

# Forest expression of networks and their applications

Yipeng Wang and Peihua Qiu

Department of Biostatistics, University of Florida, USA

October 9, 2025

## Abstract

Complex networks with hierarchical relationships and sparse structures among nodes are pervasive in both social and physical sciences. Examples include hierarchical power distribution networks and tree-like structures centered around hub nodes in disease transmission networks. A common challenge in such studies is to visualize or analyze the underlying hierarchy. Existing analytical tools are limited, often relying on ego networks or handcrafted features. In this paper, we propose a novel framework that represents each connected component of a network as a tree, and the entire network as a forest. This forest-based representation enables intuitive 3D visualizations and introduces new structural features that facilitate meaningful network comparisons. For dynamic network systems, we further develop pixel-based visualizations of these features to provide scalable overviews. Numerical studies across a range of cases demonstrate that the proposed approach offers powerful graphical and analytical tools for network analysis.

**Key words:** Dynamic networks; Features; Graph visualization; Network comparison; Tree structure

## 1 Introduction

Networks are widely used as a universal framework for describing complex systems through pairwise interactions among their entities (Newman, 2018; Chen et al., 2021; Songdechakruiwut and Chung, 2023). For example, a mobility network can represent individual trips among 401 counties in Germany over the course of a week (Schlosser et al., 2020). Hierarchical relationships are pervasive in both natural and artificial systems (Yu and Gerstein, 2006; Corominas-Murtra et al., 2013). In power systems, for instance, a radial distribution network exhibits five hierarchical levels, ranging from substations to smart meters (Alghuried and Moghaddass, 2021). Similarly, tree-like structures originating from hub nodes are commonly observed in social networks, playing a critical role in information dissemination and disease transmission (Kitsak et al., 2010; Abu-Ata and Dragan, 2016; Shu et al., 2020). Dynamic networks are often employed to capture structural changes in evolving systems (Li

et al., 2015; Sekara et al., 2016). For instance, communications among 27,436 computers in an enterprise network running Microsoft Windows over 89 days can be represented as a dynamic network (Lee et al., 2022).

Network visualization is a powerful tool for exploratory graph analysis, as it helps to depict and interpret underlying structures (Von Landesberger et al., 2011; Gray et al., 2023). Common approaches include matrix diagrams and 2D node-link diagrams, but these methods often struggle to reveal hierarchical relationships (Rubinov and Sporns, 2010; Yoghourdjian et al., 2020). The visualization of dynamic networks has been widely discussed in the literature (Moody et al., 2005; Beck et al., 2017; Linhares et al., 2022). Typically, dynamic networks are displayed either as animated diagrams or as static charts aligned along a timeline. However, animations can be unsuitable for large-scale systems, as they demand high cognitive effort to track and compare changes over time (Tversky et al., 2002). Therefore, developing effective visualizations of network sequences is essential for capturing the longitudinal patterns of dynamic networks.

In many research domains, it is often necessary to compare complex networks (Emmert-Streib et al., 2016). A common strategy involves combining network visualizations with extracted structural features. Such features are typically derived from network models or task-specific heuristics—for example, using the total number of nodes to characterize network size. Statistical methods can then be applied to quantify network similarity based on these features (Pržulj, 2007; Lee et al., 2022; Bravo-Hermsdorff et al., 2023). When two networks share an identical set of nodes, statistical comparisons can leverage a one-to-one mapping between them (Ghoshdastidar et al., 2020). In practice, however, networks often differ in their node sets. For instance, in social networks, new members may join while existing members leave over time (Panzarasa et al., 2009). The aforementioned comparison methods cannot be applied in such cases.

This paper introduces a forest expression for networks, motivated by the observation that connected components are often central to characterizing network topology (Von Lan-

desberger et al., 2009; Schieber et al., 2017; Wang et al., 2024). In this framework, each connected component is represented as a tree, which is visualized through a 3D graph. The entire network is then expressed as a forest composed of these trees. From this representation, we extract a set of features that describe the network structure from multiple perspectives. Formal definitions of these features are provided in the next section. Unlike an ego network, which consists of a central node and its immediate neighbors (Mcauley and Leskovec, 2014), a tree in the forest expression captures hierarchical relationships within a connected component. The resulting 3D graphs and extracted features enable both the identification and quantification of structural dissimilarities across networks. In addition, pixel-based visualizations of the extracted features provide a scalable overview for dynamic networks. The effectiveness of the proposed forest expression will be demonstrated using both real and simulated networks in the following sections.

It is worth noting that several existing methods decompose networks into tree-like structures. One well-known approach is tree decomposition, which maps a graph into a tree where each node of the tree, called a bag, represents a subset of nodes in the graph. This decomposition satisfies three properties: (i) all nodes of the graph are covered, (ii) every edge is included in at least one bag, and (iii) for each node, the bags containing it form a connected subtree (Halin, 1976). For disconnected graphs, each connected component is decomposed separately into a tree that meets these requirements. Consequently, a tree decomposition yields a tree (or forest) of bags, but it does not explicitly capture the hierarchical structure of the original network. Another common approach is the spanning tree of a connected graph, defined as a cycle-free subgraph that connects all nodes of a graph. Pathfinding algorithms such as Dijkstra’s algorithm construct spanning trees intermediately for individual connected graphs (Hart et al., 1968; Dijkstra, 2022). Extending this idea, a spanning forest is formed by a collection of spanning trees, each corresponding to a connected component of the original network. While a spanning forest covers all nodes of the network without forming cycles, it cannot adequately reveal the hierarchical relationships inherent in a network

system. A related but distinct line of work is community detection, where the goal is to identify groups of nodes that are more densely connected to each other than to the rest of the network (Zhao et al., 2012; Rossetti and Cazabet, 2018). Hierarchical community detection further reveals nested community structures, uncovering organization at multiple levels (Lyzinski et al., 2016; Li et al., 2022). However, while this approach focuses on uncovering intrinsic organization, relatively little attention has been paid to its use for visualization and systematic comparison of networks (Vidaurre et al., 2017).

## 2 Forest Expression

A network  $G$  consists of a set of nodes (or vertices) denoted as  $\mathcal{V}(G)$  and a set of edges (or links) denoted as  $\mathcal{E}(G)$ . The number of nodes is denoted as  $n_v = |\mathcal{V}(G)|$ , and the sum of edge weights is denoted as  $s_e = |\mathcal{E}(G)|$ . In the literature, a network  $G$  is often described by its  $n_v \times n_v$  adjacency matrix  $\mathbf{A}$  whose  $(i, j)$ th element  $a_{ij}$  denotes the “weight” of the edge between the two nodes in the  $(i, j)$ th pair, for  $1 \leq i, j \leq n_v$ , with  $a_{ij} = 0$  implying no edge between the two related nodes. In this article, we focus on undirected networks without self-loops. Thus,  $\mathbf{A}$  is a symmetric matrix with  $a_{ii} = 0$  for each  $i$ . Although the proposed methods can be extended easily to cases when  $a_{ij}$  are non-negative binary or real numbers, it is assumed that the weights  $a_{ij}$  are non-negative integers. Specifically,  $a_{ij}$  can be interpreted as the number of edges between the two nodes in the  $(i, j)$ th pair, and we have  $s_e = (\sum_{i=1}^{n_v} \sum_{j=1}^{n_v} a_{ij})/2$ .

Let the number of connected components in a network  $G$  be denoted by  $n_c$ . For the  $i$ th connected component  $C_i$ ,  $i = 1, 2, \dots, n_c$ , the central node can be selected as the root of its hierarchical structure when node attributes are available to indicate hierarchy. However, such attributes are often unavailable in network analysis. To address this issue, we propose a snowfall algorithm to identify the central node.

For a connected component  $C_i$ , the *diameter* is defined as the length of the longest

geodesic path between any pair of nodes. Specifically, the diameter of  $C_i$  is given by

$$l_i = \max_{u,v \in C_i} \text{dis}(u, v),$$

where  $\text{dis}(u, v)$  denotes the length of the shortest path between nodes  $u$  and  $v$ . Thus, an isolated node has a diameter of 0. Within  $C_i$ , the  $d$ th neighborhood of a node  $u$  is defined as the set of nodes

$$\{v \mid \text{dis}_{v \in C_i}(u, v) \leq d\},$$

and the corresponding subnetwork based on this neighborhood is denoted by  $O_u^d$ . The proposed snowfall algorithm for finding the central node of the  $i$ th connected component  $C_i$  in  $G$  is presented in Algorithm 1. For an isolated node, the central node is itself.

Let  $d_s$  denote the length of the longest geodesic path between the central node  $\mathbf{u}$  and any other node within the neighborhood where the algorithm terminates. Among all subnetworks  $\{O_v^{d_s} \mid v \in C_i\}$ , the one containing the central node has the largest number of nodes and/or the largest total edge weight. The identified central node  $\mathbf{u}$  is then designated as the *root node*, and the corresponding tree representation of  $C_i$  has  $l_i$  layers. Specifically, the  $k$ th layer consists of nodes in

$$\mathcal{V}_k^{(i)} = \{v \mid \text{dis}_{v \in C_i}(\mathbf{u}, v) = k\},$$

for  $k = 1, 2, \dots, l_i$ .

For each connected component  $C_i$ , let  $n_v^{(i)}$  and  $n_e^{(i)}$  denote the number of nodes and edges, respectively. In the worst-case scenario—when all nodes have the same degree—the time complexity of Algorithm 1 is

$$O\left((n_v^{(i)})^3 + (n_v^{(i)})^2 n_e^{(i)}\right).$$

In practice, the actual runtime is typically much lower due to variation in node degrees.

Unlike traditional tree structures (Newman, 2018; Behr et al., 2020), in our representa-

tion, nodes within the same layer can also be connected, not just those in adjacent layers. The forest representation of network  $G$  is then obtained by describing each connected component of  $G$  as a tree.

