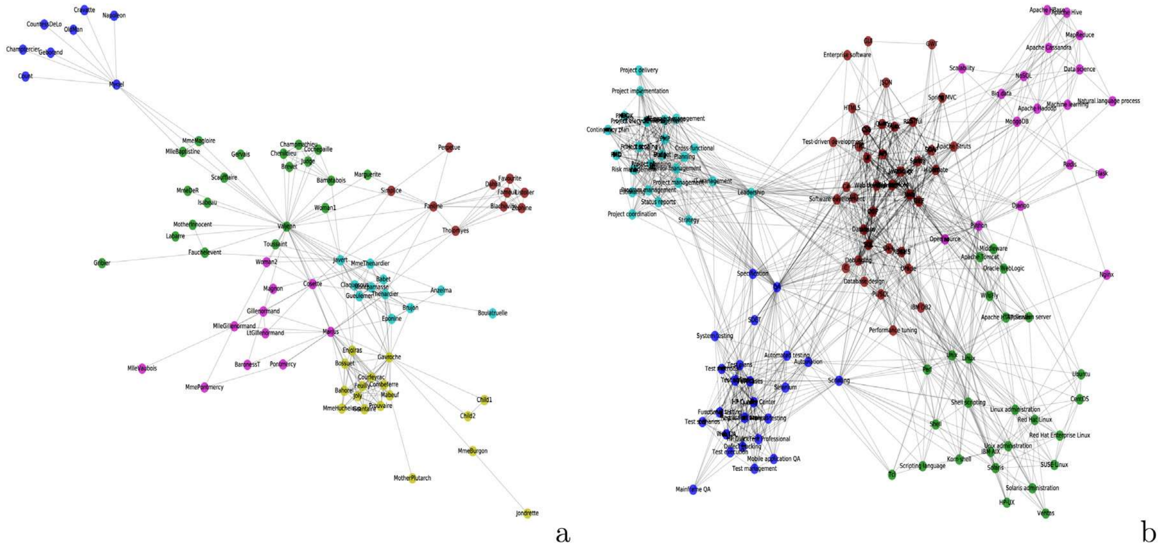

Fig. 12. (a) The Lesmis Network. (b) The Skill Graph Network.

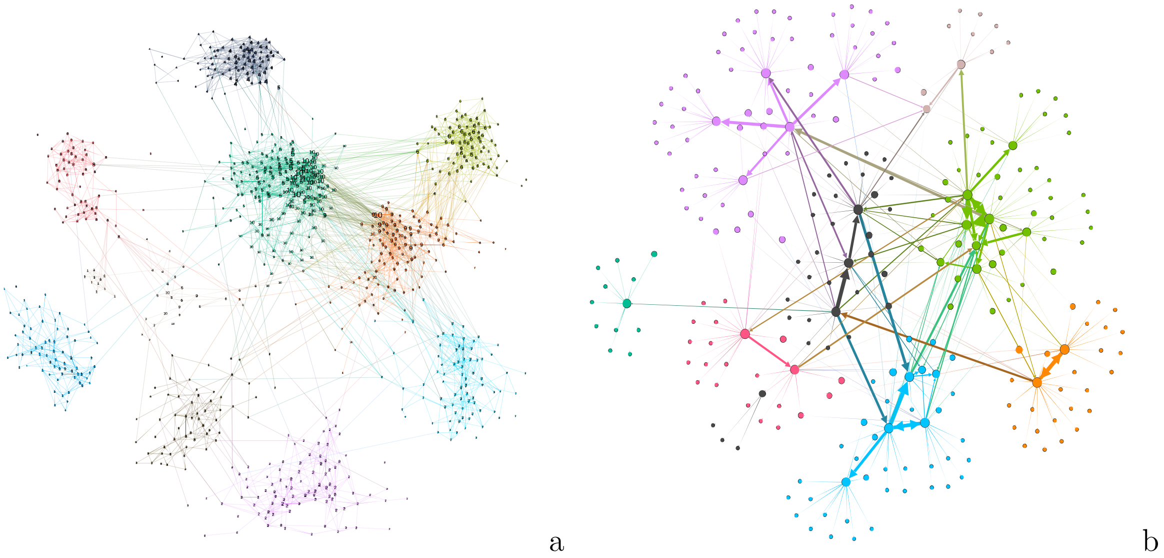

Fig. 13. Results of visualization in weighted networks. (a) The Butterfly Similarity Network (please zoom in for a better view). (b) The retweet Network.

`Skill Knowledge Graph`: The computer related Skill Knowledge Graph is provided by the Dice website.[2] The graph indicates how different skills are connected in resumes in Fig. 12b. The cyan community, green community, blue community are related to project management, Linux system, and test management skills, respectively. The Python skill is highly related to many skills including Web Development, Data Science, Machine Learning, and Natural Language Processing. In web development-related skills, the Python is connected to Django, Flask, Redis, Nginx, MongoDB, Javascript etc., which are popular technologies in web development. Java is closely associated with J2EE, Spring and Hibernate. Our visualization method helps to ascertain some intuitive insights about the data.

