Visualization of sub-network sets by iterative graph sampling from large scale networks

Namiko Toriyama
Ochanomizu University
Tokyo, Japan
toriyama.namiko@is.ocha.ac.jp

Mitsuo Yoshida
Toyohashi University of Technology
Aichi, Japan
yoshida@cs.tut.ac.jp

Takayuki Itoh
Ochanomizu University
Tokyo, Japan
itot@itolab.is.ocha.ac.jp

Abstract—Multi-layer network visualization techniques have been developed so that users can firstly overview the large-scale network and then explore the interesting parts of the data. Meanwhile, local features of the networks are often more interesting rather than their overall structures. It often happens with particular kinds of applications such as social networks. We developed a visualization technique for such types of large-scale networks. The technique iteratively applies a graph sampling algorithm to extract small-scale sub-networks from a large-scale network and then visualize the features of the sub-networks as hierarchically arranged icons. User-specified sub-networks are then visualized by applying our own graph visualization technique. Using networks generated from Twitter data, we actually visualize small-scale networks using the proposed method.

Index Terms—Visualization, Large-scale network, Sampling

I. Introduction

Various kinds of large-scale data have become readily available with the development of the information-oriented society. For example, human relationships on social networking services, transactions among companies, and data on infectious disease outbreaks are all large-scale data. It has been often difficult to understand the trends and phenomena hidden in the data while visualizing such large-scale datasets as they are. In order to make it easier for people to interpret the important meanings hidden in data, there have been many studies on the algorithms to extract essential features from input data as pre-processings before visualizing them.

Many information visualization techniques suppose to display the entire network first and then gradually focus on the local structure [Shn96] described as “Overview first, then zoom and filter, details on demand”. Many large-scale network visualization techniques also apply this operation procedure of firstly visualizing the entire network and then focusing on the local structure [Wu+18; HB05]. However, this approach is not always the most effective or efficient for all visualizations including large-scale network visualization. Specifically, we may not need to visualize the entire network first when we want to focus on the characteristic local features of the network. For example, in the case of visualizing social network data, structures and attributes of the local parts of the social network data that are more interesting to the data owner than the visualization of the entire network. Therefore, network visualization methods that extract characteristic localities from a large-scale network as the preprocessing and visualize the features of each locality have been studied recently.

Based on this context, we are developing a method to generate a large number of sub-networks by iterative network sampling processes and to visualize the sampled sub-networks as sets. Specifically, we extract a large number of sub-networks by iterative graph sampling from a large-scale network, and then represent the networks as small icons. Finally, we display the generated icons corresponding to the sub-networks as hierarchical data. Our current implementation generates pie charts based on the features of the sub-networks and displays them as icons. In addition, dimensionality reduction and clustering processes are applied to the features calculated from each sub-network, and the results are visualized as a hierarchical dataset. Furthermore, our implementation displays a sub-local network as a node-link diagram when a user clicks on the icons placed by these steps.

We introduce an example visualization using a real-world large-scale network dataset consisting of the official Twitter accounts of each political party and the accounts that followed their statements.

The remainder of this paper is organized as follows: in Section II, we introduce related works on graph sampling, visualization of icon representation, and visualization of sub-network Groups, respectively; Section III describes the flow of our proposed visualization method; Section IV describes the example of applying our method and discussion; and in Section V, we summarize the report and discuss future work.

II. Related Work

A. Graph Sampling

Graph sampling has been actively studied for a long time so that sub-networks can be appropriately extracted from large-scale networks. There have been several survey papers [HL13; LF06] on general-purpose graph sampling methods. We apply a general graph sampling method currently, but it is also important to apply the best sampling method for each application.

As an example of a network visualization method that utilizes graph sampling, Zhou et al. [Zho+20] presented a network visualization method that applies vectorizing and sampling processes. It expands the range of visual representation of the networks; however, the vectorization process may lose the context of the networks.

B. Iconic Representations of Networks

Two types of iconic representations of networks can be considered here: one is based on node features, and the other is based on network connection structures. Pie charts have been applied [MS16] for the representation of node features. Meanwhile, adjacent-matrix [Lek+18] and small multiples [Liu+15] have been applied for the representation of network connection structures.

Our current implementation simply represents the node features as pie charts. Various representations such as the above-mentioned ones can be applied to this study as future work.