---

**Algorithm 1** Snowfall

---

```

1: procedure SNOWFALL( $C_i$ ) ▷  $i = 1, 2, \dots, n_c$ 
2:    $d \leftarrow 1$ 
3:    $l_i \leftarrow \max_{u,v \in C_i} \text{dis}(u, v)$ 
4:   if  $l_i \neq 0$  then ▷ If  $l_i = 0$ ,  $C_i$  is the isolated node
5:     while  $d \leq l_i$  do
6:       if  $d = 1$  then
7:          $\mathbb{V} \leftarrow \{u \mid |\mathcal{V}(O_u^d)| = \max_{v \in C_i} |\mathcal{V}(O_v^d)|\}$ 
8:         if  $|\mathbb{V}| = 1$  then ▷ Only one node  $u$  in  $\mathbb{V}$ 
9:            $d_s \leftarrow d$ ,  $u \leftarrow u$ 
10:          return  $u, d_s$ 
11:          break
12:        else
13:           $\mathbb{E} \leftarrow \{u \mid |\mathcal{E}(O_u^d)| = \max_{v \in \mathbb{V}} |\mathcal{E}(O_v^d)|\}$ 
14:          if  $|\mathbb{E}| = 1$  then
15:             $d_s \leftarrow d$ ,  $u \leftarrow u$ 
16:            return  $u, d_s$ 
17:            break
18:          else
19:             $d \leftarrow d + 1$ 
20:          end if
21:        end if
22:      end if
23:      if  $d \geq 2$  then
24:         $\mathbb{V} \leftarrow \{u \mid |\mathcal{V}(O_u^d)| = \max_{v \in \mathbb{E}} |\mathcal{V}(O_v^d)|\}$ 
25:        if  $|\mathbb{V}| = 1$  then
26:           $d_s \leftarrow d$ ,  $u \leftarrow u$ 
27:          return  $u, d_s$ 
28:          break
29:        else
30:           $\mathbb{E} \leftarrow \{u \mid |\mathcal{E}(O_u^d)| = \max_{v \in \mathbb{V}} |\mathcal{E}(O_v^d)|\}$ 
31:          if  $|\mathbb{E}| = 1$  then
32:             $d_s \leftarrow d$ ,  $u \leftarrow u$ 
33:            return  $u, d_s$ 
34:            break
35:          else
36:             $d \leftarrow d + 1$ 
37:          end if
38:        end if
39:      end if
40:    end while
41:    if  $d = l_i + 1$  then
42:       $u \sim \mathbb{E}$  ▷ Randomly choose a node from  $\mathbb{E}$ 
43:       $d_s \leftarrow l_i$ 
44:      return  $u, d_s$ 
45:    end if
46:  end if
47: end procedure

```

---

For the  $i$ th tree (see Figure 1E for an example), we suggest using the following features to describe its structure. The number of nodes  $n_v^{(i)}$  is used to measure the tree size. The sum of edge weights, denoted by  $s_e^{(i)}$ , measures interactions among the nodes within the tree. The number of layers, denoted by  $n_l^{(i)}$ , measures the tree height. To measure the cohesion of

the tree, the generalized clustering coefficient originally proposed in Opsahl and Panzarasa (2009) is used, which is defined to be

$$\rho^{(i)} = \frac{\sum_{\tau_c \in C_i} \omega}{\sum_{\tau \in C_i} \omega},$$

where  $\tau$  denotes a triplet,  $\tau_c$  denotes a closed triplet, and  $\omega$  is the geometric mean of weighted edges in the triplet. If all triplets in the  $i$ th tree are closed, then  $\rho^{(i)} = 1$ . If there are no closed triplets in the  $i$ th tree, then  $\rho^{(i)} = 0$ . Thus,  $\rho^{(i)} \in [0, 1]$ .

Within the  $k$ th layer  $\mathcal{V}_k^{(i)}$  of the  $i$ th tree, the number of nodes  $n_{vk}^{(i)}$  is used to describe the layer size, the sum of edge weights  $s_{ek}^{(i)}$  is used to measure the amount of communications, and the ratio  $r_k^{(i)} = 2m_{ek}^{(i)}/[n_{vk}^{(i)}(n_{vk}^{(i)} - 1)]$  is used to describe the interconnected structure, where  $m_{ek}^{(i)} = [\sum_{u \in \mathcal{V}_k^{(i)}} \sum_{v \in \mathcal{V}_k^{(i)}} I(a_{uv} > 0)]/2$ . If  $n_{vk}^{(i)}$  is fixed, then  $r_k^{(i)}$  would increase as more nodes in the  $k$ th layer are connected to each other. Then, the within-layer features for the  $i$ th tree is summarized in the following matrix:

$$\mathbf{W}^{(i)} = \begin{pmatrix} n_{v1}^{(i)} & s_{e1}^{(i)} & r_1^{(i)} \\ \vdots & \vdots & \vdots \\ n_{vl_i}^{(i)} & s_{el_i}^{(i)} & r_{l_i}^{(i)} \end{pmatrix}.$$

To describe the connection structure between the  $(k-1)$ th and  $k$ th layers (including the central node and the 1st layer), we report the sum of edge weights  $s_{e(k-1)k}^{(i)}$  and the ratio  $r_{(k-1)k}^{(i)} = m_{e(k-1)k}^{(i)}/[n_{v(k-1)}^{(i)}n_{vk}^{(i)}]$ . Specifically,  $s_{e01}^{(i)}$  is the sum of edge weights between the central node and the nodes in the 1st layer,  $m_{e(k-1)k}^{(i)} = \sum_{u \in \mathcal{V}_{k-1}^{(i)}} \sum_{v \in \mathcal{V}_k^{(i)}} I(a_{uv} > 0)$ , and  $r_{01}^{(i)} = 1$  because the central node connects to all nodes in the 1st layer. Then, the between-layer features for the  $i$ th tree is summarized in the following matrix:

$$\mathbf{B}^{(i)} = \begin{pmatrix} s_{e01}^{(i)} & r_{01}^{(i)} \\ \vdots & \vdots \\ s_{e(l_i-1)l_i}^{(i)} & r_{(l_i-1)l_i}^{(i)} \end{pmatrix}.$$

It can be checked that the combination of  $n_v^{(i)}$ ,  $s_e^{(i)}$ ,  $n_l^{(i)}$ ,  $\rho^{(i)}$ ,  $\mathbf{W}^{(i)}$ , and  $\mathbf{B}^{(i)}$  would be sensitive to all seven typical structural changes shown in Figure 9 in the Appendix.

When  $n_c > 1$ , multiple trees exist in the forest representation of a network (cf., Figure 2A). Although the aforementioned tree-level features can be computed for each individual tree within the forest, many applications require overall summary measures of the entire network (Chen et al., 2021; Cakmak et al., 2022). To this end, the number of trees,  $n_c$ , can serve as a measure of the quantity of connected local communities within a network. For example, a sparse network typically has a large  $n_c$  because it may contain many isolated nodes. In addition, we propose the following weighted averages to characterize the overall structure of a forest representation:

$$\bar{n}_v = \sum_{i=1}^{n_c} w^{(i)} n_v^{(i)}, \quad \bar{s}_e = \sum_{i=1}^{n_c} w^{(i)} s_e^{(i)}, \quad \bar{n}_l = \sum_{i=1}^{n_c} w^{(i)} n_l^{(i)}, \quad \bar{\rho} = \sum_{i=1}^{n_c} w^{(i)} \rho^{(i)},$$

where  $w^{(i)} = n_v^{(i)} / n_v$  is the weight associated with the  $i$ th tree. These five features  $n_c$ ,  $\bar{n}_v$ ,  $\bar{s}_e$ ,  $\bar{n}_l$ , and  $\bar{\rho}$  jointly quantify the number, average size, average edge weight, average height, and average cohesion of the trees in a forest representation. They can thus be used as comprehensive and comparable summary measures across different networks.

## 3 Applications in Real-World Networks

### 3.1 US airport networks

A transportation network reflects the socioeconomic development of a country or region, and its structural and traffic properties have been extensively studied (Serrano et al., 2009; Ganin et al., 2017). Using data from the U.S. Bureau of Transportation Statistics, we constructed two airport networks for June 2019 and June 2020, as shown in Figure 1. In these networks, nodes represent airports in the contiguous United States, and each weighted edge indicates the number of direct flights between a pair of airports in the corresponding month.



Figures 1A and 1B display the adjacency matrices of the two networks. Most airport pairs have no direct flights (blank areas), while a small subset of pairs has many direct connections. In Figures 1C–1F, node size is proportional to node degree (i.e., the number of direct flight connections associated with an airport), and darker edges represent a greater number of direct flights between two airports. Figures 1C and 1D present the 2D node-link diagrams on a map. Compared with Figure 1C, the nodes in Figure 1D are generally smaller, indicating fewer flight connections in June 2020. The node with the highest degree in Figure 1C corresponds to Chicago O’Hare International Airport, while in Figure 1D, the largest node corresponds to Dallas–Fort Worth International Airport. These two airports serve as the central (red) nodes in the forest representations shown in Figures 1E and 1F.

Because a path exists between every pair of nodes, each airport network in this example contains only one tree. As shown in Figures 1E and 1F, each tree exhibits a hierarchical structure extending from the central node across three layers. In both trees, the connections among nodes in the first layer are denser than those among nodes in the second layer, while a few nodes in the third layer have no direct connections with one another. Consequently, most direct flights occur between the airport at the central node and those in the first layer, and most airports can be reached from the airport at the central node within two direct flights. Compared with Figure 1E, the nodes in Figure 1F are generally smaller, and the corresponding tree includes a few additional nodes in the third layer.