`Butterfly Similarity Network`: To show that the proposed method can also be applied to weighted graphs, we visualize two weighted graphs; one is the Butterfly Similarity Network from the paper [34]. Nodes represent butterflies (organisms) and edges represent visual similarities between the organisms. Visual similarities are calculated using butterfly images. From the visualization shown in Fig. 13(a), we can see that the proposed method also works in weighted graphs. Compared with that of the Fruchterman–Reingold spring method [5] (not presented due to space limitations), the

[2] https://www.dice.com.

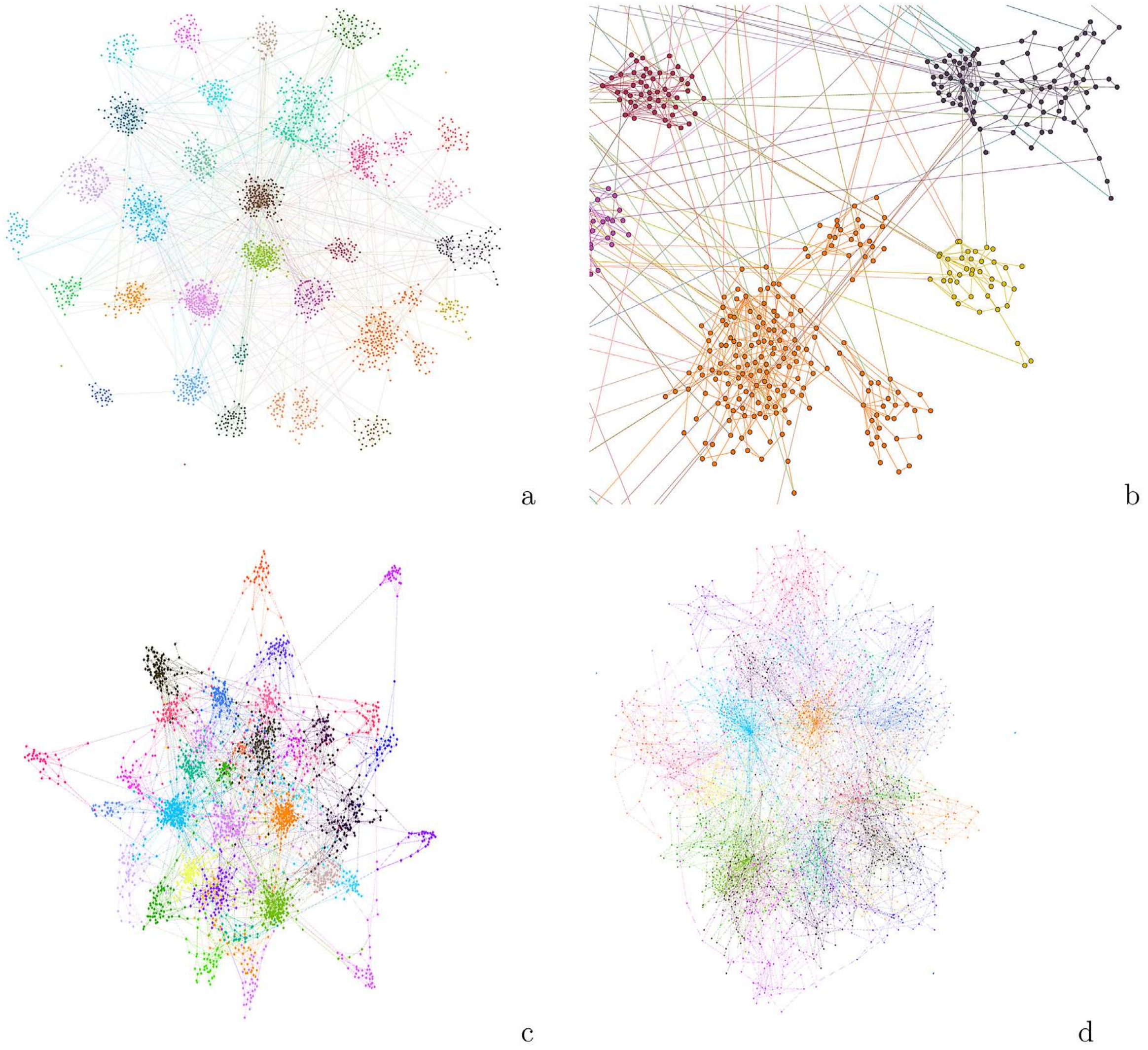

Fig. 14. Results of multiview visualization in a network with 2500 nodes.(a) The layout by GRA. (b) The local structure of communities in the layout by GRA. (c) The layout by Force Atlas2 [8]. (d) The layout by HQFD [9].

layout by GRA has a clearer structure of how each specie of butterfly similar to other species. For example, the butterflies within one species are more likely to gather together. There are more connections between species 4 and species 5 than species 10. There is a small densely connected group in species 1. Some species, e.g., species 2, has much fewer connections with butterflies from other species.

