applied sciences

MDPI

Article

Multivariate Network Layout Using Force-Directed Method with Attribute Constraints

Zhuang Xu [1,2,3,4], Tingyun Mao [1,2,3,4], Guangluan Xu [1,2,*], Yang Wang [1,2] and Daoyu Lin [1,2]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; xuzhuang19@mails.ucas.ac.cn (Z.X.); maotingyun16@mails.ucas.ac.cn (T.M.); ywang1@mail.ie.ac.cn (Y.W.); lindy@aircas.ac.cn (D.L.)
[2] Key Laboratory of Network Information System Technology (NIST), Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China
[3] University of Chinese Academy of Sciences, Beijing 100190, China
[4] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China
* Correspondence: xugl@aircas.ac.cn

Abstract: Graph visualization with proper layout is widely applied to understand the relationship between entities in a complex system and the topological structure information is mainly used. Real-world graphs often have the community structures property which is ignored in many existing graph layout methods. Thus, we propose a multivariate network layout method using the force-directed method with attribute constraints. This method can effectively take into account the hierarchical structure, connection strength, and quantitative comparison between communities. First, the layout of community centers is generated by a force-directed algorithm in which node count of the community is taken as constraints to enable area balance of community; Second, community force based on node attribute is added in the force-directed algorithm to maintain the community clarity. A visualization system is also developed to allow users to interactively generate community structure-aware layout results. qualitative and quantitative evaluation of the results verifies the usability and effectiveness of the proposed method by comparing it with other methods.

Keywords: visualization; multivariate network; attribute constraint; force-directed

check for updates

Citation: Xu, Z.; Mao, T.; Xu, G.; Wang, Y; Lin, D. Multivariate Network Layout Using Force-Directed Method with Attribute Constraints. *Appl. Sci.* **2022**, *12*, 4561. https://doi.org/10.3390/app12094561

Academic Editor: DaeEun Kim

Received: 15 March 2022
Accepted: 28 April 2022
Published: 30 April 2022

1. Introduction

As an effective way to show the relationship between entities, network layout is widely used in a large number of fields, such as social networks, logistics networks, and paper citation networks [1]. In practical scenarios, the networks often contain a lot of additional attribute information, such as the age and hobbies of user nodes in the social network [2], which are called multivariate networks. The layout of multivariate networks requires full consideration of network attribute information while expressing network topology information [3,4].

In the past few decades, many network layout methods have been proposed, and a wealth of multivariate network visualization methods have been realized on this basis, which can be categorized into three categories: node-link, matrix, and attribute. As the most important layout structure, node-link structures are used to display network information in many network visual analysis systems, e.g., Vister [5], Polaris [6], and Scalable Framework [7]. The latest research on node-link structure pays more attention to the time complexity, and many acceleration methods have been proposed [8,9]. Current commonly used node-link structure layout algorithms mainly include two categories: the force-directed method [10–14] that simulates the repulsive force and attraction between nodes in the physical system; and the dimensionality reduction method [14–16] that uses the graph distance matrix to project the graph to a high-dimensional space and then reduce the graph to a low-dimensional space. The force-directed method is simple, easy to implement, and extensible, but lacks the use of attribute information. Similarly, the dimensionality reduction method can maintain the structural consistency, but it has poor scalability and also lacks the use of attribute information.

The matrix structure [17] displays the connection relationship of the network with an adjacency relationship matrix. The hybrid structure combines the node-link structure with the matrix structure [18]. For multivariate networks, although they clearly display the connection relationship of nodes and the attribute information, the network display lacks visual perception.

The attribute structure ignores the structural information of the multivariate network and directly uses the attribute information for layout. It is possible to convert the attributes into coordinates, and then draw the position of the node in the layout space [19]. It is also possible to lay out the network with time attributes according to the time axis [20]. The advantage is that it can display the attribute information of the network without restriction, but the disadvantage is that the visualization lacks structural information, which is not conducive to exploration.

To tackle the above problems, in this paper, we propose a multivariate network layout method using the force-directed method with attribute constraints (FDAC). We utilize the network topology and node attribute information, based on the user's choice, to realize the visual exploration of multivariate networks. First, through the interactive selection from users, the attributes of the nodes are obtained and the communities are allocated, to realize the data preprocessing. Then the attribute community layout is performed to obtain the position information of the attribute community of the node. Finally, the node layout is carried out to realize the visualization of multivariate networks. In the layout process, the attribute information of the nodes is fully utilized, and the layout is realized adaptively according to the user selection. We have implemented an interactive visual system to complete the layout task of the entire multivariate network. The source code of FDAC is available at https://github.com/tddsa/FDAC (accessed on 27 April 2022). Overall, our contributions are as follows:

- We propose a multivariate network layout method FDAC based on attribute constraints. It mainly includes attribute community layout and node layout. Based on the force-directed method, multiple constraints are constructed using attribute information.
- We construct an interactive visualization system. The system allows users to select the focus and attributes of interest, generates the corresponding layout adaptively, and supports the visualization of numerical and non-numerical attributes.
- We propose two evaluation metrics to evaluate our layout results. Through qualitative and quantitative evaluation, we verify the usability and effectiveness of the proposed method compared with other state-of-the-art methods.

The structure of this paper is as follows: Section 2 introduces related work, Section 3 explains the proposed FDAC methods, Section 4 illustrates the visualization system, Section 5 displays the experiment results, and Sections 6 and 7 introduce the discussion and conclusion.

2. Related Work

2.1. Force-Directed Method

The node-link structure draws nodes and edges in the layout space according to certain rules, allowing users to select nodes and edges to achieve the ideal layout [21]. The most basic requirement is to ensure that the nodes are displayed clearly and without overlap, and the edges overlap and cross as little as possible [22]. The force-directed method is the most popular node-link structure. The pioneer of the force-directed layout algorithm is the algorithm of Tutte [23] in 1963. This method is only suitable for three connected graphs and plane graphs. In 1984, Eades [24] proposed a spring embedding algorithm. The core idea is to represent the nodes in the network as rings and the edges as springs. Fruchterman and Reingold [10] proposed the FR algorithm based on the Eades algorithm, using the iterative process of the displacement scale control algorithm to generate a uniformly distributed layout of nodes. Kamada and Kawai proposed the KK algorithm [25], which uses the method of minimizing the energy function of the spring model of the network topology to generate the results. The LinLog [26] algorithm also uses the energy function to realize network visualization, but it can gather the topological structure of the network together to realize the clustering of nodes. Jacomy et al. proposed the ForceAtlas2 [27] algorithm, which achieves speed and accuracy improvements. The stress optimization model [28–32] obtains the optimal solution of the energy function through optimization. First, construct the energy function of the network layout, then derive it, and obtain the optimal solution through iteration. This process is strictly convergent [33]. Davidson and Harel proposed the DH algorithm [34], which uses a simulated annealing algorithm to reduce the moving speed of nodes and prevent nodes from approaching unconnected edges, thereby reducing edge crossing. The DRgraph [9] method approximates the graph distance through a sparse distance matrix, uses a negative sampling technique to estimate the gradient, and uses a multi-level layout scheme to accelerate the optimization process. PIM [8] speeds up the force-guided layout by using an in-memory processing architecture.

2.2. Network Visualization

There are many methods for visual analysis of networks by using topology and attributes. Some methods analyze the network from a global view and show the overall structure to users. For example, Li et al. [35] use a topology-based approach to detect closely connected subgraphs in a network and represent them as abstract nodes to construct multi-level structures. Gemici et al. [36] analyzed the structure of social networks using information from underlying networks and hierarchies. Ge Huang et al. [37] used a radial layout algorithm based on tree layout to explore the hierarchy of the network. Stef et al. [38] used an approach based on attribute information to aggregate nodes with the same attribute value to provide a coarse network representation. Shen et al. [39] proposed a method that simultaneously utilizes network topology information and attributes information. OnionGraph [40] creates a five-level layout through attributes and topologies and allows users to access each level through context. Edge binding [41,42] helps users tease out connections in a graph layout by binding edges that are close together. Other methods analyze the network from a local view, focusing on nodes that users are interested in. FACETS [43] compute fractions according to topology and attributes to find neighbors of nodes that users are interested in. Crnovrsanin et al. [44] recommended nodes associated with focus through filtering. After users select nodes, Refinery [45] obtains correlation information through a random walk algorithm. Laumond et al. [46] allowed users to select multiple nodes continuously to help users obtain the regional structure they are interested in.

3. Methods

In this section, we first introduce the two keys of the FDAC method, i.e., attribute community layout and node layout, and then introduce the two evaluation metrics.

3.1. Attribute Community Layout

We utilize the force-directed method with constraints to calculate the location of the attribute community. The forces received by each community include *spring force*, *repulsive force*, *collision force* and *central force*. The simulated annealing method is used to obtain the community location coordinates. In the following, we introduce the problem illustration.