For the two airport networks from June 2019 and June 2020, the feature values  $(n_v, s_e, n_l, \rho)$  are  $(333, 661696, 3, 0.443)$  and  $(330, 226092, 3, 0.366)$ , respectively. These results show that the number of direct flights in June 2020 was approximately 34% of that in June 2019, while the number of operational airports remained similar across the two networks. The larger clustering coefficient in the June 2019 network indicates a higher level of local interconnectivity compared with the June 2020 network. The corresponding  $\mathbf{W}$  matrices of the two

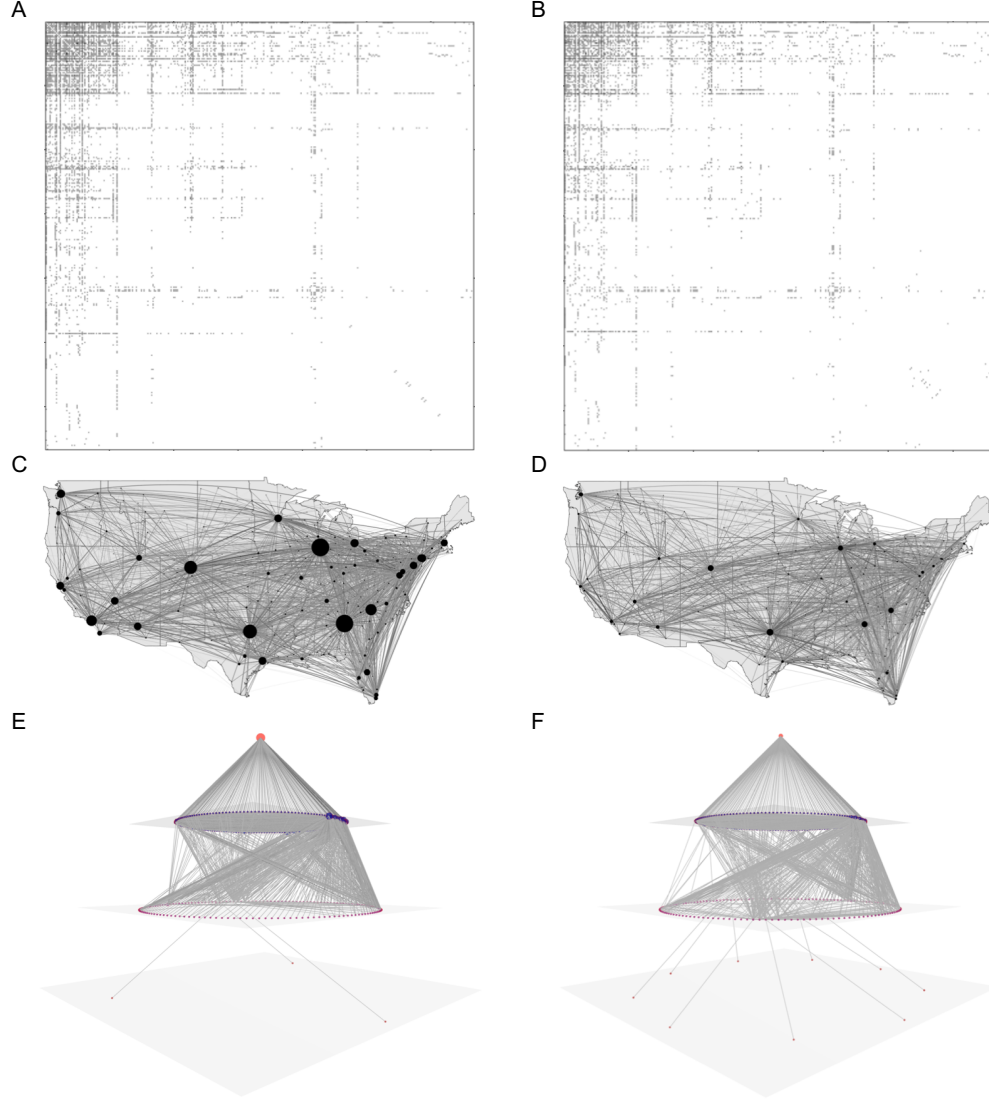


Figure 1: Visualizations of airport networks in the contiguous United States for June 2019 (left column) and June 2020 (right column). The top row displays the adjacency matrices, the middle row presents the 2D node-link diagrams on the map, and the bottom row shows the corresponding forest representations. In these plots, darker cells or edges indicate a larger number of direct flights between a pair of airports. In the middle and bottom rows, larger nodes represent airports with more direct flight connections, and red nodes denote the central nodes of the trees.

networks are

$$\begin{pmatrix} 172 & 491331 & 0.131 \\ 157 & 4844 & 0.005 \\ 3 & 0 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 166 & 152772 & 0.093 \\ 154 & 2896 & 0.009 \\ 9 & 0 & 0 \end{pmatrix},$$

and the  $\mathbf{B}$  matrices are

$$\begin{pmatrix} 69074 & 1 \\ 96389 & 0.026 \\ 58 & 0.006 \end{pmatrix} \text{ and } \begin{pmatrix} 24136 & 1 \\ 45916 & 0.025 \\ 372 & 0.006 \end{pmatrix}.$$

From these matrices, it can be observed that most direct flights occur between airports within the first neighborhood of the airport at the central node. In other words, the cone extending from the central node to the first layer in the tree contains the majority of direct flights in the airport network. Comparing the two  $\mathbf{W}$  matrices, both the number of direct flights ( $s_{e1}$ ) and the connection intensity ( $r_1$ ) among airports in the first layer of the tree in Figure 1F are substantially smaller than those in Figure 1E. Similarly, comparison of the two  $\mathbf{B}$  matrices shows that the number of direct flights and the connection intensity between the airport at the central node and those in the first layer, as well as between the first and second layers, are smaller in Figure 1F than in Figure 1E.

We also identified the ten busiest airports in each year based on node degree. Several major hubs (e.g., ORD) maintained high rankings, while others (e.g., LAX) experienced substantial declines in connectivity rank. A regional breakdown revealed that airports in the western United States suffered more severe connectivity losses. The World Health Organization declared COVID-19 a global pandemic in March 2020, and the United States declared a national emergency in the same month. The structural differences observed between the two trees confirm that the pandemic had a substantial negative impact on air transportation connectivity across the U.S.

### 3.2 Email networks of the Enron corporation

The Enron email corpus is a widely recognized network dataset that reflects communications within a real energy trading company. For the 184 Enron employees, the dataset contains their email communications over a time period from 1998 to 2002. Social dynamics among the

184 employees can be represented by a sequence of networks. Specifically, a node represents an Enron employee and a weighted edge connecting two nodes represents the number of exchanged emails between the two related employees on a given day.

Forest expressions of email communications among the Enron employees on March 27, 2001 and May 22, 2001 (two Tuesdays) are presented in Figure 2A and 2B, respectively. From the plots, it can be seen that many employees did not exchange emails with others on the two specific days. Additionally, each forest expression has a large tree and some small trees. The largest tree in Figure 2A has the features  $(n_v, s_e, n_l, \rho) = (37, 253, 7, 0.235)$ , and the largest tree in Figure 2B has the features  $(n_v, s_e, n_l, \rho) = (96, 1241, 4, 0.013)$ . Compared with the largest tree in Figure 2A, the hierarchical structure of the largest tree in Figure 2B is more evident ( $0.013 < 0.235$ ), and the central node of the largest tree in Figure 2B connects to a significantly larger number of nodes. In Figure 2A, three trees are not isolated nodes, while in Figure 2B, there are seven such trees. For these two networks on March 27 and May 22, 2001, the overall features  $(n_c, \bar{n}_v, \bar{s}_e, \bar{n}_l, \bar{\rho})$  are  $(141, 8.43, 52.28, 1.49, 0.078)$  and  $(76, 50.88, 649.03, 2.19, 0.016)$ , respectively. By comparing these metrics between the two networks, a typical tree in the second network contains more nodes and edges and is taller than that in the first network, reflecting a communication outbreak in the email network on May 22 compared with that on March 27.

Next, we consider a sequence of 537 networks constructed by aggregating daily email records from the Enron email database, covering the period from August 21, 2000, to February 8, 2002. For each network in this sequence, the features  $(n_c, \bar{n}_v, \bar{s}_e, \bar{n}_l, \bar{\rho})$  are extracted from its forest representation. Each feature is then standardized using its sample mean and standard deviation across the 537 networks. Figure 3 presents a pixel-based visualization of the standardized observations of these five features over time, where a darker pixel bar indicates a larger feature value.

From the visualization of  $n_c$ , it can be observed that pixel bars corresponding to weekends are often much darker than those of weekdays, revealing a clear day-of-week pattern

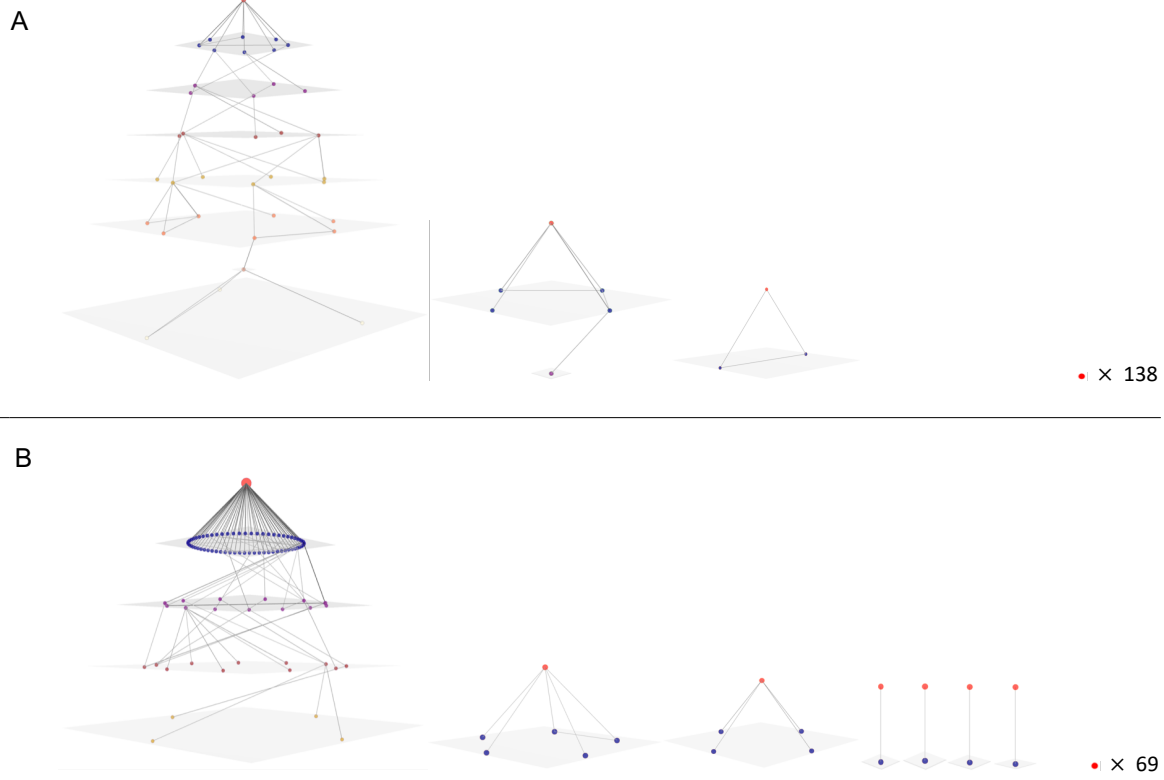


Figure 2: Forest representations of email communication networks among Enron employees on March 27, 2001 (A) and May 22, 2001 (B). In each plot, darker edges indicate a higher volume of email exchanges between pairs of employees, larger nodes represent employees with more email connections, red nodes denote the central nodes of the trees, and isolated nodes are shown as red nodes multiplied by their counts.