C. Visualization of Sub-Network Groups

Iconization of important sub-networks is an effective approach to visualize large networks. Yoghourdjian et al. [Yog+18] presented a representation to iconize the global structure of protein binding networks. This method archived the linear computation time for the iconized visualization displaying the icons on homomorphic graphs. On the other hand, the methods often generate similar icons to the sub-graphs that have different contents but similar structures. It is often difficult to see the similarity between graphs with different contents but different structures while using this method. Chen et al. [Che+19] proposed a method to abstract and visualize the global structure of graphs based on user-defined sub-networks. In this method, the structural information of the nodes is vectorized and encoded in the nodes, which allows for various representations of the sub-network.

There have been several other network visualization methods. Our method differs from them since we focus on "classifying and displaying a large number of sub-networks represented by tens or hundreds of icons according to their features."

III. Processing Flow of the Presented Technique

In this section, we show the processing flow of the proposed method consisting of the following four steps:

[Step 1] Extract sub-graphs by iterative graph sampling.
[Step 2] Display icons of sub-networks.
[Step 3] Apply a clustering process and display the hierarchy of icons.
[Step 4] Place nodes of sub-networks specified by interactive operations.

We suppose the data structure of networks as defined by Itoh et al. [IK15]. Specifically, we suppose that a network G consists of a set of nodes $V = \{v_1, v_2, ...\}$ and a set of edges $E = \{e_1, e_2, ...\}$, where a node v has a multidimensional vector that serves as the feature value.

A. Graph Sampling

Graph sampling is applied to extract a large number of sub-networks from a large-scale network. Our current implementation applies a simple breadth-first-search-based graph sampling method among the methods introduced by Hu et al. [HL13]. Here, it is effective to specify the initial node of the search process as an key node that is linked to many other nodes. Our implementation searches all the data containing the link information and created a list of key nodes that have a large number of links as a preprocessing. We can extract the important network structures built around the highly key nodes. In the case of the Twitter network described later in the next section, we suppose that key nodes correspond to persons who have many followers.

Note that, the current sampling issues of "definition of key nodes" and "performance of search methods" depend on both general graph sampling problems and application-dependent requirements. In the case of the Twitter network discussed later in the next section, it would be effective to apply sampling methods specialized for social networks [Wan+11; JRM16].

B. Sub-network icons

We visualize a large number of sub-networks generated by iterative graph sampling as icons. Mainly, "connection structure of the sub-network" and "relationship between the size of features" are desirable information to be represented by icons in this study. However, our current implementation simply generates pie charts from the features of nodes. We selected pie charts as icons because their shapes are always squarely. Several icon representations other than pie charts have been presented as introduced in a survey paper [Wu+16] on social media visualization methods. For example, icon representation by adjacency matrix [Lek+18] is worth trying. Based on the above discussion, we would like to consider the following issues.

- Representation of feature values in a form other than a pie chart.
- Generating an icon that outlines the connection structure of a sub-network.

C. Hierarchical display of icons

Next, the presented method applies dimensionality reduction and clustering processes to the features of each icon and arranges the icons on the screen applying a hierarchical data visualization method. Our current implementation uses the average feature values of nodes of a sub-network as the feature value of the icon and applies the k-means method as the clustering method. Then, we apply the hierarchical data visualization method presented by Itoh et al. [Ito+06] to arrange the clusters of icons on the screen. Our implementation displays the hierarchical structure in the left half of the window, as shown in Fig. 3.

D. Node-link Diagrams for sub-network visualization

This section introduces how we visualize sub-networks as node-link diagrams. Our implementation displays a node-link diagram of a sub-network corresponding to an icon in the right half of the window when a user specifies one of a set of icons by a click operation,

Our method adopts the general graph visualization method displaying node-link diagram presented by Itoh et al. [IK15]. This method assumes not only a general graph structure but also that each node has a multidimensional vector corresponding to a feature. The method calculates the distance between nodes based on the following two conditions:

- Two nodes are close if they have many common nodes connected by links.
- Two nodes are close if they have small dissimilarities in feature values.

The method then performs a node clustering process with the calculated distances.

The distance between a pair of nodes is calculated by the following formula.

$$d = \propto d_{vec} + (1 - \propto) d_{adj}$$

Here, d_{vec} is the distance between two nodes based on the commonality of neighboring nodes, and d_{adj} is the distance between two nodes based on the dissimilarity of features. The node clustering process is performed based on the distances between the nodes calculated according to these two conditions.