We define the entire multivariate network as $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ represents n nodes in the network, and $E = \{e_1, e_2, ..., e_m\}$ represents m edges in the network, where each edge $e_k = (v_i, v_j)$ connects two different nodes in the network. On this basis, we obtain the attribute community structure of the multivariate network $G_C = (V_C, E_C)$, where $V_C = \{v_{c_1}, v_{c_2}, ..., v_{c_n}\}$ represents the locations of c_n attribute community centers, and $E_C = \{e_{c_1}, e_{c_2}, ..., e_{c_m}\}$ represents the connection relationships between communities. Each edge $e_{c_k} = \{v_{c_i}, v_{c_j}\}$ connects two different attribute communities. Note that V_C and E_C do not exist in the original multivariate network. For each attribute community center v_{c_i} in V_C, its coordinates are randomly initialized $v_{c_i} = (x_{c_i}, y_{c_i})$. If there is an edge e_k connection between the nodes of two attribute communities, then there is an edge e_{c_k} between the two communities.

3.1.1. Spring Force

Spring force acts on two attribute communities connected by edges, and its calculation formula is as follows:

$$F_a(e_{c_k}) = (d_{pq} - L_{pq}) \cdot \alpha(t) \cdot strength_a(e_{c_k}) \tag{1}$$

where e_{c_k} is the edge of calculating the spring force, and d_{pq} is the Euclidean distance between the centers v_{c_p} and v_{c_q} of the current two attribute communities p and q, which are connected by e_{c_k}. L_{pq} is the ideal distance between the two attribute communities, i.e., the ideal length of edge e_{c_k}, and t represents time. In our method, it indicates the current iteration number of the algorithm. We set α as the global temperature at time t, which is used to implement the simulated annealing algorithm, whose function is to reduce the force on the node as the iteration proceeds, control the severity of the changes in the position of the node and finally achieve the convergence of the algorithm. $strength_a$ is the elastic strength of this edge, just like the elastic coefficient. The formulas of α and $strength_a$ are as follows:

$$\alpha(t) = \alpha(t-1) + (T - \alpha(t-1)) \cdot \alpha_{decay} \tag{2}$$

$$strength_a(e_{c_k}) = \frac{1}{min(deg_p, deg_q)} \tag{3}$$

where T is the target value of simulated annealing, deg_p and deg_q are the degrees of the centers v_{c_p} and v_{c_q} of the two attribute communities. The strength of edge e_{c_k} depends on the two connected nodes. α_{decay} is the decay rate of the simulated annealing algorithm. When α decreases to less than the threshold, the iteration is terminated.

There are differences in the number of nodes in communities with different attributes. Therefore the ideal distance between communities L_{pq} is not static. The community radius should be determined according to the number of nodes in the community. The formula of L_{pq} is as follows:

$$L_{pq} = L_0 + L_{max} \cdot \left(\sqrt{\frac{n_{c_p}}{n}} + \sqrt{\frac{n_{c_q}}{n}}\right) \tag{4}$$

where L_0 is the basic distance between communities, n is the number of all nodes, n_{c_p} and n_{c_q} are the number of nodes contained in the two attribute communities p and q. L_{max} is the maximum distance of the edge, depending on the size of the layout area. The calculation formula is as follows:

$$L_{max} = \frac{1}{2}\sqrt{W^2 + H^2} \tag{5}$$

where W and H are the width and height of the layout area, respectively. According to the Verlet [1] integration algorithm of particle motion simulation, the velocity changes Δv_{ax} and Δv_{ay} of the attribute communities p and q in the horizontal and vertical directions under the action of the attractive force F_a can be calculated. The formulas are as follows:

$$\Delta v_{ax}(q) = \frac{x_{pq}}{d_{pq}} \cdot F_a \cdot b_{pq} \tag{6}$$

$$\Delta v_{ay}(q) = \frac{y_{pq}}{d_{pq}} \cdot F_a \cdot b_{pq} \tag{7}$$

$$\Delta v_{ax}(p) = -\frac{x_{pq}}{d_{pq}} \cdot F_a \cdot (1 - b_{pq}) \tag{8}$$

$$\Delta v_{ay}(p) = -\frac{y_{pq}}{d_{pq}} \cdot F_a \cdot (1 - b_{pq}) \tag{9}$$

$$b_{pq} = \frac{deg_p}{deg_p + deg_q} \tag{10}$$

where x_{pq} and y_{pq} are the distances between the attribute communities p and q in the horizontal and vertical directions, respectively. b_{pq} is determined by the degree of the center of the two attribute communities v_{c_p} and v_{c_q}. This makes the node with the smaller degree at both ends of the same edge move more.

3.1.2. Repulsive Force

The repulsive force is the force that an attribute community is repelled by others, and its function is to spread the entire network layout on the plane. For community p, the formula for the repulsive force exerted by the community q is as follows:

$$F_r(p,q) = \alpha(t) \cdot strength_r(p,q) \cdot \delta \tag{11}$$

where $strength_r(p,q)$ is the repulsive strength of the community p by the community q, δ is a multi-segment function, the formula is as follows:

$$strength_r(p,q) = \sqrt{\frac{n_{c_p} + n_{c_q}}{n}} \cdot S_{max} \tag{12}$$

$$\delta = \begin{cases} 1, d_{pq} \leq L_{pq} \\ k, d_{pq} > L_{pq} \end{cases} \tag{13}$$

where S_{max} is the maximum setting of the repulsive strength and the default value of S_{max} is 600, and k is a coefficient between 0 and 1. When the distance between the communities is less than the target distance, the node is subject to greater repulsion. Finally, the horizontal and vertical velocity changes of the attribute community p under the action of repulsive force are as follows:

$$\Delta v_{rx}(p) = -\frac{x_{pq}}{d_{pq}} \cdot F_r \tag{14}$$

$$\Delta v_{ry}(p) = -\frac{y_{pq}}{d_{pq}} \cdot F_r \tag{15}$$

3.1.3. Collision Force

Collision force is the force that two attribute communities receive when they overlap. It acts to repel the two communities in opposite directions along the centerline, which is similar to the repulsive force. When the actual distance is greater than the sum of the community radius, the collision force will disappear. The formula is as follows:

$$F_c(p,q) = \begin{cases} -strength_c \cdot (1 - \frac{d_{pq}}{L_{pq} - L_0}) & , d_{pq} \leq L_{pq} - L_0 \\ 0 & , d_{pq} > L_{pq} - L_0 \end{cases} \tag{16}$$

where $strength_c$ is the default collision strength. The closer the two communities are, the greater the collision force they receive. Under the effect of the collision force, the velocity changes of the two overlapping attribute communities p and q in the horizontal and vertical directions are as follows:

$$\Delta v_{cx}(q) = \frac{x_{pq}}{d_{pq}} \cdot F_c \cdot b_{pq} \tag{17}$$

$$\Delta v_{cy}(q) = \frac{y_{pq}}{d_{pq}} \cdot F_c \cdot b_{pq} \tag{18}$$

$$\Delta v_{cx}(p) = -\frac{x_{pq}}{d_{pq}} \cdot F_c \cdot (1 - b_{pq}) \tag{19}$$

$$\Delta v_{cy}(p) = -\frac{y_{pq}}{d_{pq}} \cdot F_c \cdot (1 - b_{pq}) \tag{20}$$

$$b_{pq} = \frac{{n_{c_q}}^2}{{n_{c_p}}^2 + {n_{c_q}}^2} \tag{21}$$

The purpose is to make the attribute community with many nodes move slowly, and make the community with few nodes move fast. This guarantees the stability of the network visualization.

3.1.4. Central Force

The central force refers to the force received by each attribute community from the center of the layout area. It makes the center point of the entire network layout close to or even coincides with the center point of the layout area, ensuring that the network is drawn in the center of the layout area. Define the coordinates of the center point of the layout area as (x_o, y_o), and the coordinates of the center point of the network as (x'_o, y'_o), the formula is as follows:

$$x_o = \frac{W}{2} \tag{22}$$

$$y_o = \frac{H}{2} \tag{23}$$

$$x'_o = \frac{\sum_{i=1}^{c_n} n_{c_i} \cdot x_i}{n} \tag{24}$$

$$y'_o = \frac{\sum_{i=1}^{c_n} n_{c_i} \cdot y_i}{n} \tag{25}$$

where (x_i, y_i) is the coordinates of the center of the i-th attribute community, n_{c_i} is the number of nodes in the i-th attribute community. This makes it possible to consider the influence of communities with more nodes on the central location when calculating the central location of the network. The formula of the central force is as follows:

$$F_o(p) = \sqrt{(x'_o - x_o)^2 + (y'_o - y_o)^2} \cdot strength_o \tag{26}$$

where $strength_o$ is the strength of the central force. Therefore, the velocity change of the attribute community p in the horizontal and vertical directions under the action of the central force is as follows:

$$\Delta v_{ox}(p) = (x'_o - x_o) \cdot F_o \tag{27}$$

$$\Delta v_{oy}(p) = (y'_o - y_o) \cdot F_o \tag{28}$$

Since the expression of the result of the force in each attribute community is the speed in the horizontal and vertical directions, in the process of each iteration, these speeds can be accumulated, and at the end of each iteration, the location of the attribute community is moved. Therefore, combining the effects of the above four forces on the community center, the total speed change is as follows:

$$\Delta v_x(p) = \sum_a \Delta v_{ax}(p) + \sum_r \Delta v_{rx}(p) + \sum_c \Delta v_{cx}(p) + \Delta v_{ox}(p) \tag{29}$$

$$\Delta v_y(p) = \sum_a \Delta v_{ay}(p) + \sum_r \Delta v_{ry}(p) + \sum_c \Delta v_{cy}(p) + \Delta v_{oy}(p) \tag{30}$$

where $\sum_a \Delta v_{ay}(p)$ is the total amount of velocity change in the horizontal direction of the attribute community p under the attractive action, and the formulas of repulsive and collision forces are similar. Finally, the update formula of the velocity change to the location of the center (x_{c_p}, y_{c_p}) of the attribute community p is as follows:

$$x_{c_p} = x'_{c_p} + (v'_x(p) + \Delta v_x(p)) \cdot \beta \tag{31}$$

$$y_{c_p} = y'_{c_p} + (v'_y(p) + \Delta v_y(p)) \cdot \beta \tag{32}$$

where (x'_{c_p}, Y'_{c_p}) is the position of the center of the attribute community p in the previous iteration, $v'_x(p)$ and $v'_y(p)$ are the speed of the previous iteration, and β is the decay rate of speed, and the default value of β is 0.5, which is used to control the severity of changes in node position.