`Retweet Network`: Another weighted graph used in this paper is the Retweet Network from the paper [35]. We construct a social relationship surrounding one user and record retweet interactions between the users. The edge weight is the times of retweet from one user to the other. The edges are weighted and directed. We transform the directed graph into an undirected graph by simply changing the type of directed edges into undirected. The edge weights are also considered in the GRA. The node size in visualization represents their degree that contains information on social influence. From Fig. 13(b), we can see that the nodes linked with higher weights are placed in nearer places than smaller weighted neighbor nodes. By the visualization, one can simply ascertain the intimate friends who are more likely to interact. From the visualization, we can also infer that the person with high social influence or more fans is more likely to retweet the messages from high social influence people in social networks, but they rarely retweet messages from fans with low social influence.

5.2. Larger graph visualization

In this section, we verify the performance of the proposed model on larger graphs. Compared with Force Atlas2 (Fig. 14(c)) and HQFD (Fig. 14 (d)) on a graph with 2500 nodes generated by the LFR [36] benchmark algorithm, GRA

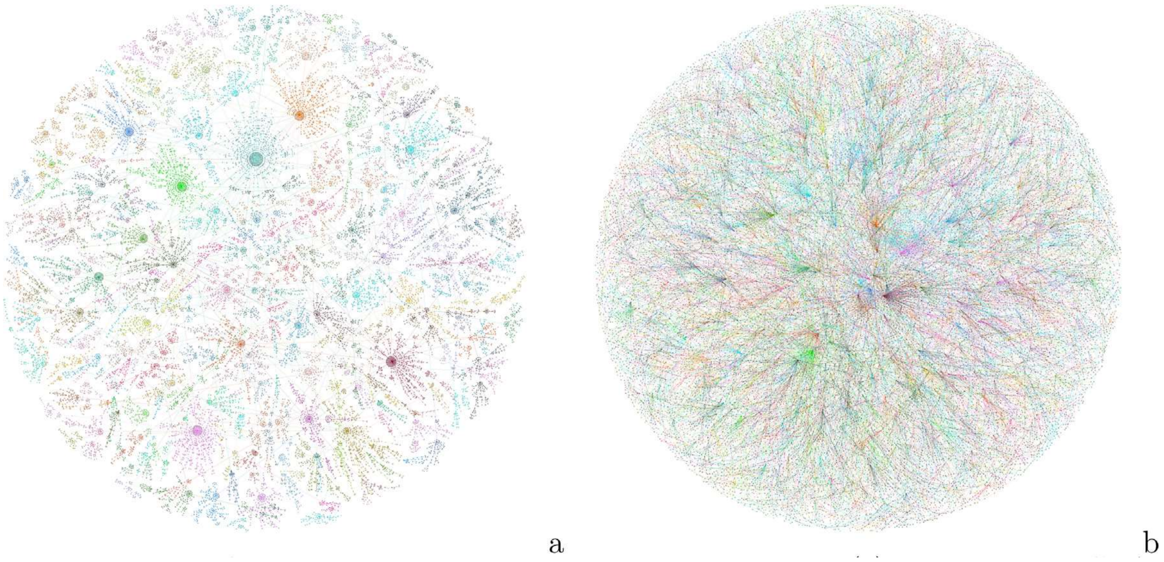

Fig. 15. Results of visualization on a network with 25, 000 nodes. (a) The layout by GRA. (b) The layout by FR spring [5].