The above-mentioned clustering method can separate key nodes that are connected to many other nodes from the cluster. This feature makes the visualization suitable for displaying the relationship between key nodes such as influencers and other groups in social media visualization. Meanwhile, Itoh et al.'s method can apply a general community extraction type of clustering methods where clusters are defined as areas with dense links.

After the clustering process described above, the method treats each cluster as a node and calculates the radius of the cluster according to the number of nodes belonging to the cluster. Then, the method places the graph on the screen by applying a force-directed model to the graph where the clusters are considered as nodes. After that, we apply Laplacian smoothing with the calculated radius, and arrange the nodes in the clusters radially. Fig. 1 illustrates these processes.

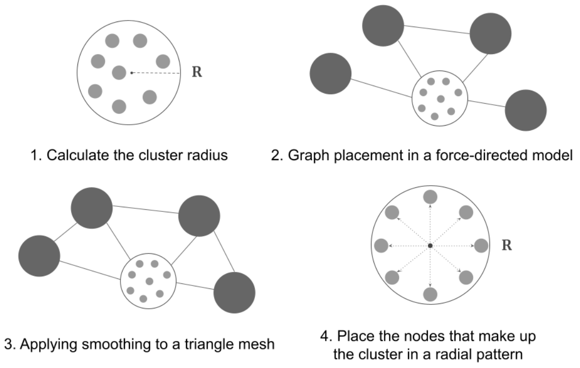

R
1. Calculate the cluster radius
2. Graph placement in a force-directed model
3. Applying smoothing to a triangle mesh
4. Place the nodes that make up the cluster in a radial pattern
R

Fig. 1. Processing flow of the network layout.

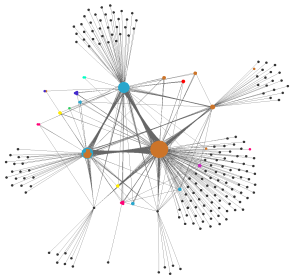


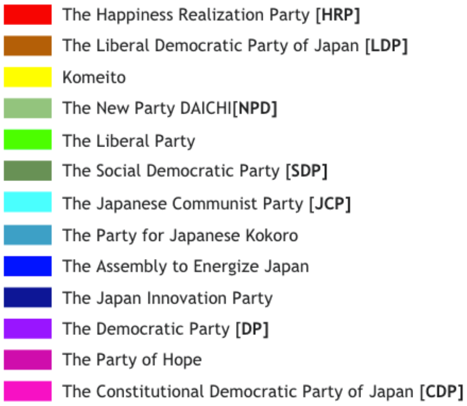
Fig. 2. An example of network placement [IK15] to a sub-network obtained by the graph sampling process. An key node is selected as the starting point of the graph sampling.

Fig. 2 shows an example of the graph sampling described in Section III-A and the resulting sub-network visualized by applying the above-mentioned method.

The clustering and network placement processed applied by our method require an average of 10 seconds for a network that has several thousand nodes. This computation time may be too large for interactive graph visualization. Instead, we need to implement off-line processes that calculate the positions of nodes on the screen in advance and read the results in response to interactive operations. In the current implementation, network clustering and network placement are implemented as separate software, and the node-link diagrams are saved as captured images. Since this implementation is undesirable in terms of interacting with the network visualization results, we will improve it so that it can be calculated and drawn when users specify the icons by click operations.

IV. Example and Discussion

In this section, we introduce a case study of visualizing a network consisting of the official Twitter accounts of political parties and the user accounts that followed the



The Happiness Realization Party [HRP]
The Liberal Democratic Party of Japan [LDP]
Komeito
The New Party DAICHI[NPD]
The Liberal Party
The Social Democratic Party [SDP]
The Japanese Communist Party [JCP]
The Party for Japanese Kokoro
The Assembly to Energize Japan
The Japan Innovation Party
The Democratic Party [DP]
The Party of Hope
The Constitutional Democratic Party of Japan [CDP]

Fig. 3. Correspondence between the colors of the pie chart and the political parties.

official accounts. This data was collected from September to October 2017, where nodes represent Twitter accounts and edges represent following relationships. The total number of nodes was 460,683, and the total number of edges was 1,425,696,164. The official accounts of the 13 political parties are the Happy Realization Party (HRP), the Liberal Democratic Party (LDP), Komeito, the New Party DAICHI, the Liberal Party, the Social Democratic Party (SDP), the Japanese Communist Party (JCP), the Party for Japanese Kokoro, the Japan Innovation Party, the Japan Innovation Party, the Democratic Party (DP), the Party of Hope, and the Constitutional Democratic Party of Japan (CDP).