Therefore, as the algorithm runs and the number of iterations increases, the global temperature α of the simulated annealing algorithm continues to decrease, thereby gradually achieving the convergence of the algorithm, and finally obtaining the ideal location of the center of each attribute community.

3.2. Node Layout

Similar to the Section 3.1 attribute community layout method, in the node layout process, each node will be also subject to attraction, repulsion, collision, and central force. In addition, each node is also subject to the community force exerted by its corresponding attribute community. If each attribute community is regarded as a circular area, then the role of community force is to draw the nodes of the same attribute to the range of the circular area. However, only relying on the action of these five forces cannot achieve satisfactory visualization results, so additional constraints must be introduced to these forces to achieve multivariate network visualization based on attribute constraints.

3.2.1. Spring Force Constraint

Spring force acts on nodes connected by one edge. The difference from the spring force of attribute community layout is that the ideal distance L_{ij} between the two nodes i and j.

When the attributes of two nodes are the same, the two nodes belong to the same attribute community. Therefore, the formula for calculating L_{ij} is as follows:

$$L_{ij} = L_{max} \cdot \sqrt{\frac{n_{c_p}}{n}} - r(\sqrt{\frac{deg_i}{deg_{max}}} + \sqrt{\frac{deg_j}{deg_{max}}}) \tag{33}$$

where r is the default radius of the node, deg_i is the degree of the node i, deg_{max} is the maximum degree of all nodes.

When the attributes of two nodes are different, the two nodes belong to different attribute communities p and q. Therefore, the formula for calculating L_{ij} is as follows:

$$L_{ij} = \sqrt{(x_{c_p} - x_{c_q})^2 + (y_{c_p} - y_{c_q})^2} \tag{34}$$

3.2.2. Repulsive Force Constraint

For each node, the repulsive force by the node with the same attribute and by the node with different attributes should be different, so the repulsive force strength of node i received by node j is modified; the formula is as follows:

$$strength_r(i,j) = \begin{cases} S & ,a_i = a_j \\ \frac{\sqrt{(x_{c_p} - x_{c_q})^2 + (y_{c_p} - y_{c_q})^2}}{L_{max}} \cdot S \cdot \mu & ,a_i \neq a_j \end{cases} \tag{35}$$

where a_i is the attribute of the node i, S is the default repulsive force strength, the default value S is -30 , and μ is an adjustable parameter.

3.2.3. Community Force

Community force refers to the attraction that the center of the attribute community exerts on the nodes of the same attribute. Its function is to draw the nodes of the same attribute into the same attribute community. The formula for node i in the attribute community to be attracted by the community center $v_{c_p} = (x_{c_p}, y_{c_p})$ is as follows:

$$F_k(i) = (d_{ip} - L_{max} \cdot \sqrt{\frac{n_{c_p}}{n}} \cdot \sigma) \cdot \alpha(t) \cdot strength_k(i, p) \tag{36}$$

where d_{ip} is the Euclidean distance between the node i and the community center v_{c_p}, σ is a parameter to control the ideal distance between the node and the community center, $strength_k(i, p)$ is the strength of the community force. Its formula is as follows:

$$strength_k(i, p) = S \cdot \sqrt{\frac{n_{i-d}}{n_{i-s}}} \tag{37}$$

where S is the default value of community force strength, n_{i-d} is the number of nodes with different attributes connected to node i, n_{i-s} is the number of nodes with the same attributes connected to node i. In this way, nodes with more different types of nodes than nodes of the same type in the neighborhood can be closer to their own attribute communities to maintain discrimination. Further, under the influence of the community force, the velocity change of the node i in the horizontal and vertical directions is:

$$\Delta v_{kx}(i) = \frac{x_{ip}}{d_{ip}} \cdot F_k \cdot b_{ip} \tag{38}$$

$$\Delta v_{ky}(i) = \frac{y_{ip}}{d_{ip}} \cdot F_k \cdot b_{ip} \tag{39}$$

$$b_{ip} = (\frac{deg_{p-max}}{deg_i + deg_{p-max}})^2 \tag{40}$$

where x_{ip} and y_{ip} are the distance between node i and the center of the attribute community v_{c_p} in the horizontal and vertical directions, respectively, and deg_{p-max} is the maximum value of the node degree in the attribute community p.

Finally, under the action of attractive force, repulsive force, collision force, central force, and attribute community force, the position of each node is calculated; the formula is as follows:

$$\Delta v_x(i) = \sum_a \Delta v_{ax}(i) + \sum_r \Delta v_{rx}(i) + \sum_c \Delta v_{cx}(i) + \Delta v_{ox}(i) + \Delta v_{kx}(i) \tag{41}$$

$$\Delta v_y(i) = \sum_a \Delta v_{ay}(i) + \sum_r \Delta v_{ry}(i) + \sum_c \Delta v_{cy}(i) + \Delta v_{oy}(i) + \Delta v_{ky}(i) \tag{42}$$

$$x_i = x_i' + (v_x'(i) + \Delta v_x(i)) \cdot \beta \tag{43}$$

$$y_i = y_i' + (v_y'(i) + \Delta v_y(i)) \cdot \beta \tag{44}$$

Algorithm 1 is a flow chart of the entire algorithm process, which introduces the overall layout algorithm of the multivariate network with attribute constraints.

Algorithm 1: Attribute Constraint Layout

```
Input: $G = (V, E)$: a multivariate networks; $T$: the target temperature; $\alpha$: the
       current temperature
Output: $V = \{v_i, v_2, ..., v_n\}$: the position of nodes
Randomly initialize the attribute community center $V_C = \{v_{c_1}, v_{c_2}, ..., v_{c_n}\}$;
for every pair of nodes in G do
    Generate edges $E_C = (e_{c_1}, e_{c_2}, \ldots, e_{c_m})$ between communities according to the
     connection relationship between nodes;
while $\alpha > T$ do
    Update current temperature $\alpha$;
    for each node $v_{c_i}$ in $V_C$ do
        Get the velocity change $\Delta v_a, \Delta v_r, \Delta v_c and \Delta v_o$;
        Get the total velocity change $\Delta v$;
        Update the position of $v_{c_i}$;
while $\alpha > T$ do
    Update current temperature $\alpha$;
    for each node $v_i$ in $V$ do
        Get the velocity change $\Delta v_a, \Delta v_r, \Delta v_c, \Delta v_o and \Delta v_k$;
        Get the total velocity change $\Delta v$;
        Update the position of $v_i$;
return $V$;
```

3.3. Evaluation Metrics

This paper proposes two evaluation metrics to measure the clarity and discrimination of the attribute community in the layout results from two aspects.

The first metric is the *average distance between nodes in attribute communities* (ADIAC). This metric can measure the degree of cohesion of the attribute community. On the one hand, the index reflects the clustering degree of nodes within the attribute community. The smaller its value is, the higher the distinguishable degree of the community is. On the other hand, when the ADIAC tends to 0, it means that nodes are stacked towards the community center, which will cause serious node overlap and make it difficult to show the connection relationship between nodes. The formula of ADIAC is as follows:

$$\text{ADIAC} = \frac{1}{m}\sum_{k=1}^{m}\frac{1}{n_k}\sum_{i\neq j\cap i,j\in C_k}\sqrt{(x_i^* - x_j^*)^2 + (y_i^* - y_j^*)^2} \tag{45}$$

where m is the number of all communities under the constraint of the attribute, C_k is the k-th attribute community, n_k is the number of nodes in C_k, i and j are two different nodes in C_k, x_i^* is the abscissa of node i after maximum and minimum normalization, and y_i^* is the ordinate. The formula for maximum and minimum normalization is as follows:

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{46}$$

where x is the abscissa of the node in the layout result, x_{min} is the minimum value of the abscissa of all points, x_{max} is the maximum value of the abscissa of all points, x^* is the result of maximum and minimum normalization. Through the maximum and minimum normalization, the error interference of the size of the layout area on the evaluation result can be eliminated.