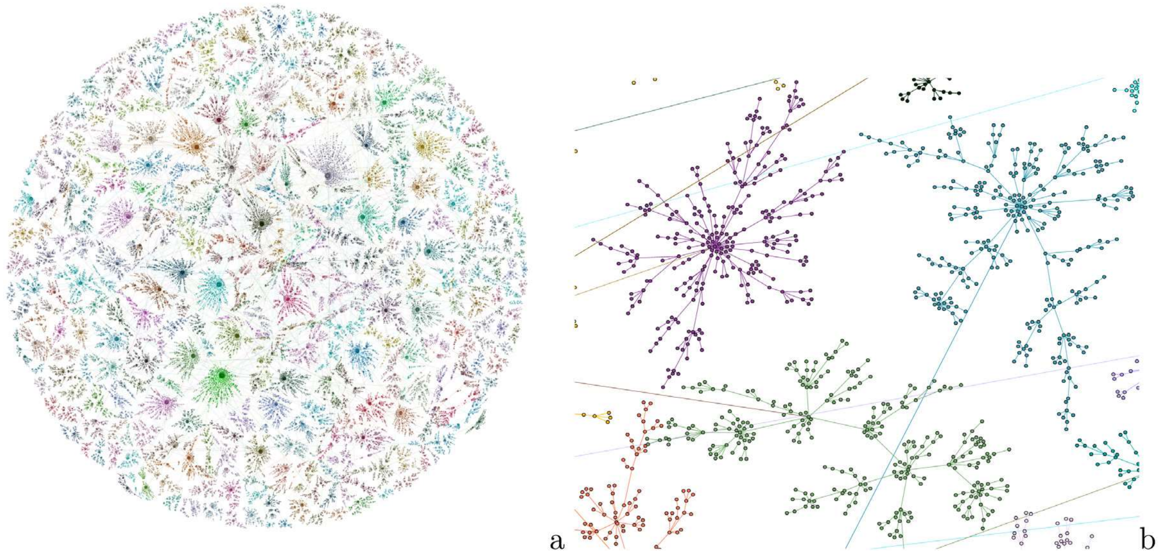

Fig. 16. Results of GRA on the networks with 10, 0000 nodes. (a) The whole layout by GRA. (b) A part of the layout by GRA.

(Fig. 14 (a)) produces a clearer visualization on network structures both within a community and on connections between communities. The layout produced by HQFD has many overlapping areas between communities, causing a global structure of networks that is not clearly visualized. In ForceAtlas2, nodes in the same communities are clustered together, but the structure within communities is not well represented. From the visualization by GRA, we can see both local and global structures of networks and communities, as shown in Fig. 14(a) and (b). Our model can also be applied to complex networks with 25,000 nodes, as shown in Fig. 15 and 100,000 nodes, as shown in Fig. 16. The networks are noted as BA25000 and BA1000000 since they are generated by the BA scale-free network model [37].

5.3. Social interaction networks

There are ten classes and ten teachers in a primary school in France. The proximity-sensing infrastructure based on radio-frequency identification (RFID) devices is deployed to collect data of contacts between students and teachers in the primary school [38]. The challenge to visualize such kind of networks is that their nodes are densely connected. However, even in this extreme case, our model is stable and produces a relatively clear visualization result. Compared with Fig. 17(a)

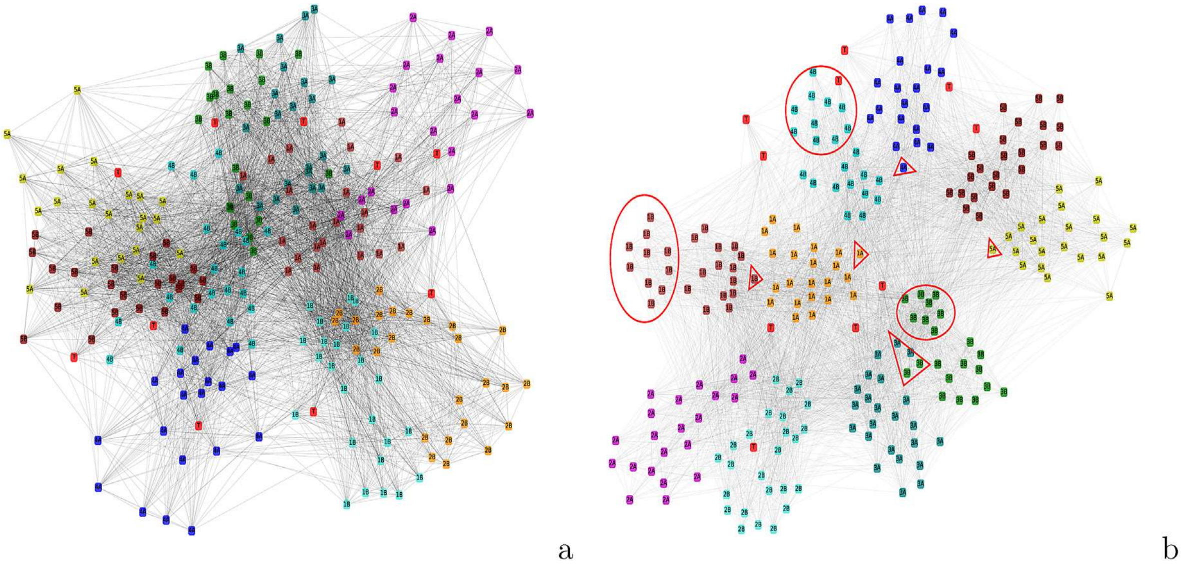

Fig. 17. Visualization in the primary school network by (a) FR spring [5], and (b) GRA, $\alpha = 0.25$.