Each account has a 13-dimensional feature vector based on the frequency of retweets the official account of each of the 13 parties. For each icon, the average of the corresponding feature values of each account is calculated as a 13-dimensional vector and used as a feature value. Fig. 3 shows the correspondence between the colors of each political party and the pie chart.

We iterated the process of extracting sub-networks from the Twitter data using the breadth-first search shown in Section III-A. The number of iterations is 100 in this experiment. We also limited the starting point of the breadth-first search to 84,043 accounts [YT18a; YT18b] that retweeted at least one tweet of an official account of a political party. We started the process from the node with the highest number of followers/followers as the initial position of the sampling. The second highest number of followers/followers node is chosen as the starting point in the second iteration. Similarly, the k-th highest number of followers/followers node is selected in the k-th iteration. Remark that the number of nodes in the sub-network is fixed as 5000 in this experiment. We would like to change the number of nodes in the sub-network according to the connection structure as future work.

Fig. 4 shows an example of visualization of hierarchically clustered icons corresponding to a set of sub-networks. The icons are divided into five clusters in this visualization example. The number of clusters can be changed dynamically, and the clustering can be rerun each time. A sub-network corresponding to a user-specified icon is displayed on the right side of the screen when a user has a mouse over one of the icons.

We can see that each cluster consists of a group of pie charts with a similar color scheme as displayed in the left side of Fig. 4. For example, in the lower-left corner, we can see many sub-networks consisting of users who spread the statements of the Constitutional Democratic Party (CDP) and the Japanese Communist Party (JCP). We can observe the distribution of the features of a group of sub-networks (in this case, a vector consisting of the number of times each party diffuses), and understand the structure of diffusion by looking at the sub-networks displayed on the right side of the screen. For example, we can observe the difference in structure between a “wide and shallow” sub-network with one extremely powerful person who spread tweets, and a “narrow and deep” sub-network where diffusion to a small number of people is repeated many times. Fig. 5 shows some examples. We can find various patterns in the diffusion of the tweets of the Constitutional Democratic Party and the Japanese Communist Party just by looking at these examples.

On the other hand, it is difficult to distinguish the sub-networks in a cluster because most of the icons that consist of the clusters displayed on the left side of Fig. 4 look very similar. This suggests our key issue to develop a design of complex icons that simultaneously represent not only features but also connection structures and a clustering method for such complex icons.

V. Conclusion and Future Work

This paper proposed a method for extracting a large number of sub-networks by iterative network sampling, representing them with icons, and visualizing the hierarchically clustered icons. The presented visualization method was applied to the network of the official Twitter accounts of Japanese political parties and their followers.

As future work, we would like to reconsider the implementation of graph sampling. We applied a general sampling method that is not customized to specific applications in the current implementation. In other words, the sampling results do not take into account the features of the application-specific data or the visualization process. Therefore, we would like to consider applying a sampling method specific to social networking service for the data introduced in this paper as future work. Another problem is that the number of nodes in the sub-network extracted by this method is fixed. We would like to review this problem and implement an additional mechanism to discontinue the search with an appropriate criterion so that we can extract appropriately sized interesting parts of the network.

Another issue is to reconsider the icon design and clustering. The current icon is simply a pie chart representing the average value of the feature values. It does not represent the network connection structure. Therefore, we would like to enable to create icons that represent both the network connection structure and feature values at the same time. In addition, we would like to extend the clustering method, for example, clustering by features first, and then clustering by connection structure again.

We would like to improve the implementation of subnetwork placement. Since clustering and network placement require a lot of computation time, the current implementation performs these calculations in advance as preprocessing and then saves the results of network placement as a captured image. Then, the captured image is displayed in response to a click operation on demand. We would like to improve the implementation because this is also undesirable in terms of interactive operation.