The second metric is the *average distance between attribute communities* (ADBAC), which can measure the degree of distinguishability of attribute communities. The larger the distance between attribute communities, the greater the degree of distinction between attribute communities, and the layout results are presented more clearly. When the value of the metric tends to 0, it means that the different attribute communities are very close to each other. At this time, different attribute communities overlap with each other, and it is difficult to distinguish the existence of attribute communities from the layout results. The formula of ADBAC is as follows:

$$\text{ADBAC} = \frac{2}{m(m-1)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \sqrt{(\bar{x}_i - \bar{x}_j)^2 + (\bar{y}_i - \bar{y}_j)^2} \tag{47}$$

where i and j are the i-th and j-th attribute communities respectively, $\bar{x}_i$ is the abscissa of the centroid of the i-th attribute community, $\bar{x}_j$ is the ordinate. The formula of $(\bar{x}, \bar{y})$ is as follows:

$$(\bar{x}, \bar{y}) = \frac{\sum_i (x_i^*, y_i^*)}{n} \tag{48}$$

where (x_i^*, y_i^*) is the coordinate of the i-th node after the maximum and minimum normalization in a certain attribute community, and n is the number of nodes in this attribute community.

4. System

As shown in Figure 1, we introduce the general framework of our method. The input is a multivariate network $G_{in} = (V, E, A)$, where V is a node set, E is an edge set and $A = \{attr_1, ..., attr_m\}$ is the attributes for every node. The output is the layout of the whole of the multivariate network. The general framework consists of two major modules: data preprocessing and network layout module. After data preprocessing, every node has a label and it belongs to an attribute community, then we can use FDAC to start the network layout. In our method, the data preprocessing module mainly consists of three steps: *user select node of interest*, *user select attribute of interest* and *node attribute community distribution*.

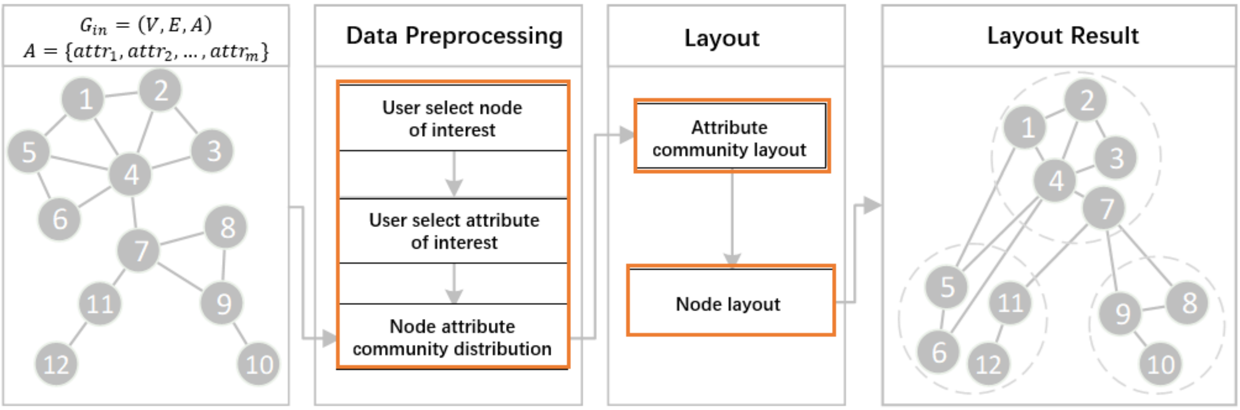
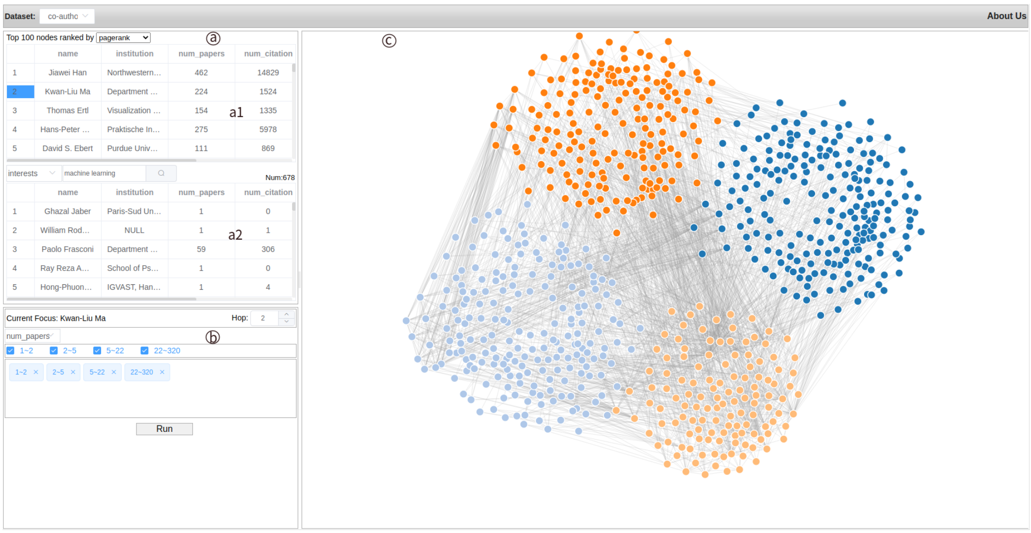

$G_{in} = (V, E, A)$
$A = \{attr_1, attr_2, \dots, attr_m\}$
1
2
3
4
5
6
7
8
9
10
11
12
Data Preprocessing
User select node of interest
User select attribute of interest
Node attribute community distribution
Layout
Attribute community layout
Node layout
Layout Result

Figure 1. The general framework of our approach.

On this basis, we developed a visualization system for applying the FDAC method. As shown in Figure 2, our system consists of three parts: (1) the **focus selection view** is used for the user to select the node of interest. (2) the **attribute selection view** is used for the user to select the attribute of interest and distribute the node attribute community (3) the **layout result view** is used to display multivariate network layouts.



Dataset: co-autho

Top 100 nodes ranked by pagerank

	name	institution	num_papers	num_citation
1	Jiawei Han	Northwestern…	462	14829
2	Kwan-Liu Ma	Department …	224	1524
3	Thomas Ertl	Visualization …	154	1335
4	Hans-Peter …	Praktische In…	275	5978
5	David S. Ebert	Purdue Univ…	111	869

interests machine learning Num:678

	name	institution	num_papers	num_citation
1	Ghazal Jaber	Paris-Sud Un…	1	0
2	William Rod…	NULL	1	1
3	Paolo Frasconi	Department …	59	306
4	Ray Reza A…	School of Ps…	1	0
5	Hong-Phuon…	IGVAST, Han…	1	4

Current Focus: Kwan-Liu Ma Hop: 2

num_papers

☑ 1~2 ☑ 2~5 ☑ 5~22 ☑ 22~320

1~2 × 2~5 × 5~22 × 22~320 ×

Figure 2. Visual system for user interaction with FDAC. (a) The focus selection view with (a1) the automatic screening by the system and (a2) the search view. (b) The attribute selection view. (c) The layout result view.

4.1. Focus Selection View

The main task is to select the nodes of interest based on the specified dataset. As shown in Figure 2a, we provide two options.

Automatic screening by the system. Automatic screening by the system As shown in a1 in Figure 2a, shows the top 100 results after sorting based on a certain metric. The ranking metrics include PageRank [47], the degree of the node, and some attribute values of the node. PageRank is automatically calculated by the system after selecting the dataset. This option is mainly provided to users who are not familiar with the dataset.

Search view. As shown in a2 in Figure 2a, shows the search results based on the attributes and attribute values, including the total number of nodes and the corresponding information of all nodes. After selecting the dataset, the system will count the attributes of the node and the corresponding attribute values for the user to select in the search box.

Through the above two options, the user can select the nodes of interest for subsequent network layout tasks. The result of the selection will be displayed in *Current Focus*, and the user also needs to select *hop*, which is used to determine the size of the subgraph used for the layout.

4.2. Attribute Selection View

As shown in Figure 2b, the user needs to select the attribute used for the layout and the attribute value under that attribute. For **numerical attributes** and **non-numerical attributes**, we have designed different attribute community distribution methods.

Numerical attributes. Taking the *age* attribute of each node in a social network as an example, users are not very interested in a specific age most of the time. Therefore, we provide an interval division method for numerical attributes. The user can select the expected number m of attribute communities to be divided. Knowing that n nodes are waiting to be laid out in the network, the formula for the expected number of nodes n_c in each community is as follows:

$$n_c = \frac{n}{m} \tag{49}$$

Sort the numerical attributes of all nodes, and divide the sequence with n_c as the length. When the sum of the number of nodes corresponding to consecutive numerical attributes reaches n_c, it is classified as an attribute community. The label of the attribute community is the mark from the minimum to the maximum value of the numerical attribute. Each node in the attribute community can be assigned this mark.

Non-numerical attributes. Take the *keyword* attribute of each node in the paper citation network as an example. A paper often contains multiple keywords. When users explore the keyword attributes, they often choose multiple keywords as the focus of attention. Using a method based on perfect matching, the dataset is traversed, and the node attributes that exactly match the user's selection are labeled. For nodes that match multiple attributes at the same time, they contain multiple tags and are assigned to a new attribute community.