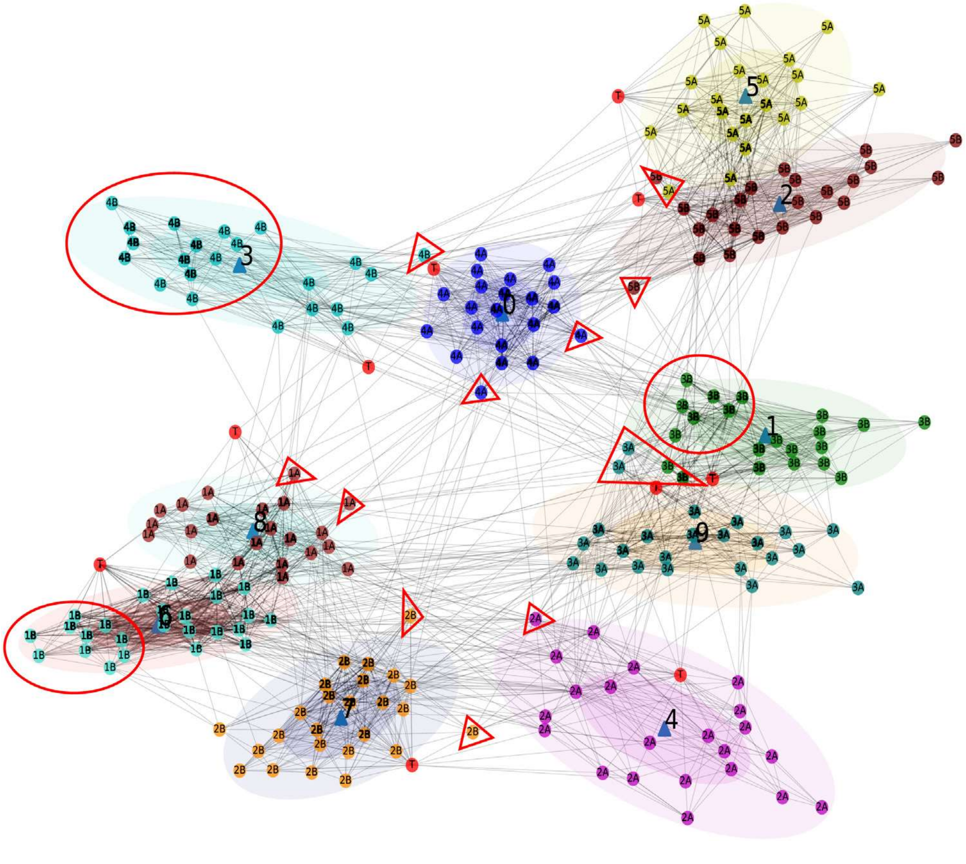

Fig. 18. Visualization in the primary school network by GRA, $\alpha = 0.25$.

of Fruchterman–Reingold spring, our method GRA in Fig. 17(b) produces few cross-edges between communities. The network in Fig. 18 is the sparse network obtained by removing the link with less than 120 s. From the visualization, it can be inferred that the classes in the same or closer grades are more likely to have a connection. For example, class 2B is close to class 2A, 1B and 3A, while it has fewer connections with 5A and 4A. The community 1A can have some small