After the above improvements, we would like to discuss with experts in the data domain (e.g., experts in Twitter data analysis) to advance our research in a direction that is suitable as an analysis tool for large networks. Also, we would like to extend the method to other networks in addition to Twitter's political party networks.

References

[Shn96] B. Shneiderman. "The eyes have it: a task by data type taxonomy for information visualizations." In: IEEE Symposium on Visual Languages (1996), pp. 336–343.

[HB05] J. Heer and D. Boyd. "Vizster: visualizing online social networks." In: IEEE Symposium on Information Visualization (2005), pp. 32–39.

[Ito+06] T. Itoh et al. "Hierarchical Visualization of Network Intrusion Detection Data in the IP Address Space". In: IEEE Computer Graphics and Applications 26.2 (2006), pp. 40–47.

[LF06] J. Leskovec and C. Faloutsos. "Sampling from Large Graphs." In: Association for Computing Machinery (2006), pp. 631–636.

[Wan+11] T. Wang et al. "Understanding Graph Sampling Algorithms for Social Network Analysis." In: International Conference on Distributed Computer Systems Workshop (2011), pp. 123–128.

[HL13] P. Hu and W. C. Lau. "A Survey and Taxonomy of Graph Sampling, arXiv preprint." In: 2013.

[IK15] T. Itoh and K. Klein. "Key-node-Separated Graph Clustering and Layout for Human Relationship Graph Visualization." In: IEEE Transactions on Visualization and Computer Graphics 35.6 (2015), pp. 30–40.

[Liu+15] Xiaotong Liu et al. "Correlated Multiples: Spatially Coherent Small Multiples With Constrained Multi-Dimensional Scaling." In: Computer Graphics Forum 37 (2015). doi: 10.1111/cgf.12526.

[JRM16] Z. S. Jalali, A. Rezvanian, and M. R. Meybodi. "Social Network Sampling Using Spanning Trees". In: International Journal of Modern Physics C 27.5 (2016).

[MS16] Christopher Steven Marcum and David R. Schaefer. "Save Room for Pie: Adding Pie Charts to Network Visualizations in R with Statnet and Plotrix." In: SocArXiv (2016).

[Wu+16] Y. Wu et al. "A Survey on Visual Analytics of Social Media Data, IEEE Transactions on Multimedia". In: IEEE Transactions on Multimedia 28.11 (2016), pp. 2135–2148.

[Lek+18] Fritz Lekschas et al. "HiPiler: Visual Exploration of Large Genome Interaction Matrices with Interactive Small Multiples." In: IEEE Transactions on Visualization and Computer Graphics 24.1 (2018), pp. 522–531.

[Wu+18] Y. Wu et al. "StreamExplorer: A Multi-Stage System for Visually Exploring Events in Social Streams." In: IEEE transactions on visualization and computer graphics 24.10 (2018), pp. 2758–2772.

[Yog+18] V. Yoghourdjian et al. "Graph Thumbnails: Identifying and Comparing Multiple Graphs at a Glance." In: IEEE Transactions on Visualization and Computer Graphics 24.12 (2018), pp. 3081–3095.

[YT18a] M. Yoshida and F. Toriumi. "Analysis of Political Party Twitter Accounts' Retweeters during Japan's 2017 Election". In: IEEE/WIC/ACM International Conference on Web Intelligence (WI) (2018), pp. 736–739.

[YT18b] M. Yoshida and F. Toriumi. "Information Diffusion Power of Political Party Twitter Accounts During Japan's 2017 Election". In: 10th International Conference on Social Informatics (SocInfo 2018) (2018), pp. 334–342.

[Che+19] W. Chen et al. "Structure-Based Suggestive Exploration: A New Approach for Effective Exploration of Large Networks". In: IEEE Transactions on Visualization and Computer Graphics 25.1 (2019), pp. 555–565. doi: 10.1109/TVCG.2018.2865139.

[Zho+20] Zhiguang Zhou et al. "Context-aware Sampling of Large Networks via Graph Representation Learning." In: IEEE Trans Vis Comput Graph 72.3 (2020), pp. 213–218.