Node attribute community distribution. Based on the choices, we need to assign attribute labels to nodes. For some nodes, its attribute value matches only one of the choices, then the label of this node is the attribute value. For other nodes, if its attribute value matches multiple attribute values, then the label of this node is the combination of the attribute values and this node will be assigned to a new attribute community where nodes have the same label also be in. For example, *keywords* attribute may contain multiple values, like *graph layout*, *visualization*, *data mining*, etc. the combination of the attribute values is graph *layout+visualization*.

4.3. Layout Result View

Based on the user's selection result, the data are obtained from the database and preprocessed, and then the force-directed layout algorithm based on attribute constraints is used for layout. As shown in Figure 2c, the layout result is drawn on the view.

When users are faced with a crowded network layout, the complex relationship formed by nodes and edges will interfere with users' exploration of the network structure. Therefore, we have realized the function of assisting users to explore in the visual system. As shown in Figure 3a, users first acquire the structure of the entire network when exploring the nodes they are interested in. As shown in Figure 3b, users can focus on a local region by zooming and exploring the connection relationship between nodes in the corresponding region. As shown in Figure 3c, we also support users to select and drag nodes, allowing users to make temporary changes to the layout of the network to better identify the connections between nodes.

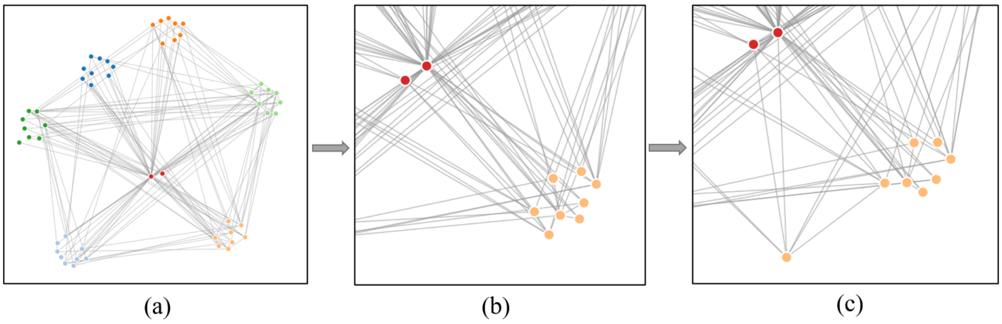


Figure 3. Illustration of User detail exploration. (**a**) global view; (**b**) local view; (**c**) local view after interaction.

5. Evaluation

We compared the three layout methods in terms of evaluation indicators (ADIAC and ADBAC), layout visualization, and edge length. The complexity of our method is analyzed. A user experiment is designed to verify the effectiveness of the method. We conduct all experiments on a laptop computer with Intel(R) Core(TM) i7-9750H CPU, 16 GB memory, and Windows 10 installed.

5.1. Dataset

This paper uses three graph datasets. The first one is the paper citation network in the field of visualization (VisPCNet), and the second one is the Researcher Collaboration Network (RCNet) composed of the paper collaborations of researchers in the four fields of visualization (Vis), data mining (DM), human–computer interaction (HCI), and machine learning (ML). Both are based on the open source dataset of Aminer. The last one is an office message network (OMNet) dataset based on VAST 2012 mini-challenge II network [48]

However, the comparison method in this paper, that is, the traditional force-guided layout method, performs very poorly on large-scale datasets. Therefore, this paper extracts four moderate-scale subgraphs from a relatively large-scale researcher cooperation network, which are named RCNet1, RCNet2, RCNet3, RCNet4, and OMNet1, respectively. The evaluation part of this article mainly uses these six datasets, and their specific information is shown in Table 1.

Table 1. Illustration of the datasets.

Dataset	Num of Nodes	Num of Edges	Attributes of Nodes
VisPCNet	513	1890	title, keywords, journal, year, number of citations
RCNet1	176	425	name, interests, number of papers, number of citations, H index
RCNet2	473	1623	name, interests, number of papers, number of citations, H index
RCNet3	450	1258	name, interests, number of papers, number of citations, H index
RCNet4	204	825	name, interests, number of papers, number of citations, H index
OMNet1	52	140	IP, name, working years message type, message number, device type

VisPCNet. The network consists of 513 visualization papers (nodes) and the citation relationships (edges) between them. Among them, each paper has five attributes: *title*, *keywords*, *journal*, *year*, *number of citations*.

RCNet1, RCNet2, RCNet3 and RCNet4. These four networks are composed of researchers in the field of visualization, data mining, human–computer interaction, and machine learning, as well as their cooperation in the field. The node represents the researcher. If two researchers have published a paper together, then there is an edge connection between them. In each network, researchers have five attributes: *name*, *interests*, *number of papers*, *number of citations*, *H index*.

OMNet1. The network consists of 52 devices(nodes) and 140 communications links(edges) between nodes. Each node has six attributes: *IP*, *user name*, *working years of user*, *message type*, *message number*, and *device type*.

5.2. Compared Methods

We compare our method with three layout methods, i.e., FR layout [10], LinLog layout [26], and ForceAtlas2 layout [27]. FR is the most classic force-directed model, and the subsequent various force-directed methods are improved versions of it. The LinLog model is a force-directed layout method based on an energy function, and its layout can present the structural community in the graph data. The nodes in the structural community may have the same attribute information. At this time, the LinLog model can present the attribute community, so we also use it as a comparison method. ForceAtlas2 is a force-directed method integrated in the well-known graph visualization tool Gephi. It uses various constraints to optimize the graph layout and can well reveal the structural information of the graph data.

5.3. Experiment

5.3.1. Quantitative Results

Tables 2 and 3 are the comparison of experimental results under ADIAC and ADBAC evaluation metrics, respectively. We compare the four methods on the four attributes of the above six datasets. Attribute 1 of VisPCNet is the keywords, attribute 2 is the journal, attribute 3 is the year, and attribute 4 is the number of citations; Attribute 1 of RCNet1, RCNet2, RCNet3, and RCNet4 is interests, attribute 2 is the number of papers, attribute 3 is the number of citations, and attribute 4 is the H index. Attribute 1 of OMNet1 is the working years, attribute 2 is the message type, attribute 3 is the message number, and attribute 4 is the device type.

It can be seen from Table 2 that under each attribute of each dataset, the ADIAC value of the method in this paper is much smaller than the ADIAC value of the other three methods. Compared with other methods, ADIAC in this paper is the lowest, about 0.1. This shows that our method has good layout stability, can obviously gather nodes with the same attributes together, and ensure the differentiation between nodes. It can be seen from Table 3 that under each attribute of each dataset, the ADBAC value of this method is much larger than that of other methods. The results of the comparison of evaluation metrics show that compared with other methods, the method in this paper can obtain clearer and more distinguishable results of attribute community layout.

Table 2. Average distance between nodes in attribute community.

	Attribute 1				**Attribute 2**			
	ours	FR [10]	linLog [26]	FA2 [27]	ours	FR [10]	linLog [26]	FA2 [27]
VisPCNet	**0.10**	0.27	0.54	0.53	**0.09**	0.29	0.52	0.52
RCNet1	**0.10**	0.38	0.54	0.5	**0.13**	0.42	0.54	0.53
RCNet2	**0.11**	0.44	0.52	0.54	**0.11**	0.40	0.51	0.50
RCNet3	**0.12**	0.35	0.53	0.53	**0.15**	0.36	0.54	0.54
RCNet4	**0.08**	0.28	0.57	0.52	**0.12**	0.37	0.51	0.53
OMNet1	**0.13**	0.31	0.50	0.49	**0.15**	0.38	0.48	0.50
	Attribute 3				**Attribute 4**			
	ours	FR	linLog	FA2	ours	FR	linLog	FA2
VisPCNet	**0.14**	0.33	0.52	0.54	**0.13**	0.34	0.52	0.53
RCNet1	**0.12**	0.42	0.55	0.47	**0.13**	0.42	0.52	0.54
RCNet2	**0.10**	0.41	0.52	0.54	**0.12**	0.38	0.53	0.52
RCNet3	**0.13**	0.35	0.54	0.53	**0.14**	0.34	0.51	0.51
RCNet4	**0.11**	0.38	0.51	0.53	**0.10**	0.39	0.54	0.55
OMNet1	**0.12**	0.30	0.49	0.52	**0.13**	0.37	0.47	0.52

Table 3. Average distance between attribute communities.