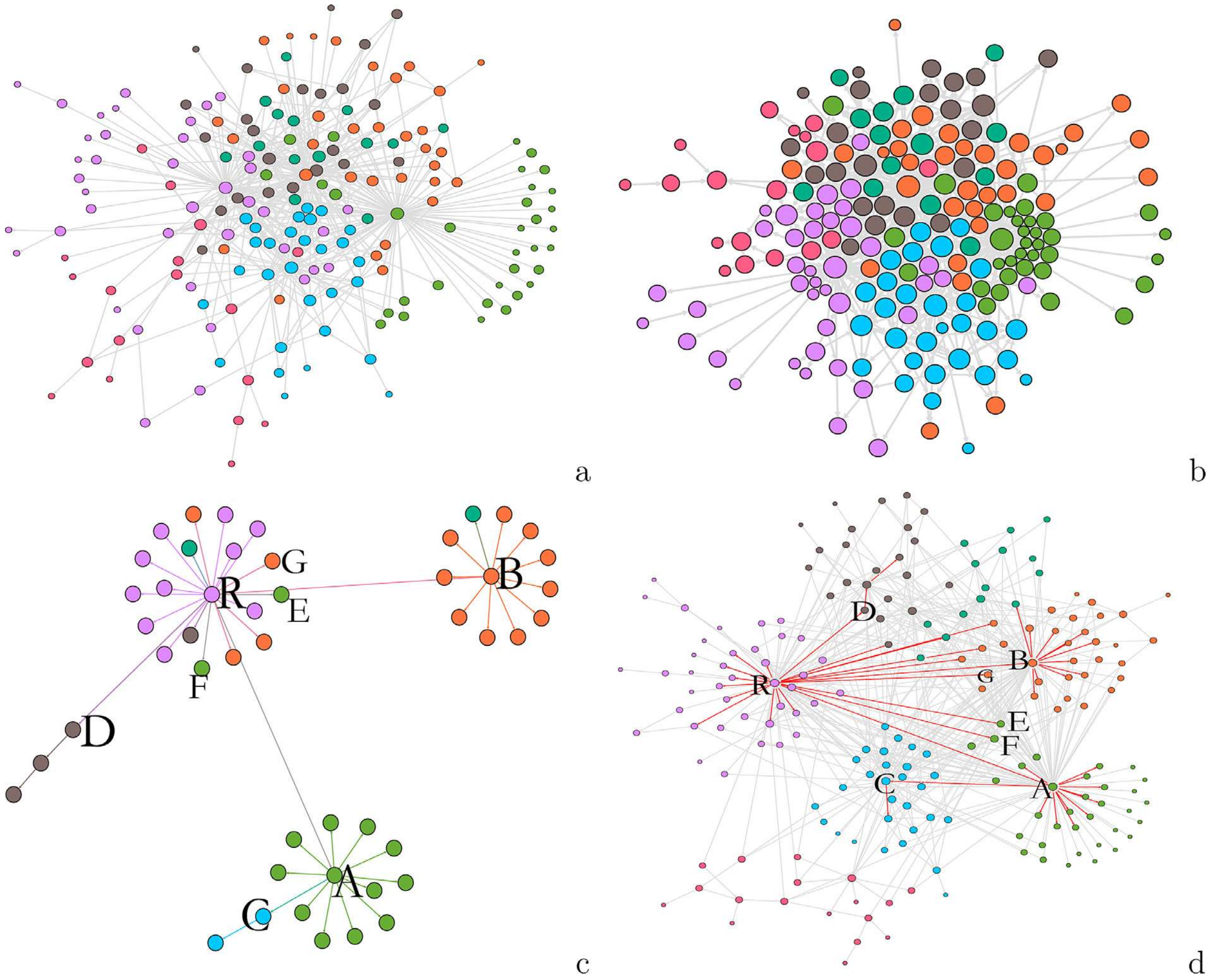

Fig. 19. The visualization on the information diffusion network. (a) Layout by FR spring method [5]. (b) Layout by Force Atlas2 [8]. (c) The information diffusion tree. (d) Layout by GRA and how information propagates across communities. Please zoom in for a better view.

dense groups. Moreover, the overlapping nodes or structural roles that serve as a correspondent between communities are highlighted with the red triangles. In our visualization, the hierarchical structure inside a class can also be naturally visualized. For instance, a small group with seven students (highlighted in red circle) is formed in the class 3B (Fig. 17(b). The results have shown that our method can visualize both global and local structures of networks.

5.4. Information propagation visualization

To demonstrate that our visualization method is able to reveal community structures and facilitate social network analysis, a real-world example of information propagation is introduced. The user social relationship networks are formed by the following functionality in a Twitter-like platform: Sina Microblog. Users can post messages and retweet them according to their interests; therefore, many cascades are formed similar to Facebook cascades [39]. One cascade is in Fig. 19(c) from [12]. The node *R* presents the user who posts the original Microblog, while the rest of the nodes are the users involved in forwarding the Microblog. We visualize both the relationship network and information diffusion tree to discover how information propagates across different communities. Fig. 19(a) depicts a subgraph of the relationship network of those users involved in the cascade visualized by the Fruchterman–Reingold spring layout [5]. Fig. 19(b) shows the same network visualized by ForceAtlas2. Fig. 19(c) shows the visualization produced by GRA. It can be seen that GRA produces a clearer layout that reveals how the communities are connected than that by the Fruchterman–Reingold spring and ForceAtlas2. In visualization by GRA, we can easily understand how information diffuses among different communities. In this case, users can follow multiple friends. For example, in Fig. 19(d), the node *F* is the followers of both the hub node *R* in the cyan community and the hub node *A* in the green community. Therefore, node *F* can receive original and forward Microblogs from *R* and *A* respectively. However, user *F* chooses to forward the Microblog posted by node *R*. This phenomenon commonly occurs in social information propagation since nodes in the different communities can share similar interests to a certain extent or in some dimension. Furthermore, because node *A* in the green community forwarded the information, the message is visible to some nodes (e.g., node *C*) in the blue community and thus forms a deep propagation. Nodes *A* and *D* are referred to as structure holes in the paper [40], and they help the information propagate wider and deeper. From the layout by GRA, we can have an intuitive understanding of how information travels across different communities and what kinds of users lead to another outbreak of information propagation.

6. Conclusions and future works

The community structure is a significant property in complex networks and has various kinds of applications; however, the property is rarely applied in improving current network visualization algorithms. To address the problem of layout algorithms failing to uncover network structures clearly in some complicated cases, a generalized repulsive and attractive model that is aware of community structure has been proposed to visualize the complex network. By exploring community structures and giving proper repulsive and attractive force weights, a better layout of networks can be produced. Compared with previous approaches, our method reflects the local and global structures of networks. Both structures within and between communities are displayed. The detected communities are also applied in graph compression to improve the speed and quality in processing relatively larger graphs. To quantify the quality of community detection and facilitate parameter estimation, a metric is introduced based on a mixture of Gaussian distributions with a noise tolerance. Our model has demonstrated predominant performance compared with strong baselines on many complex networks. The experiments have shown that our model facilitates the analysis of different complex network tasks, such as revealing how information propagates between different communities and how communities in social networks interact with each other.