in the data. Comparing the 219th and 275th networks yields results consistent with the earlier comparison of the networks from March 27, 2001, and May 22, 2001. Furthermore, Figure 3 shows that the pixel bars of  $(\bar{n}_v, \bar{s}_e, \bar{n}_l, \bar{\rho})$  are generally darker between the 400th (September 24, 2001) and 465th (November 28, 2001) networks. During this period, the pixel bars of  $n_c$  are lighter on weekdays. These patterns suggest that a typical tree during this time interval contained more nodes and edges, was taller, and exhibited a higher level of clustering, while more nodes were connected on weekdays. Collectively, these findings indicate that email communications among Enron employees were particularly intense on weekdays between September 24 and November 28, 2001. According to the Enron timeline (<https://www.agsm.edu.au/bobm/teaching/BE/Enron/timeline.html>), this

surge in communication coincides with major corporate and legal developments listed in Table 1. Finally, the darkest pixel bar for  $\bar{s}_e$  at the 535th network indicates a communication outbreak on February 6, 2002—one day before key witnesses testified regarding Enron’s business practices, alleged accounting irregularities, and the company’s eventual financial collapse.

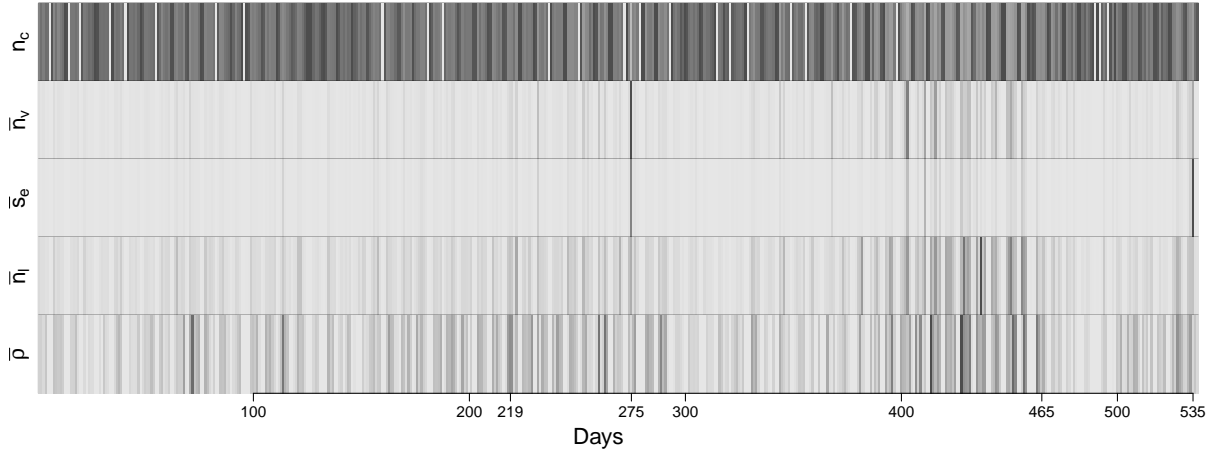


Figure 3: Pixel-based visualizations of the standardized observations of the five features  $n_c, \bar{n}_v, \bar{s}_e, \bar{n}_l, \bar{\rho}$  for the 537 daily networks constructed from the Enron email data between August 21, 2000, and February 8, 2002. The 219th and 275th networks correspond to March 27, 2001, and May 22, 2001, respectively. For each feature, a darker pixel bar represents a larger standardized value.

Table 1: Some social events in the Enron timeline

Date	Events
9/26/2001	Lay tells Enron Stock is an "incredible bargain" in employee meeting
10/16/2001	Enron reports a 638 million dollar third quarter loss
10/31/2001	Securities and Exchange Commission inquiry has been upgraded to a formal investigation
11/19/2001	Enron restates its third quarter earnings
11/28/2001	Enron shares plunge below one dollar

## 4 Simulation Studies

In this section, we perform Monte Carlo simulations to compare networks by analyzing the features obtained from their forest expressions.

### 4.1 Cases with a single large tree

In practice, a network often has a single large tree with all nodes connected. In such cases, the hurdle degree corrected stochastic blockmodel (HDCSBM) has been proposed to account for community structure and degree heterogeneity in sparse networks (Motalebi et al., 2021). Following the taxonomy of community changes in Peel and Clauset (2015), the ten cases when each network has the two-community structure are considered. In each case, 1,000 network pairs are generated from HDCSBM, and the community structure is controlled by the tuning parameters in a propensity matrix. There should be no structural changes between two simulated networks from a same model, and structural changes are introduced by using different parameters in the network model. The simulation settings are presented in the following section.

#### 4.1.1 Simulation setup for comparing the five features between two single-tree networks

The adjacency matrix  $\mathbf{A}$  is obtained by generating edges under the HDCSBM framework as described below:

$$P(a_{ij} = u) = \begin{cases} \mathfrak{S}_{c_i c_j}, & \text{if } u = 0, \\ \frac{(1 - \mathfrak{S}_{c_i c_j}) \exp(-\mathfrak{P}_{c_i c_j} \theta_i \theta_j) (\mathfrak{P}_{c_i c_j} \theta_i \theta_j)^u}{u! [1 - \exp(-\mathfrak{P}_{c_i c_j} \theta_i \theta_j)]}, & \text{if } u > 0, \end{cases}$$

where  $\mathfrak{S}_{c_i c_j}$  is the probability that there is no edge between nodes  $i$  and  $j$  in the communities  $c_i$  and  $c_j$ ,  $\mathfrak{P}_{c_i, c_j}$  is the propensity for connection between the nodes in communities  $c_i$  and  $c_j$ , and  $\theta_i$  is a degree parameter for the propensity of the node  $i$ . For a given network size  $n_v$ ,

we consider two equal-sized (or nearly equal-sized) communities in HDCSBM. Thus,  $\mathfrak{S}$  and  $\mathfrak{P}$  are both  $2 \times 2$  matrices. The sizes of the two communities are  $(n_v + 1)/2$  and  $(n_v - 1)/2$  if  $n_v$  is odd, and both equal to  $n_v/2$  if  $n_v$  is even. We randomly generate  $\theta_i$  from the uniform distribution  $U(0.5, 1.5)$ , and the degree parameters in the community  $r$  are scaled to meet the constraint that  $\sum_{i \in c_r} \theta_i = n_{v, c_r}$ , where  $n_{v, c_r}$  is the number of nodes in the community  $r$ . The diagonal elements of  $\mathfrak{S}$  that control the within-community sparsity are set to be 0.5, and the off-diagonal elements of  $\mathfrak{S}$  that control the between-community sparsity are set to be 0.9. If there is no further specification, we randomly choose an integer from 450 to 550 to be  $n_v$ , and the propensity matrix  $\mathfrak{P}$  is  $\begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}$ . Figure 4 presents a network generated by HDCSBM. From the figure, it can be seen that the two communities can be identified by the dense interconnections among their nodes.

Network features are often used to compare two networks (Chen et al., 2021). When each of the two networks has a single tree, we can extract the features  $(n_v, s_e, \rho, \mathbf{W}, \mathbf{B})$  from each tree. The differences of the features  $n_v$ ,  $s_e$ , and  $\rho$  between the two networks are denoted as NVD, SED, and CD. While  $CD \in [-1, 1]$ , NVD or SED can take positive, negative, or zero values beyond this range.