	Attribute 1				**Attribute 2**			
	ours	FR [10]	linLog [26]	FA2 [27]	ours	FR [10]	linLog [26]	FA2 [27]
VisPCNet	**0.38**	0.21	0.09	0.10	**0.37**	0.18	0.09	0.10
RCNet1	**0.45**	0.22	0.09	0.12	**0.53**	0.11	0.11	0.09
RCNet2	**0.42**	0.17	0.12	0.04	**0.40**	0.10	0.07	0.10
RCNet3	**0.46**	0.14	0.06	0.09	**0.42**	0.19	0.09	0.12
RCNet4	**0.49**	0.30	0.13	0.12	**0.58**	0.17	0.13	0.11
OMNet1	**0.45**	0.29	0.12	0.13	**0.44**	0.15	0.14	0.12
	Attribute 3				**Attribute 4**			
	ours	FR	linLog	FA2	ours	FR	linLog	FA2
VisPCNet	**0.53**	0.08	0.06	0.07	**0.45**	0.07	0.04	0.09
RCNet1	**0.52**	0.15	0.12	0.25	**0.51**	0.12	0.12	0.09
RCNet2	**0.40**	0.40	0.10	0.08	**0.42**	0.10	0.12	0.05
RCNet3	**0.46**	0.13	0.07	0.08	**0.46**	0.16	0.09	0.08
RCNet4	**0.54**	0.16	0.07	0.11	**0.50**	0.14	0.12	0.10
OMNet1	**0.50**	0.28	0.11	0.15	**0.48**	0.12	0.13	0.11

5.3.2. Qualitative Results

After visualizing the layout results, qualitatively comparing the layout results with human eyes can visually compare the differences between the methods. We selected 4 sets of layout results corresponding to VisPCNet and 1 set of layout results corresponding to OMNet1 to visualize. Among them, different colors represent different a ttribute communities. As shown in Figures 4–8, the comparison results of the layout under the four attributes are drawn separately. The results indicate that our method can draw the attribute community and display the attribute information well according to the attributes of the nodes.

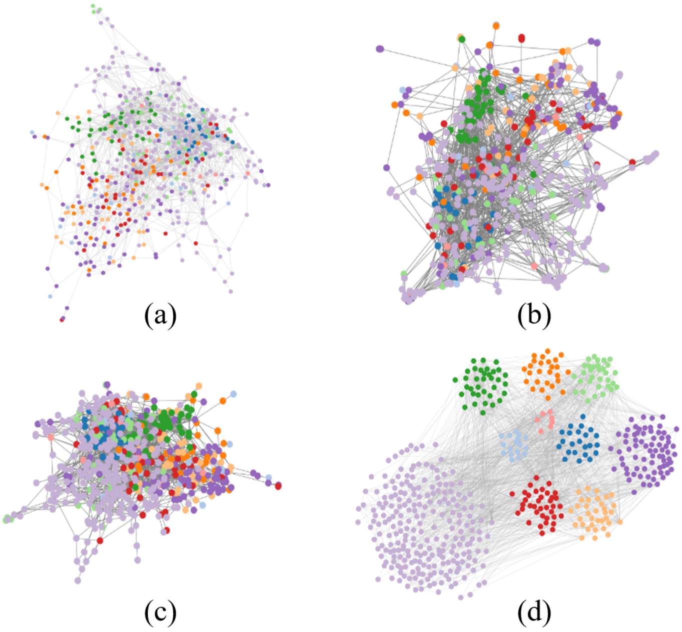


Figure 4. The result of attribute community layout under the attribute 'keywords'. (**a**) FR layout; (**b**) LinLog layout; (**c**) ForceAtlas2 layout; (**d**) layout of our method.

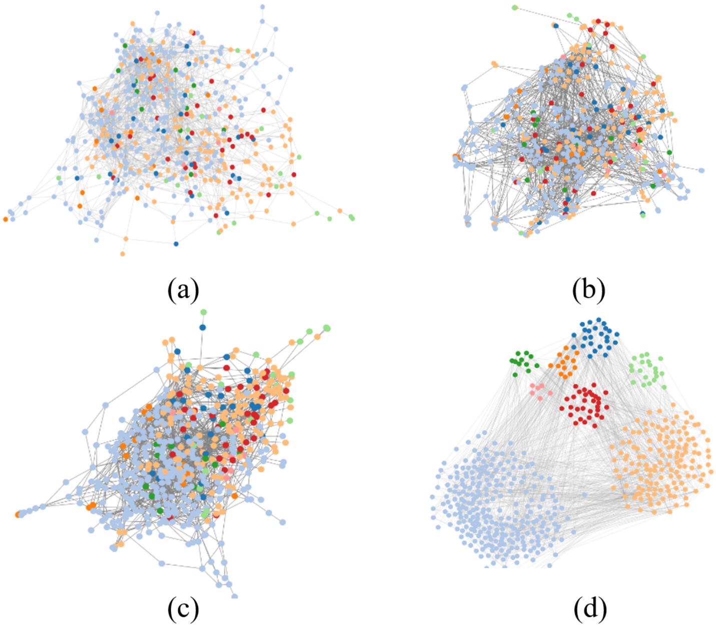


Figure 5. The result of attribute community layout under the attribute 'Journal'. (**a**) FR layout; (**b**) LinLog layout; (**c**) ForceAtlas2 layout; (**d**) layout of our method.

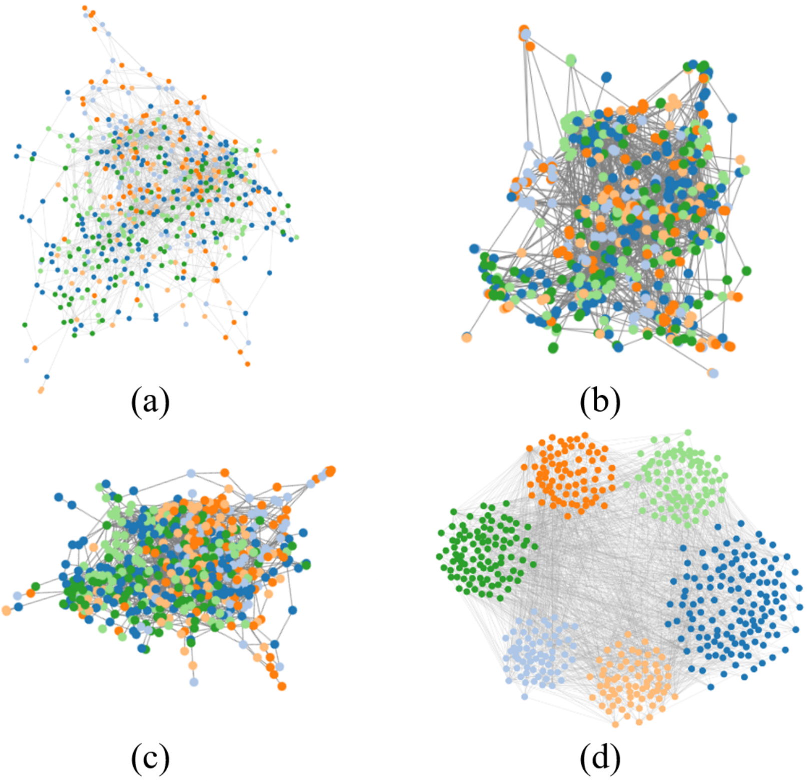


Figure 6. The result of attribute community layout under the attribute 'Year'. (**a**) FR layout; (**b**) LinLog layout; (**c**) ForceAtlas2 layout; (**d**) layout of our method.

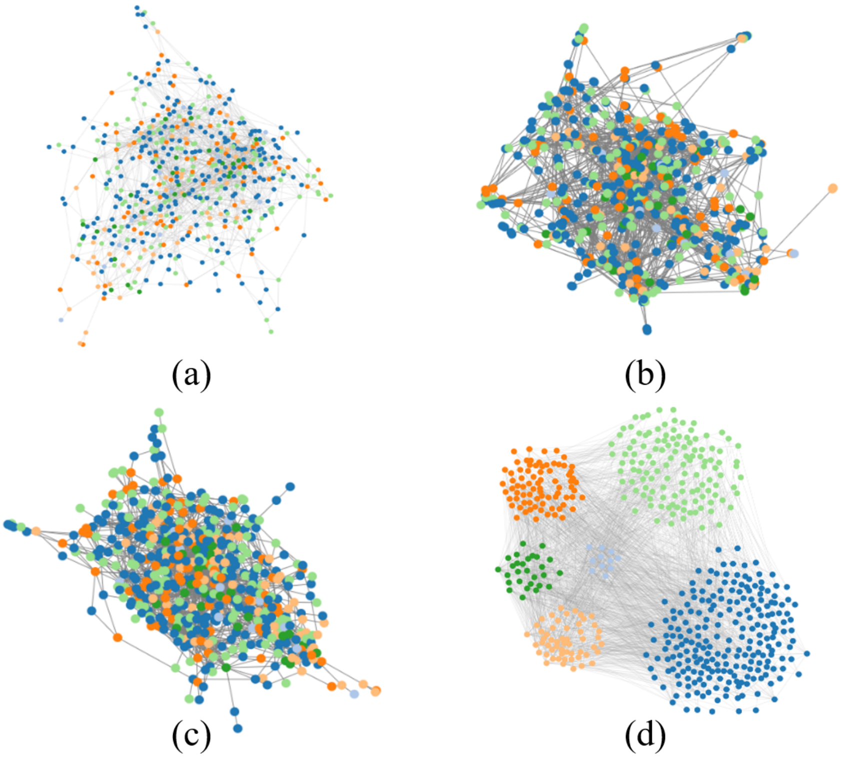


Figure 7. The result of attribute community layout under the attribute 'Number of Citations'. (**a**) FR layout; (**b**) LinLog layout; (**c**) ForceAtlas2 layout; (**d**) layout of our method.