One of the limitations of the proposed method is that it cannot currently be directly be used on directed graphs or dynamic graphs. In future work, the model can be extended to different directions. One direction is to visualize dynamic communities and directed graphs. It is possible to adopt rich contexts and various relations and links for discovering large heterogeneous networks. The other direction is to apply the Barnes and Hut tree to accelerate the speed of the algorithm.

CRediT authorship contribution statement

Zhenhua Huang: Conceptualization, Methodology, Coding, Software, Experiments, Writing - original draft. **Junxian Wu:** Software developing. **Wentao Zhu:** Concept, Methodology, Revising. **Zhenyu Wang:** Supervision, Validation, Resources, Investigation. **Sharad Mehrotra:** Supervision, Revising. **Yangyang Zhao:** Suggestion, Revising.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Natural Science Foundation, China (No. 61876207), the Natural Science Foundation of Guangdong Province, China (No. 2019A1515011792), Natural Science Foundation of Anhui Province, China (No. 1908085QF283), Science and Technology Program of Guangzhou, China (No. 201802010025), University Innovation and Entrepreneurship Education Fund Project of Guangzhou, China (No. 2019PT103), Guangdong Province Major Field Research and Development Program Project, China (No. 2019B010154004), Doctoral Research Startup Foundation, China (No. 2019jb08), and IBM-CSC Y-100 Young Big Data Scientists program, China. The authors also thank the editors and reviewers for their constructive editing and reviewing.

References

[1] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826, http://dx.doi.org/10.1073/pnas.122653799.

[2] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026–113, http://dx.doi.org/10.1103/PhysRevE.69.026113.

[3] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814, http://dx.doi.org/10.1038/nature03607.

[4] T. Kamada, S. Kawai, et al., An algorithm for drawing general undirected graphs, Inform. Process. Lett. 31 (1) (1989) 7–15, http://dx.doi.org/10.1016/0020-0190(89)90102-6.

[5] T.M. Fruchterman, E.M. Reingold, Graph drawing by force-directed placement, Softw. - Pract. Exp. 21 (11) (1991) 1129–1164, http://dx.doi.org/10.1002/spe.4380211102.

[6] A. Nocaj, M. Ortmann, U. Brandes, Adaptive disentanglement based on local clustering in small-world network visualization, IEEE Trans. Vis. Comput. Graphics 22 (6) (2016) 1662–1671, http://dx.doi.org/10.1109/TVCG.2016.2534559.

[7] M.E. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (3) (2006) 036104, http://dx.doi.org/10.1103/PhysRevE.74.036104.

[8] M. Jacomy, T. Venturini, S. Heymann, M. Bastian, Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software, PLoS One 9 (6) (2014) e98679, http://dx.doi.org/10.1371/journal.pone.0098679.

[9] Y. Hu, Efficient, high-quality force-directed graph drawing, Math. J. 10 (1) (2005) 37–71.

[10] J. Xie, S. Kelley, B.K. Szymanski, Overlapping community detection in networks: The state-of-the-art and comparative study, ACM Comput. Surv. 45 (4) (2013) 43, http://dx.doi.org/10.1145/2501654.2501657.

[11] H. Shen, X. Cheng, K. Cai, M.-B. Hu, Detect overlapping and hierarchical community structure in networks, Physica A 388 (8) (2009) 1706–1712, http://dx.doi.org/10.1016/j.physa.2008.12.021.

[12] Z. Huang, Z. Wang, Y. Zhu, C. Yi, T. Su, Prediction of cascade structure and outbreaks recurrence in microblogs, in: Chinese National Conference on Social Media Processing, 2017, pp. 53–64, http://dx.doi.org/10.1007/978-981-10-6805-8_5.

[13] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473, http://dx.doi.org/10.1086/jar.33.4.3629752.

[14] C. Kosak, J. Marks, S. Shieber, Automating the layout of network diagrams with specified visual organization, IEEE Trans. Syst. Man Cybern. 24 (3) (1994) 440–454, http://dx.doi.org/10.1109/21.278993.

[15] A. Noack, An energy model for visual graph clustering, in: International Symposium on Graph Drawing, Springer, 2003, pp. 425–436, http://dx.doi.org/10.1007/978-3-540-24595-7_40.

[16] R. Bourqui, D. Auber, P. Mary, How to draw clustered weighted graphs using a multilevel force-directed graph drawing algorithm, in: 2007 11th International Conference Information Visualization (IV'07), 2007, pp. 757–764, http://dx.doi.org/10.1109/IV.2007.65.