To compare the within-layer structure of two trees, we define the following within-layer feature differences. The  $\mathbf{W}$  matrices of the two trees are denoted as  $\mathbf{W}_1$  and  $\mathbf{W}_2$ . Specifically,  $\mathbf{W}_1$  and  $\mathbf{W}_2$  are  $n_{l1} \times 3$  and  $n_{l2} \times 3$  matrices, respectively. If  $n_{l1}$  and  $n_{l2}$  are both nonzero, let  $n_l^c = \max\{n_{l1}, n_{l2}\}$ . If  $n_{l1} < n_{l2}$ , define  $\mathbf{S}_1 = \begin{pmatrix} \mathbf{W}_1 \\ \mathbf{0}_{(n_l^c - n_{l1}) \times 3} \end{pmatrix}$  and  $\mathbf{S}_2 = \mathbf{W}_2$ . If  $n_{l2} < n_{l1}$ , define  $\mathbf{S}_2 = \begin{pmatrix} \mathbf{W}_2 \\ \mathbf{0}_{(n_l^c - n_{l2}) \times 3} \end{pmatrix}$  and  $\mathbf{S}_1 = \mathbf{W}_1$ . Finally, if  $n_{l1} = n_{l2}$ , define  $\mathbf{S}_1 = \mathbf{W}_1$  and  $\mathbf{S}_2 = \mathbf{W}_2$ . The  $(i, j)$ th element of  $\mathbf{S}_r$  ( $r = 1, 2$ ) is denoted as  $x_{r, ij}$ , and all  $x_{r, ij} \geq 0$ . The within-layer



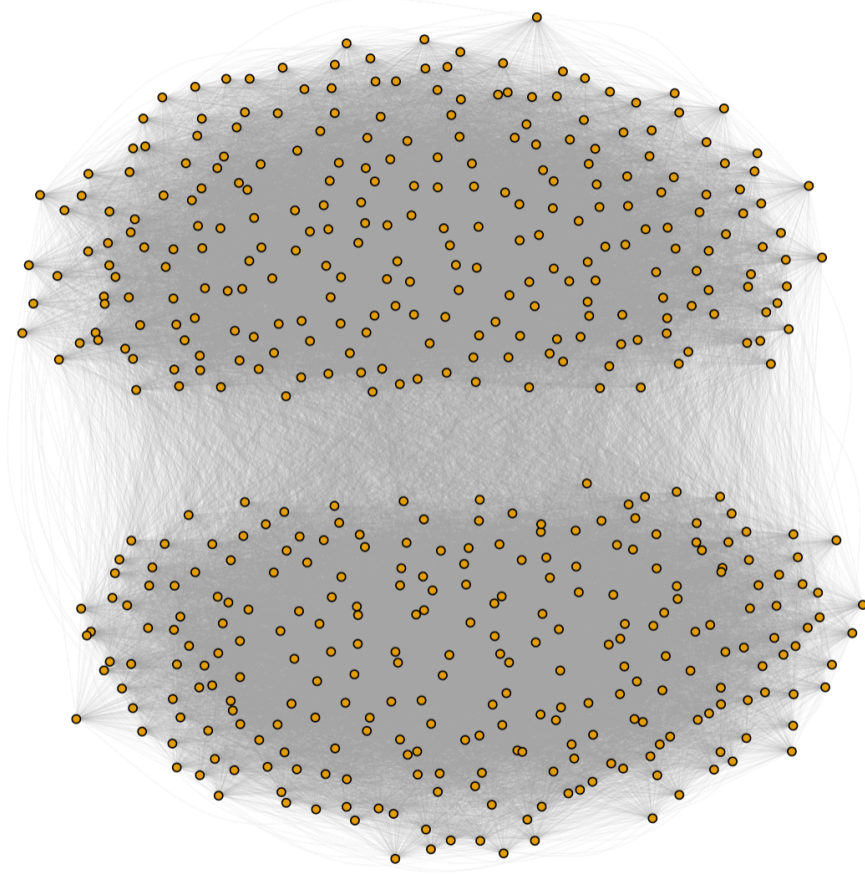


Figure 4: A network generated by HDCSBM. The network is a connected component with a two-community structure, and interconnections within each community are dense.

feature difference is then computed by  $\sum_{i=1}^{n_l^c} \sum_{j=1}^3 q_{ij} / 3n_l^c$ , where

$$q_{ij} = \begin{cases} 0, & \text{if } x_{1,ij} = 0 \text{ and } x_{2,ij} = 0, \\ \frac{2|x_{1,ij} - x_{2,ij}|}{x_{1,ij} + x_{2,ij}}, & \text{otherwise.} \end{cases}$$

Namely, the above feature difference is defined to be the average of relative mean absolute differences (RMDs) between elements in  $\mathbf{S}_1$  and  $\mathbf{S}_2$ , and denoted as WRMD. From the definition, the WRMD is expected to be small if the within-layer structures of the two trees are similar.

The between-layer feature difference of two networks can be defined similarly as follows. Let  $\mathbf{B}_1$  and  $\mathbf{B}_2$  have the dimensions  $n_{l1} \times 2$  and  $n_{l2} \times 2$ , respectively. To compare the

between-layer structure of the two trees, the between-layer feature difference is defined to be the average of RMDs between the elements of  $\mathbf{U}_1$  and  $\mathbf{U}_2$  defined below. If  $n_{l1} < n_{l2}$ , define  $\mathbf{U}_1 = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{0}_{(n_l^c - n_{l1}) \times 2} \end{pmatrix}$  and  $\mathbf{U}_2 = \mathbf{B}_2$ . If  $n_{l2} < n_{l1}$ , define  $\mathbf{U}_2 = \begin{pmatrix} \mathbf{B}_2 \\ \mathbf{0}_{(n_l^c - n_{l2}) \times 2} \end{pmatrix}$  and  $\mathbf{U}_1 = \mathbf{B}_1$ . Finally, if  $n_{l1} = n_{l2}$ , define  $\mathbf{U}_1 = \mathbf{B}_1$  and  $\mathbf{U}_2 = \mathbf{B}_2$ . Then, the between-layer feature difference is denoted as BRMD, and defined to be

$$\frac{\sum_{i=1}^{n_l^c} \sum_{j=1}^2 \frac{2|y_{1,ij} - y_{2,ij}|}{y_{1,ij} + y_{2,ij}}}{2n_l^c},$$

where  $y_{r,ij}$  is the  $(i, j)$ th element of  $\mathbf{U}_r$ , for  $r = 1, 2$ . In the above summation, the  $(i, j)$ th term is set to be 0 if  $y_{1,ij} = y_{2,ij} = 0$ . The value of BRMD is expected to be small if the between-layer structures of the two trees are similar. From the above definitions, both WRMD and BRMD would fall within the interval  $[0, 2)$ . In addition, an isolated node does not have any hierarchical structure. For a pair of two networks, both WRMD and BRMD are set to be 2 if only one of them is an isolated node (e.g.,  $n_{l1} \geq 1$  and  $n_{l2} = 0$ ), and 0 if both of them are isolated nodes.

To study the performance of the five feature differences when there are structural changes between two networks, we consider the ten cases shown in Figure 5 in Monte Carlo simulations. In each case, 1,000 network pairs are simulated from HDCSBM, and the five feature differences are computed for each network pair. We report the sample means and the standard errors of the five feature differences. To check whether the means of NVD, SED, and CD are significantly different from 0, the Wilcoxon rank test is used if the data are not normally distributed. Otherwise, the  $t$ -test is used. Community structures of the two networks in a pair are different in Cases I0, II0, and III0. In these cases, no structural changes are expected in the simulated network pairs. We use the Pearson's chi-square test to compare the sample distributions of WRMD (or BRMD). Specifically, the distribution of WRMD (or BRMD) in Cases I1–I5 is compared with that in Case I0, and the distribution of WRMD (or BRMD) in Case II1 (or Case III1) is compared with that in Case II0 (or

Case III0). For WRMD, the Pearson's chi-square test is based on the following ten intervals:  $[0, 0.02], (0.02, 0.04], \dots, (0.16, 0.18], (0.18, 2]$ . For BRMD, the Pearson's chi-square test is based on the following ten intervals:  $[0, 0.02], \dots, (0.12, 0.14], (0.14, 0.15], (0.15, 0.16], (0.16, 2]$ .

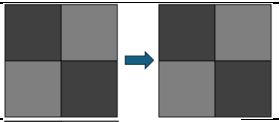

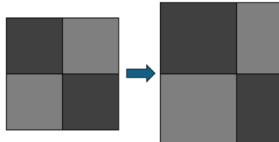
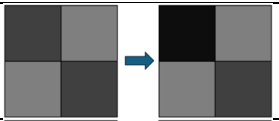





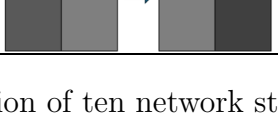
Case	Change description	Graphic representation	Parameter change representation
I0	No structural change		$\mathfrak{P} = \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}$ is used to simulate network pairs
I1	Decreased network size		$n_v$ is randomly chosen from: $\{450, 451, \dots, 550\} \rightarrow \{350, 351, \dots, 450\}$
I2	Increased network size		$n_v$ is randomly chosen from: $\{450, 451, \dots, 550\} \rightarrow \{550, 551, \dots, 650\}$
I3	Intensified communication		$\mathfrak{P}: \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{pmatrix} \rightarrow \begin{pmatrix} 0.5 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}$
I4	Reduced communication		$\mathfrak{P}: \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{pmatrix} \rightarrow \begin{pmatrix} 0.1 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}$
I5	Fragment		$\mathfrak{P}: \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{pmatrix} \rightarrow \begin{pmatrix} 0.4 & 0.2 \\ 0.2 & 0.1 \end{pmatrix}$
II0	No structural change		$\mathfrak{P} = \begin{pmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{pmatrix}$ is used to simulate network pairs
II1	Split		$\mathfrak{P}: \begin{pmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{pmatrix} \rightarrow \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}$
III0	No structural change		$\mathfrak{P} = \begin{pmatrix} 0.4 & 0.2 \\ 0.2 & 0.1 \end{pmatrix}$ is used to simulate network pairs
III1	Form		$\mathfrak{P}: \begin{pmatrix} 0.4 & 0.2 \\ 0.2 & 0.1 \end{pmatrix} \rightarrow \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}$

Figure 5: Taxonomy and representation of ten network structural changes using HDCSBM.

### 4.1.2 Simulation results

The means and standard errors of the feature differences from 1,000 simulated network pairs are summarized in Table 2. From the table, it can be seen that significant mean differences in NVDs occur only when the number of nodes changes. In all cases without structural changes, the means of SEDs and CDs are not significantly different from zero. In contrast, in most cases with structural changes, the means of SEDs and CDs are significantly different from zero. Moreover, when the mean differences of NVD, SED, or CD are significant, their magnitudes tend to be large.

The distributions of WRMDs in different cases are shown in Figure 6. From the figure, it can be seen that the distributions of WRMDs in Cases I1–I5 differ significantly from that in Case I0, with the corresponding mean values (cf., Table 2) being consistently higher than that in Case I0. Similarly, Figure 7 shows that the distributions of BRMDs in Cases I1–I5 are significantly different from that in Case I0. The distribution of BRMDs in Case II1 also differs significantly from that in Case II0, and that in Case III1 differs from Case III0. The mean BRMD values are larger in Cases I1–I5 than in Case I0, larger in Case II1 than in Case II0, and smaller in Case III1 than in Case III0. These simulation results collectively confirm the effectiveness of the proposed forest expression and associated network features for comparing network structures across the scenarios considered.