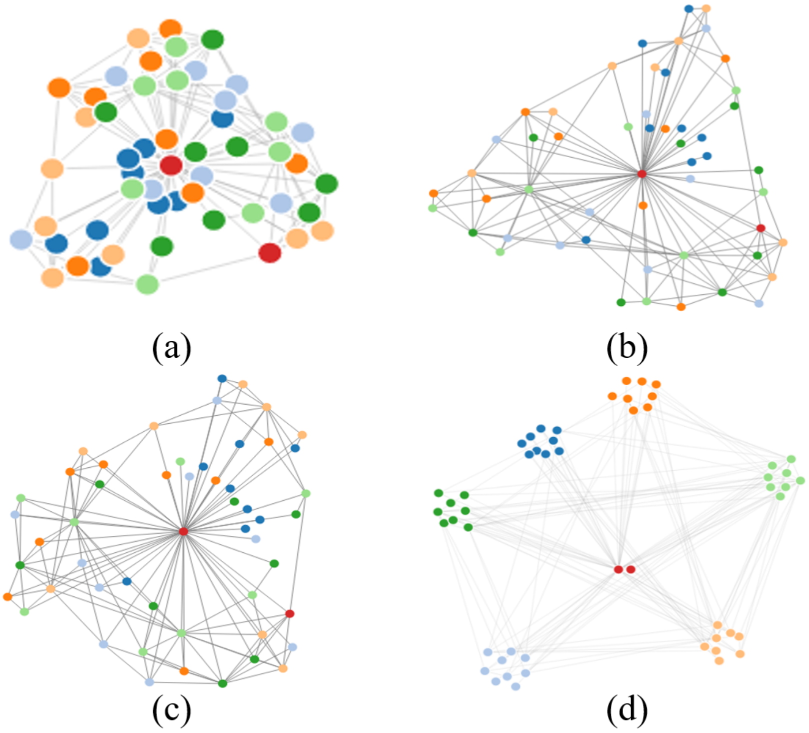


Figure 8. The result of attribute community layout under the attribute 'working years'. (**a**) FR layout; (**b**) LinLog layout; (**c**) ForceAtlas2 layout; (**d**) layout of our method.

5.4. Edge Length

We use two metrics: uniform edge length and maximum edge length to measure edges in the layout result. For the metric of uniform edge length, we calculate the standard deviation of the length of all edges. The experimental results are shown in Table 4. The smaller the standard deviation and maximum value, the more balanced the network layout. From the results, although our method does not reach the best level, the cost is acceptable compared to the attribute information representation it brings.

Table 4. Metric comparison of edge lengths.

	Standard Deviation of Edge Length				**Maximum of Edge Length**			
	Ours	**FR**	**linLog**	**FA2**	**Ours**	**FR**	**linLog**	**FA2**
VisPCNet	80	60	99	100	324	247	510	476
RCNet1	95	79	102	109	331	204	477	503
RCNet2	90	74	113	93	351	202	489	498
OMNet1	100	65	102	111	397	219	486	496

5.5. Complexity Analysis

5.5.1. Time Complexity

The time complexity of our algorithm includes attribute community layout $O(T|V_C|^2)$, node layout $(T|V|^2)$, where $|V|$ is the number of nodes, $|V_C|$ is the number of communities, and T is the number of iterations. Because the number of communities in the network is much smaller than the number of nodes, the total computational complexity of FDAC is derived as $(T|V|^2)$. Table 5 shows the running times of our method for different data sets.

Table 5. Running times of our method.

	Node Number	**Edge Number**	**Time**
OMNet1	52	140	0.18 s
RCNet1	176	425	0.56 s
VisPCNet	513	1890	3.8 s

5.5.2. Space Complexity

The data in the network consist of nodes and edges. Hence, the space complexity of our method is $O(|V| + |E| + |V_C| + |E_C|)$, where $|E|$ is the number of edges, and $|E_C|$ is the number of edges between communities.

5.6. User Study

To verify the effectiveness of our method in the representation of attribute information, we designed a user study. We defined three tasks:

- Community task: can users intuitively distinguish the number of attribute communities?
- Node task: Based on the nodes that the user is interested in, can users intuitively judge the proportion of nodes with the same attribute and nodes with different attributes in its neighbor nodes?
- Edge task: Based on the node that the user is interested in, can users intuitively explore other nodes with the same attribute connected to the structure along the edge?

We compared it with FR, linLog, and FA2. We invited five graduate students interested in the field of network layout as users. Users scored the experimental results of the four methods on multiple datasets according to the above tasks. The highest score is 5, which means it is easy to complete the task, and the lowest score is 1 which means it is difficult to complete the task. We averaged all the user scores. The experimental results are shown in Table 6.

Table 6. Average scores of users study.

	Ours	**FR**	**linLog**	**FA2**
task1	**4.6**	2.2	2.6	3.2
task2	**3.8**	1.6	2	1.5
task3	**3.4**	2.8	2	2.4

It can be seen from the experimental results that our method has obvious advantages in helping users to distinguish the number of attribute communities and judge node neighborhood relations. In terms of edge exploration, due to the influence of attribute community force, our method has more interference between edges in a larger graph, resulting in a lower overall score. In conclusion, our method can provide effective help for users to explore attribute information in the network.

6. Discussion

In this paper, we propose a method to visualize the layout of the graphs considering the attribute information of nodes. Compared with the traditional graph visualization layout method, the method in this paper can reflect the attribute information of the community graph data well. As in the above example, the clustering in the graph layout result can intuitively reflect the attribute information implicit in each attribute of the graph data. In the layout method that does not consider attributes, some nodes may be far away from most nodes with the same attributes due to the structural connection relationship. Although the layout uses colors and shapes for the logo, it still gives people a mixed feeling. On the contrary, the introduction of additional powerful constraints through attribute similarity can enable people to better confirm the attribute information of nodes.

In our method, attribute community force is applied to nodes to gather nodes with the same attribute into the same area. For some nodes, attribute constraints cause them to move away from neighboring nodes with different attributes, which creates additional edge crossings in the layout. This makes it difficult for users to explore the larger attribute community. On the one hand, we strengthen users' exploration ability through interaction design in the system. On the other hand, by defining community force strength, nodes closely connected with other attributes are more likely to be distributed on the periphery of the community to reduce crossover.

Our method automatically calculates the location of the attribute communities in space, allowing for a flexible layout. The idea of tree layout can be used in future work. Predetermine the location of each community layout based on the distribution of leaf nodes in the tree. For some nodes in the network, it may be connected to a large number of edges. We can also borrow PLANET's [37] idea of using polar coordinates to increase angles between different edges and enhance resolution.

Our method allows users to visualize the graph layout by selecting the node attributes they are interested in, to explore the attribute information implied by the graph data under this attribute. This kind of user interest-driven model makes the method in this paper flexible. If the node contains a non-numeric attribute, then each attribute value under the attribute is regarded as a kind of attribute community. If the node contains numerical attributes, users can discretize the attribute values according to their preferences, and each interval is an attribute community.

Adjusting the strength of attribute constraints has a significant impact on the results of the layout. When the attribute constraints are too tight, the nodes will closely surround the community center, resulting in a large number of overlapping nodes within the community. It is unable to distinguish the connection relationship between the same community node. When the attribute constraints are too loose, although the structural information between the same community nodes can be displayed well, the degree of discrimination between attribute communities will decrease.

7. Conclusions

In this paper, we propose a multivariate network layout method using force-directed with attribute constraints(FDAC), which can generate layout results with attribute information. We adopt a hierarchical layout strategy. By introducing attribute community force and multiple layout constraints in the force-directed layout, nodes with the same or similar attribute values are brought close to each other to form clusters. In this way, the attribute information contained in the network data can be intuitively conveyed to the user.

Aiming at the layout quality, this paper proposes two evaluation metrics, *average distance between nodes in attribute communities* and *average distance between attribute communities*. We compare three representative force-guided layout methods on six datasets to verify the effectiveness of this method.

The method in this paper generates large attribute communities when dealing with large networks, which is not conducive to users' exploration. We consider designing a multi-level attribute community layout in future work to reduce community size. There are a large number of edges between communities with different attributes. These edges interfere with the exploration of the network. We consider using edge bundling to simplify the structure. The time complexity of the algorithm can also be optimized. In the future, we consider completing both attribute community layout and node layout in one iteration. Our method automatically calculates the location of the attribute communities in space. We consider allowing users to decide where to place the attribute communities. We also consider using the tree layout structure to predetermine the location of each community and increase angles between different edges to enhance resolution.

Author Contributions: Conceptualization, Z.X., T.M., G.X. and Y.W.; data curation, Z.X. and T.M.; formal analysis, Z.X., T.M. and Y.W.; investigation, Z.X.; methodology, Z.X., T.M., G.X., Y.W. and D.L.; software, Z.X. and T.M.; supervision, G.X., Y.W. and D.L.; validation, Z.X., G.X. and Y.W.; visualization, all authors; writing—original draft, Z.X.; writing—review and editing, Z.X., G.X. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: http://www.arnetminer.org/data (accessed on 7 March 2022), http://www.vacommunity.org/VAST+Challenge+2012 (accessed on 7 March 2022).