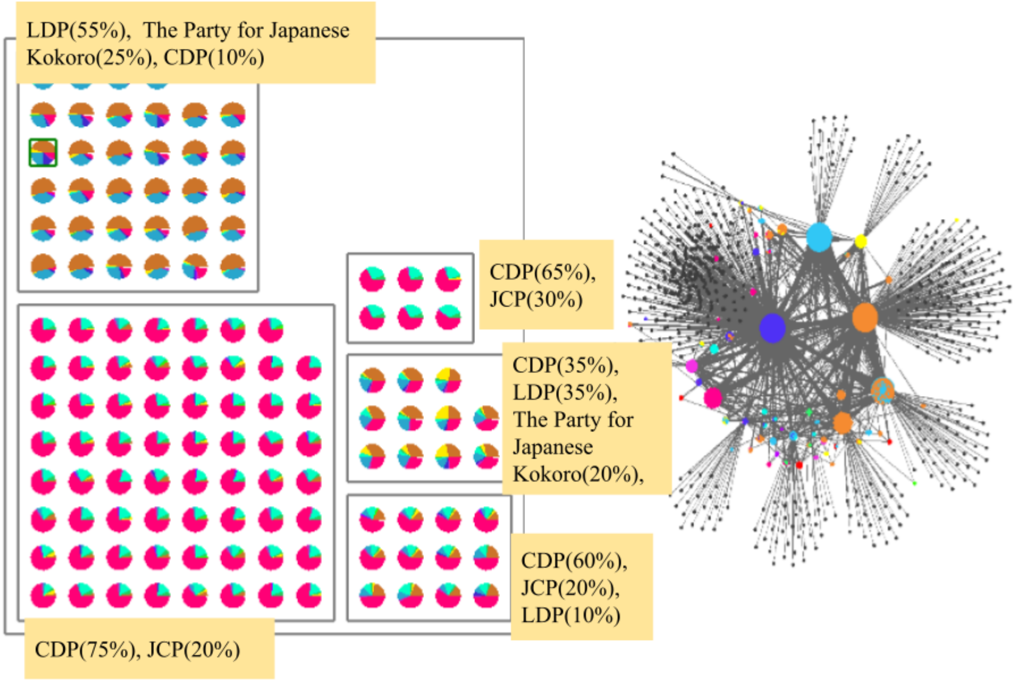

LDP(55%), The Party for Japanese Kokoro(25%), CDP(10%)
CDP(65%), JCP(30%)
CDP(35%), LDP(35%), The Party for Japanese Kokoro(20%),
CDP(60%), JCP(20%), LDP(10%)
CDP(75%), JCP(20%)

Fig. 4. Example visualization. On the left side of the screen, a set of sub-networks is displayed as icons (currently a pie chart) and classified into six clusters. When a user specifies one of the pie charts by a mouse over operation, the corresponding sub-network is displayed on the right side of the screen. Remark that the network layout is not performed in real time, but an image captured as the result of the network placement is displayed.

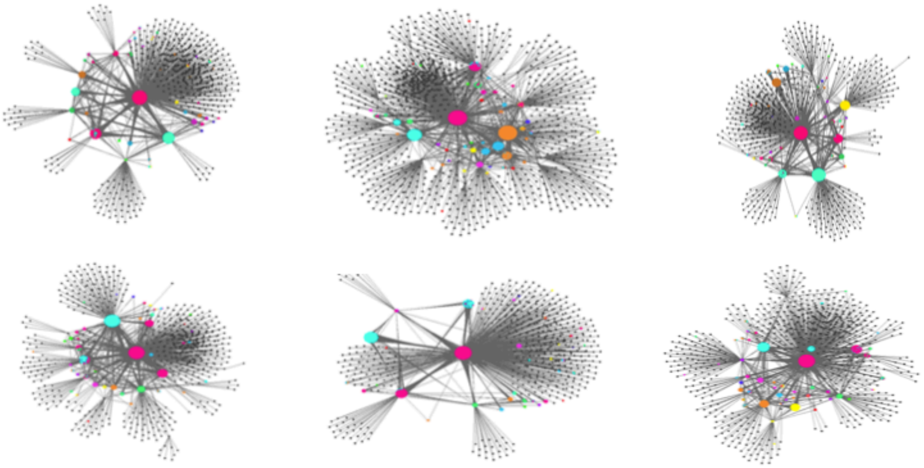


Fig. 5. Six sub-networks belonging to the cluster labeled "Constitutional Democratic Party (approx. 75%), Communist Party (approx. 20%)" shown in Fig. 4. Each sub-network in this figure has its own connection structure.