[17] P. Gajer, S.G. Kobourov, Grip: Graph drawing with intelligent placement, in: Graph Drawing, 2000, http://dx.doi.org/10.1007/3-540-44541-2_21.

[18] M. Baur, U. Brandes, J. Lerner, D. Wagner, Group-level analysis and visualization of social networks, in: Algorithmics of Large and Complex Networks, 2009, pp. 330–358, http://dx.doi.org/10.1007/978-3-642-02094-0_16.

[19] C. Vehlow, T. Reinhardt, D. Weiskopf, Visualizing fuzzy overlapping communities in networks, IEEE Trans. Vis. Comput. Graphics 19 (12) (2013) 2486–2495, http://dx.doi.org/10.1109/TVCG.2013.232.

[20] B. Adamcsek, G. Palla, I.J. Farkas, I. Derényi, T. Vicsek, Cfinder: locating cliques and overlapping modules in biological networks, Bioinformatics 22 (8) (2006) 1021–1023, http://dx.doi.org/10.1093/bioinformatics/btl039.

[21] S. Parveen, J. Sreevalsan-Nair, Visualization of small world networks using similarity matrices, in: BDA, Vol. 8302, 2013, pp. 151–170, http://dx.doi.org/10.1007/978-3-319-03689-2_10.

[22] D. Auber, Y. Chiricota, F. Jourdan, G. Melançon, Multiscale visualization of small world networks, in: IEEE Symposium on Information Visualization 2003, 2003, pp. 75–81, http://dx.doi.org/10.1109/INFVIS.2003.1249011.

[23] J. Barnes, P. Hut, A hierarchical o (n log n) force-calculation algorithm, nature 324 (6096) (1986) 446, http://dx.doi.org/10.1038/324446a0.

[24] L. Spinelli, P. Gambette, C.E. Chapple, B. Robisson, A. Baudot, H. Garreta, L. Tichit, A. Guénoche, M.-C. Brun, Clustsee: A cytoscape plugin for the identification, visualization and manipulation of network clusters, Bio Syst. 113 (2) (2013) 91–95, http://dx.doi.org/10.1016/j.biosystems.2013.05.010.

[25] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (10) (2008) P10008, http://dx.doi.org/10.1103/PhysRevE.69.026113.

[26] Z. Zhang, Z. Wang, Mining overlapping and hierarchical communities in complex networks, Physica A 421 (2015) 25–33, http://dx.doi.org/10.1016/j.physa.2014.11.023.

[27] M.A. Riolo, M. Newman, Consistency of community structure in complex networks, Phys. Rev. E 101 (5) (2020) 052306, http://dx.doi.org/10.1103/PhysRevE.101.052306.

[28] M.E. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133, http://dx.doi.org/10.1103/PhysRevE.69.066133.

[29] C. Walshaw, A multilevel algorithm for force-directed graph drawing, in: International Symposium on Graph Drawing, Springer, 2000, pp. 171–182, http://dx.doi.org/10.1142/9789812773296_0012.

[30] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2015.

[31] D.E. Knuth, The Stanford GraphBase: A Platform for Combinatorial Computing, ACM Press, New York, 1993.

[32] T.B. Achacoso, W.S. Yamamoto, AY's Neuroanatomy of C. Elegans for Computation, CRC Press, 1991.

[33] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behav. Ecol. Sociobiol. 54 (4) (2003) 396–405, http://dx.doi.org/10.1007/s00265-003-0651-y.

[34] J. Wang, K. Markert, M. Everingham, Learning models for object recognition from natural language descriptions, in: BMVC, Vol. 1, 2009, p. 2.

[35] Z. Huang, Z. Wang, Y. Zhu, C. Yi, T. Su, Prediction of cascade structure and outbreaks recurrence in microblogs, in: National Social Media Processing SMP 2017, 2017, pp. 53–64, http://dx.doi.org/10.1007/978-981-10-6805-8_5.

[36] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (4) (2008) 046110, http://dx.doi.org/10.1103/PhysRevE.78.046110.

[37] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, science 286 (5439) (1999) 509–512, http://dx.doi.org/10.1126/science.286.5439.509.

[38] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. Van den Broeck, C. Régis, B. Lina, et al., High-resolution measurements of face-to-face contact patterns in a primary school, PLoS One 6 (8) (2011) e23176, http://dx.doi.org/10.1371/journal.pone.0023176.

[39] J. Cheng, L. Adamic, P.A. Dow, J.M. Kleinberg, J. Leskovec, Can cascades be predicted? in: Proceedings of the 23rd International Conference on World Wide Web, ACM, 2014, pp. 925–936.

[40] T. Lou, J. Tang, Mining structural hole spanners through information diffusion in social networks, in: Proceedings of the 22nd International Conference on World Wide Web, ACM, 2013, pp. 825–836, http://dx.doi.org/10.1145/2488388.2488461.