Acknowledgments: The authors would like to thank all the colleagues for the fruitful discussions on Multivariate Network Layout.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dwyer, T. Scalable, versatile and simple constrained graph layout. *Comput. Graph. Forum* **2009**, *28*, 991–998.
2. Nobre, C.; Meyer, M.; Streit, M.; Lex, A. The state of the art in visualizing multivariate networks. *Comput. Graph. Forum* **2019**, *38*, 807–832.
3. Boz, H.A.; Bahrami, M.; Suhara, Y.; Bozkaya, B.; Balcısoy, S. An exploratory visual analytics tool for multivariate dynamic networks. In Proceedings of the EuroVis Workshop on Visual Analytics (EuroVA), Online, 25 May 2020; pp. 19–23.
4. Knittel, J.; Lalama, A.; Koch, S.; Ertl, T. Visual Neural Decomposition to Explain Multivariate Data Sets. *IEEE Trans. Vis. Comput. Graph.* **2020**, *27*, 1374–1384. [CrossRef] [PubMed]
5. Heer, J.; Boyd, D. Vizster: Visualizing online social networks. In Proceedings of the IEEE Symposium on Information Visualization, 2005, INFOVIS 2005. Minneapolis, MN, USA, 23–25 October 2005; pp. 32–39.
6. Stolte, C.; Tang, D.; Hanrahan, P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans. Vis. Comput. Graph.* **2002**, *8*, 52–65. [CrossRef]
7. Kreuseler, M.; López, N.; Schumann, H. A scalable framework for information visualization. In Proceedings of the IEEE Symposium on Information Visualization 2000, INFOVIS 2000, Salt Lake City, UT, USA, 9–10 October 2000; pp. 27–36.
8. Li, R.; Song, S.; Wu, Q.; John, L.K. Accelerating Force-directed Graph Layout with Processing-in-Memory Architecture. In Proceedings of the 2020 IEEE 27th International Conference on High Performance Computing, Data, and Analytics (HiPC), Pune, India, 16–19 October 2020; pp. 271–282.
9. Zhu, M.; Chen, W.; Hu, Y.; Hou, Y.; Liu, L.; Zhang, K. DRGraph: An Efficient Graph Layout Algorithm for Large-scale Graphs by Dimensionality Reduction. *IEEE Trans. Vis. Comput. Graph.* **2020**, *27*, 1666–1676. [CrossRef]
10. Fruchterman, T.M.; Reingold, E.M. Graph drawing by force-directed placement. *Softw. Pract. Exp.* **1991**, *21*, 1129–1164. [CrossRef]
11. Harel, D.; Koren, Y. A fast multi-scale method for drawing large graphs. In *International Symposium on Graph Drawing*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 183–196.
12. Hachul, S.; Jünger, M. Drawing large graphs with a potential-field-based multilevel algorithm. In *International Symposium on Graph Drawing*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 285–295.
13. Gajer, P.; Kobourov, S.G. Grip: Graph drawing with intelligent placement. In *International Symposium on Graph Drawing*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 222–228.
14. Harel, D.; Koren, Y. Graph drawing by high-dimensional embedding. In *International Symposium on Graph Drawing*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 207–219.
15. Kruiger, J.F.; Rauber, P.E.; Martins, R.M.; Kerren, A.; Kobourov, S.; Telea, A.C. Graph Layouts by t-SNE. *Comput. Graph. Forum* **2017**, *36*, 283–294.
16. Brandes, U.; Pich, C. Eigensolver methods for progressive multidimensional scaling of large data. In *International Symposium on Graph Drawing*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 42–53.
17. Ghoniem, M.; Fekete, J.D.; Castagliola, P. On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis. *Inf. Vis.* **2005**, *4*, 114–135. [CrossRef]
18. Henry, N.; Fekete, J.D.; McGuffin, M.J. NodeTrix: A hybrid visualization of social networks. *IEEE Trans. Vis. Comput. Graph.* **2007**, *13*, 1302–1309. [CrossRef]
19. Wattenberg, M. Visual exploration of multivariate graphs. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montréal, QC, Canada, 22–27 April 2006; pp. 811–819.
20. Shi, L.; Wang, C.; Wen, Z.; Qu, H.; Lin, C.; Liao, Q. 1.5 D egocentric dynamic network visualization. *IEEE Trans. Vis. Comput. Graph.* **2014**, *21*, 624–637. [CrossRef]
21. Cheong, S.H.; Si, Y.W. Force-directed algorithms for schematic drawings and placement: A survey. *Inf. Vis.* **2020**, *19*, 65–91. [CrossRef]
22. Huang, W.; Eades, P.; Hong, S.H.; Lin, C.C. Improving multiple aesthetics produces better graph drawings. *J. Vis. Lang. Comput.* **2013**, *24*, 262–272. [CrossRef]
23. Tutte, W.T. How to draw a graph. *Proc. Lond. Math. Soc.* **1963**, *3*, 743–767. [CrossRef]
24. Eades, P. A heuristic for graph drawing. *Congr. Numer.* **1984**, *42*, 149–160.
25. Kamada, T.; Kawai, S. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.* **1989**, *31*, 7–15. [CrossRef]
26. Noack, A. Energy models for graph clustering. *J. Graph Algorithms Appl.* **2007**, *11*, 453–480. [CrossRef]
27. Jacomy, M.; Venturini, T.; Heymann, S.; Bastian, M. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PLoS ONE* **2014**, *9*, e98679. [CrossRef]
28. Gansner, E.R.; Koren, Y.; North, S. Graph drawing by stress majorization. In *International Symposium on Graph Drawing*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 239–250.

29. Koren, Y.; Civril, A. The binary stress model for graph drawing. In *International Symposium on Graph Drawing*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 193–205.
30. Dwyer, T.; Koren, Y.; Marriott, K. Constrained graph layout by stress majorization and gradient projection. *Discret. Math.* **2009**, *309*, 1895–1908. [CrossRef]
31. Chen, L.; Buja, A. Stress functions for nonlinear dimension reduction, proximity analysis, and graph drawing. *J. Mach. Learn. Res.* **2013**, *14*, 1145.
32. Ko, Y.J.; Yen, H.C. Drawing clustered graphs using stress majorization and force-directed placements. In Proceedings of the 2016 20th International Conference Information Visualisation (IV), IEEE, Lisbon, Portugal, 19–22 July 2016; pp. 69–74.
33. Tamassia, R. *Handbook of Graph Drawing and Visualization*; CRC Press: Boca Raton, FL, USA, 2013.
34. Davidson, R.; Harel, D. Drawing graphs nicely using simulated annealing. *ACM Trans. Graph. (TOG)* **1996**, *15*, 301–331. [CrossRef]
35. Li, C.; Baciu, G.; Wang, Y. Module-based visualization of large-scale graph network data. *J. Vis.* **2017**, *20*, 205–215. [CrossRef]
36. Gemici, K.; Vashevko, A. Visualizing hierarchical social networks. *Socius* **2018**, *4*, 2378023118772982. [CrossRef]
37. Huang, G.; Li, Y.; Tan, X.; Tan, Y.; Lu, X. PLANET: A radial layout algorithm for network visualization. *Phys. Stat. Mech. Its Appl.* **2020**, *539*, 122948. [CrossRef]
38. Van den Elzen, S.; Van Wijk, J.J. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 2310–2319. [CrossRef]
39. Shen, Z.; Ma, K.L.; Eliassi-Rad, T. Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE Trans. Vis. Comput. Graph.* **2006**, *12*, 1427–1439. [CrossRef]
40. Shi, L.; Liao, Q.; Tong, H.; Hu, Y.; Wang, C.; Lin, C.; Qian, W. OnionGraph: Hierarchical topology+ attribute multivariate network visualization. *Vis. Inform.* **2020**, *4*, 43–57. [CrossRef]
41. Holten, D. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. Vis. Comput. Graph.* **2006**, *12*, 741–748. [CrossRef]
42. Zhou, H.; Xu, P.; Yuan, X.; Qu, H. Edge bundling in information visualization. *Tsinghua Sci. Technol.* **2013**, *18*, 145–156. [CrossRef]
43. Pienta, R.; Kahng, M.; Lin, Z.; Vreeken, J.; Talukdar, P.; Abello, J.; Parameswaran, G.; Chau, D.H. Facets: Adaptive local exploration of large graphs. In Proceedings of the 2017 SIAM International Conference on Data Mining. SIAM, Houston, Texas, USA, 27–29 April 2017; pp. 597–605.
44. Crnovrsanin, T.; Liao, I.; Wu, Y.; Ma, K.L. Visual recommendations for network navigation. *Comput. Graph. Forum* **2011**, *30*, 1081–1090.
45. Kairam, S.; Riche, N.H.; Drucker, S.; Fernandez, R.; Heer, J. Refinery: Visual exploration of large, heterogeneous networks through associative browsing. *Comput. Graph. Forum* **2015**, *34*, 301–310.
46. Laumond, A.; Melançon, G.; Pinaud, B. eDOI: Exploratory degree of interest exploration of multilayer networks based on user interest. In Proceedings of the VIS 2017, Poster Session, Phoenix, AZ, USA, 1–6 October 2017. [CrossRef]
47. Haveliwala, T.H. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Trans. Knowl. Data Eng.* **2003**, *15*, 784–796. [CrossRef]
48. IEEE VAST Challenge. 2012. Available online: http://www.vacommunity.org/VAST+Challenge+2012 (accessed on 5 June 2022).