## 4.2 Cases with multiple trees

Networks with multiple connected components are common in large dynamic systems. For example, a linear relationship between the number of non-isolated nodes and the number of binary edges has been observed in real temporal networks (Leskovec and Krevl, 2014). We leverage this linear relationship along with the Poisson hurdle model to generate network pairs in seven specific cases. [In each case, 1,000 network pairs are generated, with the network structure controlled by tuning parameters in the Poisson hurdle model.](#) The detailed simulation settings are described in the following section.

Table 2: Sample means and standard errors (in parentheses) of the five feature differences across Cases I0–III1, each based on 1,000 simulated network pairs. The sample means of CD are reported on the scale of  $10^{-5}$ . The sample means of NVD, SED, and CD marked with \* (or \*\*) indicate cases where the  $p$ -values for testing their difference from 0 are less than 0.05 (or 0.01). For WRMD and BRMD, each pair of values represents the sample mean (left) and the standard error (right).

Cases	NVD	SED	CD	WRMD	BRMD
I0	−0.32 (1.29)	−53.9 (221.05)	8.14 (8.15)	(0.078, 0.001)	(0.062, 0.001)
I1	99.68 (1.33)**	15345.19 (206.97)**	13.59 (9.50)	(0.227, 0.003)	(0.171, 0.002)
I2	−102.46 (1.28)**	−19236.33 (241.91)**	−11.32 (7.82)	(0.192, 0.002)	(0.143, 0.002)
I3	1.06 (1.26)	−1616.34 (220.63)**	−163.15 (8.19)**	(0.079, 0.001)	(0.067, 0.001)
I4	0.10 (1.31)	1717.25 (219.39)**	147.30 (8.32)**	(0.081, 0.001)	(0.066, 0.001)
I5	1.62 (1.32)	756.65 (223.90)**	230.16 (8.36)**	(0.081, 0.001)	(0.068, 0.001)
II0	−0.19 (1.28)	−30.05 (211.97)	11.85 (8.39)	(0.076, 0.001)	(0.061, 0.001)
II1	1.83 (1.27)	−1087.39 (214.62)**	−300.81 (8.08)**	(0.078, 0.001)	(0.063, 0.001)
III0	1.69 (1.31)	288.61 (222.18)	−1.95 (8.48)	(0.083, 0.001)	(0.070, 0.001)
III1	1.96 (1.32)	−125.08 (224.69)	−218.19 (8.50)**	(0.081, 0.001)	(0.067, 0.001)

#### 4.2.1 Simulation setup for comparison of the five features between two multi-tree networks

A network can contain multiple connected components. In our simulation study, we generate networks using the following model. The edge distribution for each network is specified by a Poisson hurdle model as follows: for  $1 \leq i \neq j \leq n_v$ ,

$$P(a_{ij} = z) = \begin{cases} \pi, & \text{if } z = 0, \\ \frac{(1-\pi)\exp(-\lambda)\lambda^z}{z![1-\exp(-\lambda)]}, & \text{if } z > 0, \end{cases}$$

where  $\pi$  is a sparsity parameter denoting the probability of no edge between two nodes in any pair, and the connectivity parameter  $\lambda$  controls the edge weight. This network model is based on a mixture of a Bernoulli distribution with parameter  $\pi$  and a positive Poisson distribution with parameter  $\lambda$  (Mullahy, 1986).

A complete network is then generated as follows. First, if there is no further specification,  $n_v$  is a randomly chosen integer from 450 to 550. Second, the expected number of nodes that are not isolated is denoted as  $\mathbf{n}_v$ , and  $\mathbf{n}_v$  is set to be  $\mathbf{n}_v = \langle pn_v \rangle$ , where  $p$  is randomly

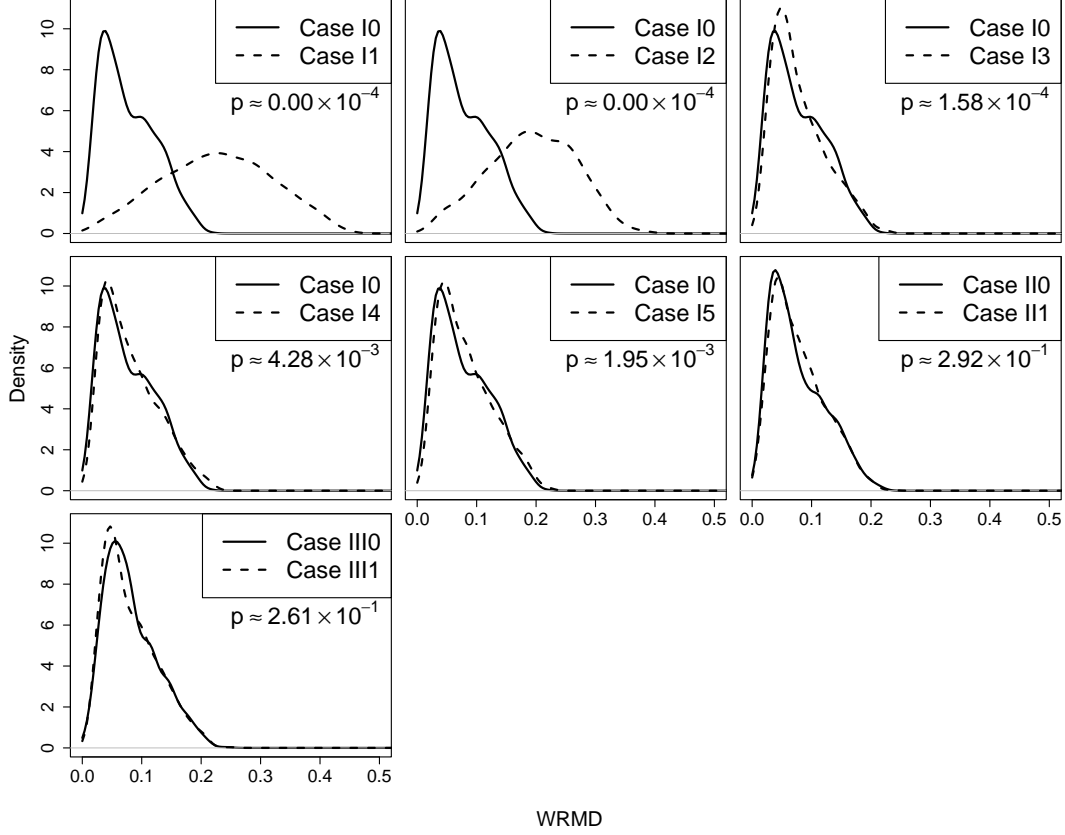


Figure 6: Density plots of WRMDs from simulated network pairs across the ten cases considered. In each plot, the density for a case without structural change is shown as a solid line, while the density for a case with structural change is shown as a dashed line. The  $p$ -value from Pearson's chi-square test is also reported.

generated from the Uniform distribution on the interval  $[p_1, p_2]$ , and  $p_1$  and  $p_2$  represent the lower and upper proportions of nodes that are expected to have edges in the network. The number of expected binary edges is denoted as  $\mathbf{n}_e$ . Its maximum value is  $\mathbf{n}_e^{\max} = \mathbf{n}_v(\mathbf{n}_v - 1)/2$ , and its minimum value is  $\mathbf{n}_e^{\min} = (\mathbf{n}_v + 1)/2$  when  $\mathbf{n}_v$  is odd and  $\mathbf{n}_e^{\min} = \mathbf{n}_v/2$  when  $\mathbf{n}_v$  is even. Then,  $\mathbf{n}_e$  is chosen to be

$$\mathbf{n}_e = \begin{cases} \langle \mathbf{n}_v + \epsilon \rangle, & \text{if } \mathbf{n}_e^{\min} \leq \langle \mathbf{n}_v + \epsilon \rangle \leq \mathbf{n}_e^{\max}, \\ \mathbf{n}_e^{\min}, & \text{if } \mathbf{n}_e^{\min} > \langle \mathbf{n}_v + \epsilon \rangle, \\ \mathbf{n}_e^{\max}, & \text{if } \mathbf{n}_e^{\max} < \langle \mathbf{n}_v + \epsilon \rangle, \end{cases}$$

where  $\epsilon$  is a random number generated from the  $N(0, 3^2)$  distribution. In the above equation,

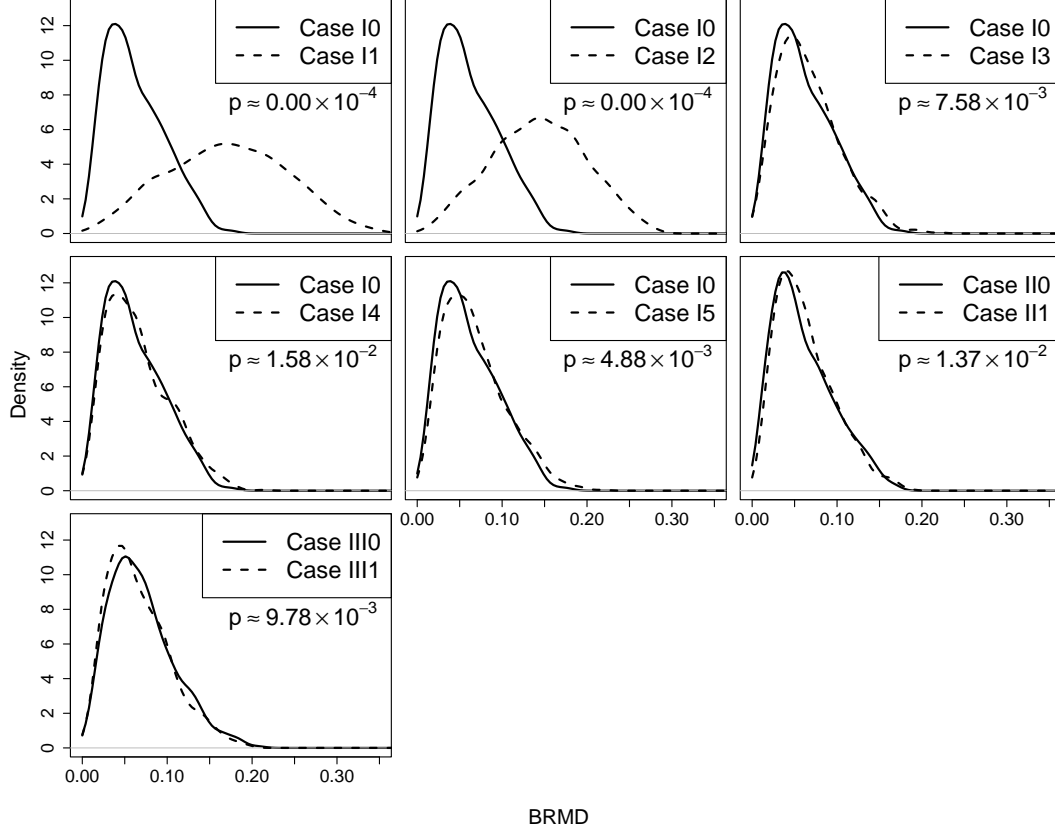


Figure 7: Density plots of BRMDs from simulated network pairs across the ten cases considered. In each plot, the density for a case without structural change is shown as a solid line, while the density for a case with structural change is shown as a dashed line. The  $p$ -value from Pearson's chi-square test is also reported.

the linear relationship between the number of non-isolated nodes and the number of binary edges is used. After  $\mathbf{n}_e$  is determined, the value of the sparsity parameter is chosen to be

$$\pi = 1 - \frac{2\mathbf{n}_e}{n_v(n_v - 1)}.$$

Third, the value of  $\lambda$  used in the model is determined as follows. Let  $\mu_e$  be a pre-specified value (e.g., 2) of the expected degree of a node. Based on the Poisson hurdle model, this number can be calculated to be  $(n_v - 1)E(a_{ij})$ , which is the sum of the expected edge weights between node  $i$  and all other nodes. Thus, we have

$$\frac{\mu_e}{(n_v - 1)(1 - \pi)} = \frac{\lambda}{1 - \exp(-\lambda)}.$$

Because  $n_v$  and  $\pi$  in the above equation have been determined in advance, the value of  $\lambda$  can be determined by solving the above equation, which is unique when  $\pi > 1 - \mu_e/(n_v - 1)$ .

After the values of  $\{n_v, \pi, \lambda\}$  are determined as described above, the adjacency matrix  $\mathbf{A}$  can be generated by the Poisson hurdle model. From the above description, it can be seen that the structure of the generated network is uniquely determined by the pre-specified values of  $\{n_v, p_1, p_2, \mu_e\}$ . Figure 8 presents one such network as a demonstration.

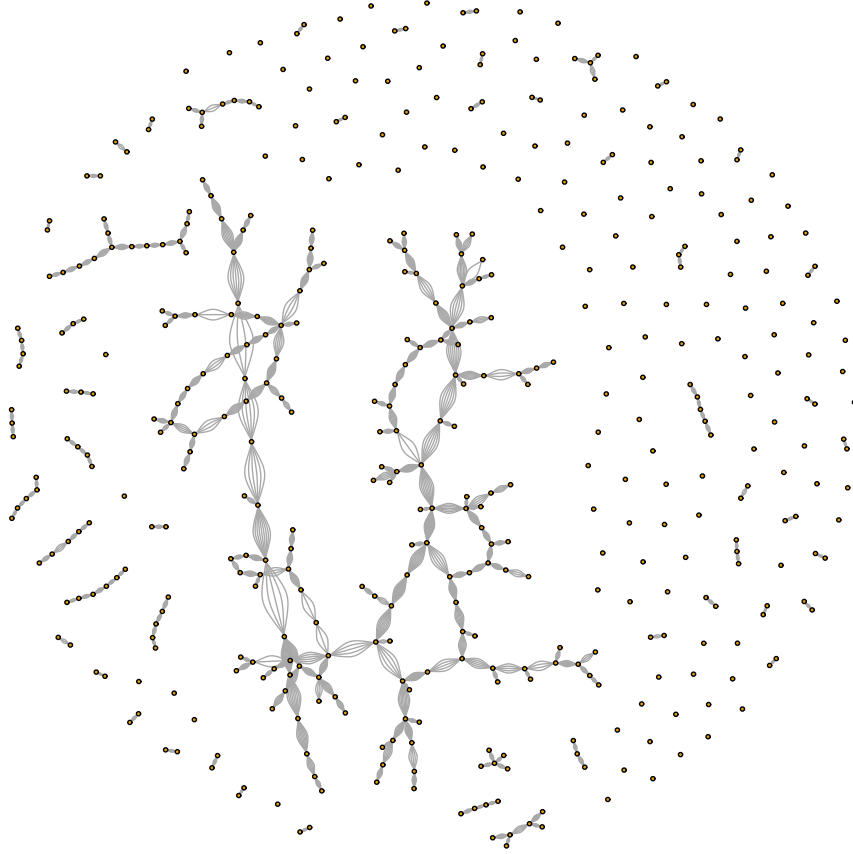


Figure 8: A network generated by the Poisson hurdle model with  $n_v = 452$ ,  $p_1 = 0.4$ ,  $p_2 = 0.6$ , and  $\mu_e = 8$ .

As in the single-tree network cases discussed in Section 4.1, differences in the five features  $n_c$ ,  $\bar{n}_v$ ,  $\bar{s}_e$ ,  $\bar{n}_l$ , and  $\bar{\rho}$  between the two networks in a pair can be computed based on all trees in their forest expressions. These five feature differences are denoted by NCD, ANVD, ASED, ANLD, and ACD, respectively. It can be verified that  $ACD \in [-1, 1]$ , while the other four feature differences may take positive, negative, or zero values outside this range. To eval-



uate the performance of the five feature differences when they are used for comparing two networks with structural changes, the following seven cases are considered in simulations.

**Case IV0:** For both networks in a network pair, the expected proportion of non-isolated nodes in a network is chosen between  $p_1 = 0.4$  and  $p_2 = 0.6$ , the expected degree of a node in a network is set to be  $\mu_e = 8$ , and the number of nodes  $n_v$  of a network is a randomly chosen integer from 450 to 550. Using these parameters, 1,000 network pairs are generated. In this case, no structural change is expected within a network pair because both networks in the pair are generated using the same parameters in the Poisson hurdle model.

**Case IV1:** Same as Case IV0, except that  $n_v$  of the second network in a network pair is a randomly chosen integer from 350 to 450. Thus, the network size decreases from the first network to the second network in a network pair.

**Case IV2:** Same as Case IV0, except that  $n_v$  of the second network in each network pair is a randomly chosen integer from 550 to 650. Thus, the network size increases from the first network to the second network in a network pair.

**Case IV3:** Same as Case IV0, except that the second network in each network pair is generated from the Poisson hurdle model using  $p_1 = 0.3$ ,  $p_2 = 0.5$ , and  $\mu_e = 6$ . Thus, the communication decreases from the first network to the second network in a network pair.

**Case IV4:** Same as Case IV0, except that the second network in each network pair is generated from the Poisson hurdle model using  $p_1 = 0.5$ ,  $p_2 = 0.7$ , and  $\mu_e = 10$ . Thus, the communication increases from the first network to the second network in a network pair.

**Case IV5:** Same as Case IV0, except that the second network has  $n_v$  randomly chosen from 350 to 450, and it is generated from the Poisson hurdle model using  $p_1 = 0.3$ ,

$p_2 = 0.5$ , and  $\mu_e = 6$ . Thus, this network shrinks compared to the first network in a network pair.

**Case IV6:** Same as Case IV0, except that the second network has  $n_v$  randomly chosen from 550 to 650, and it is generated from the Poisson hurdle model using  $p_1 = 0.5$ ,  $p_2 = 0.7$ , and  $\mu_e = 10$ . Thus, this network expands compared to the first network in a network pair.

For each case considered, the sample means and standard errors of the five feature differences NCD, WNVD, WSED, WNLD, and WCD are computed. To assess whether the means of these feature differences are significantly different from zero, the Wilcoxon rank-sum test is applied when the data deviate from normality; otherwise, a two-sample  $t$ -test is used.

#### 4.2.2 Simulation results

Table 3 presents the sample means and standard errors of the five feature differences across all cases, each computed from 1,000 simulated network pairs. From the results, several conclusions can be drawn. In Case IV0, where no structural change is present, none of the five mean feature differences are significantly different from zero. In contrast, all five means differ significantly from zero in Cases IV3–IV6, where communication or evolution changes are introduced. The means of NCDs are also significantly different from zero in Cases IV1 and IV2, where the number of nodes changes. These findings indicate that the five feature differences are generally sensitive to structural variations across Cases IV1–IV6.

## 5 Conclusions

The proposed forest expression provides a novel and flexible framework for representing networks, accommodating a wide range of network data, including dynamic systems. This representation effectively captures both hierarchical and component-wise structures through interpretable topological features. Numerical studies demonstrate that the proposed feature

Table 3: Sample means and standard errors (in parentheses) of the five feature differences in Cases IV0–IV6, each based on 1,000 simulated network pairs. The sample means of ACD are reported on the scale of  $10^{-5}$ . The symbols “\*” and “\*\*” indicate cases where the means of the corresponding feature differences are significantly different from zero, with  $p$ -values less than 0.05 and 0.01, respectively.

Cases	NCD	ANVD	ASED	ANLD	ACD
IV0	−1.79 (1.54)	1.53 (0.88)	11.11 (6.17)	0.15 (0.11)	11.63 (7.52)
IV1	50.83 (1.47)**	1.94 (0.78)	14.52 (5.49)	0.09 (0.10)	31.75 (9.38)
IV2	−49.02 (1.67)**	−2.90 (0.93)*	−20.57 (6.52)*	−0.26 (0.11)	9.08 (7.12)
IV3	−46.45 (1.58)**	12.31 (0.63)**	95.38 (4.35)**	1.98 (0.08)**	25.79 (6.61)**
IV4	44.36 (1.54)**	−37.39 (1.54)**	−318.02 (11.80)**	−3.16 (0.13)**	46.77 (8.70)**
IV5	9.98 (1.52)**	13.55 (0.66)**	104.23 (4.62)**	2.15 (0.09)**	32.06 (7.03)**
IV6	7.72 (1.62)**	−51.17 (1.75)**	−426.81 (13.40)**	−4.03 (0.13)**	−24.49 (8.08)*

differences are highly effective in detecting structural changes under various scenarios, such as variations in network size, edge density, community structure, and connectivity patterns. Visualizations based on the forest expression offer valuable insights into network structures, enhancing interpretability in network analysis. Moreover, forest expressions can be readily applied to online monitoring of dynamic networks (Liu et al., 2021; Yu et al., 2023; Wang et al., 2024), which will be further investigated in our future work.

## Acknowledgements

The authors are grateful to the editor, associate editor, and referee for their valuable and constructive comments, which have significantly enhanced the quality of this paper.

## Appendix: Typical Structural Changes of Connected Components

A connected component of an undirected network is a subnetwork in which every pair of nodes is connected by a path, and no nodes outside the subnetwork are connected to it. Connected components are fundamental building blocks of network structure (Von Landesberger et al., 2009; Sekara et al., 2016). Following the task taxonomy proposed by Ahn et al. (2013), seven typical types of structural changes in connected components are summarized in Figure 9. As shown in the figure, “Growth” (or “Contraction”) of a connected component

refers to an increase (or decrease) in the number of edges. “Merging” (or “Splitting”) leads to fewer (or more) connected components. “Birth” represents the emergence of a new connected component, while “Death” indicates the disappearance of an existing one. Finally, “Shape Change” describes cases where the numbers of nodes and edges remain constant, but the internal structure of the component changes.

## References

- Abu-Ata, M. and Dragan, F. F. (2016). Metric tree-like structures in real-world networks: an empirical study. *Networks*, 67(1):49–68.
- Ahn, J.-w., Plaisant, C., and Shneiderman, B. (2013). A task taxonomy for network evolution analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):365–376.
- Alghuried, A. and Moghaddass, R. (2021). Anomaly detection in large-scale networks: A state-space decision process. *Journal of Quality Technology*, 54(1):65–92.
- Beck, F., Burch, M., Diehl, S., and Weiskopf, D. (2017). A taxonomy and survey of dynamic graph visualization. In *Computer Graphics Forum*, volume 36, pages 133–159.
- Behr, M., Ansari, M. A., Munk, A., and Holmes, C. (2020). Testing for dependence on tree structures. *Proceedings of the National Academy of Sciences*, 117(18):9787–9792.
- Bravo-Hermsdorff, G., Gunderson, L. M., Maugis, P.-A., and Priebe, C. E. (2023). Quantifying network similarity using graph cumulants. *Journal of Machine Learning Research*, 24(187):1–27.
- Cakmak, E., Fuchs, J., Jäckle, D., Schreck, T., Brandes, U., and Keim, D. (2022). Motif-based visual analysis of dynamic networks. In *IEEE Visualization in Data Science*, pages 17–26.
- Chen, M. K., Chevalier, J. A., and Long, E. F. (2021). Nursing home staff networks and covid-19. *Proceedings of the National Academy of Sciences*, 118(1):e2015455118.

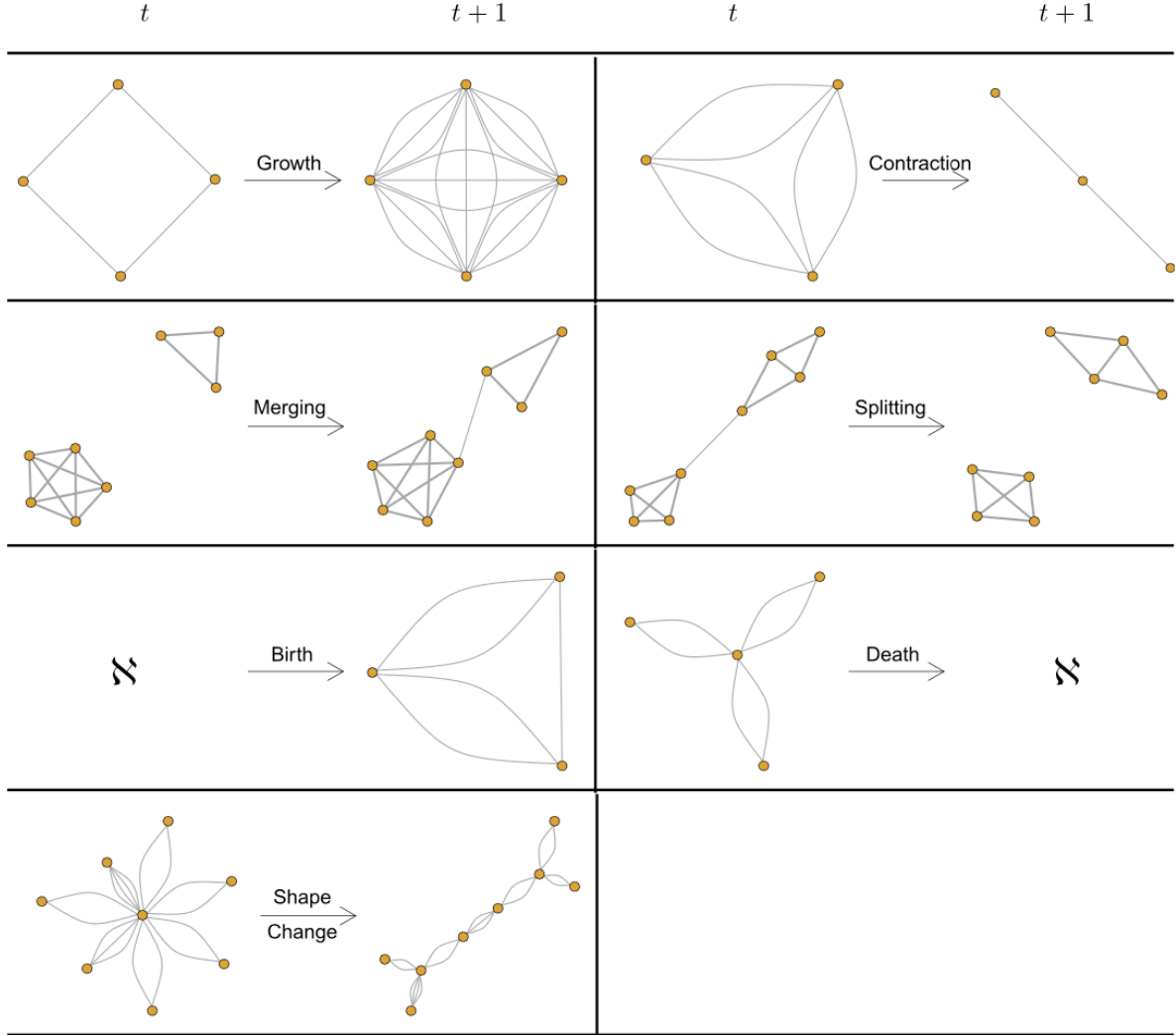


Figure 9: Typical structural changes of connected components. For birth or death of a connected component, the empty graph with no nodes is denoted as  $\mathcal{N}$ . In the demonstrated shape change of a connected component, the numbers of nodes and edges remain the same. For plots in the second row, the unchanged connected components at two time points are shown with thicker edges.

- Corominas-Murtra, B., Goñi, J., Solé, R. V., and Rodríguez-Caso, C. (2013). On the origins of hierarchy in complex networks. *Proceedings of the National Academy of Sciences*, 110(33):13316–13321.
- Dijkstra, E. W. (2022). A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His life, work, and legacy*, pages 287–290.
- Emmert-Streib, F., Dehmer, M., and Shi, Y. (2016). Fifty years of graph matching, network alignment and network comparison. *Information Sciences*, 346:180–197.
- Ganin, A. A., Kitsak, M., Marchese, D., Keisler, J. M., Seager, T., and Linkov, I. (2017). Resilience and efficiency in transportation networks. *Science Advances*, 3(12):e1701079.
- Ghoshdastidar, D., Gutzeit, M., Carpentier, A., and Von Luxburg, U. (2020). Two-sample hypothesis testing for inhomogeneous random graphs. *The Annals of Statistics*, 48(4):2208–2229.
- Gray, K., Li, M., Ahmed, R., Rahman, M. K., Azad, A., Kobourov, S., and Börner, K. (2023). A scalable method for readable tree layouts. *IEEE Transactions on Visualization and Computer Graphics*, 30(2):1564–1578.
- Halin, R. (1976). S-functions for graphs. *Journal of Geometry*, 8(1-2):171–186.
- Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.
- Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., and Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature Physics*, 6(11):888–893.
- Lee, W., McCormick, T. H., Neil, J., Sodja, C., and Cui, Y. (2022). Anomaly detection in large-scale networks with latent space models. *Technometrics*, 64(2):241–252.
- Leskovec, J. and Krevl, A. (2014). Snap datasets: Stanford large network dataset collection.

- Li, D., Fu, B., Wang, Y., Lu, G., Berezin, Y., Stanley, H. E., and Havlin, S. (2015). Percolation transition in dynamical traffic network with evolving critical bottlenecks. *Proceedings of the National Academy of Sciences*, 112(3):669–672.
- Li, T., Lei, L., Bhattacharyya, S., Van den Berge, K., Sarkar, P., Bickel, P. J., and Levina, E. (2022). Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association*, 117(538):951–968.
- Linhares, C. D., Ponciano, J. R., Pedro, D. S., Rocha, L. E., Traina, A. J., and Poco, J. (2022). Largenetvis: Visual exploration of large temporal networks based on community taxonomies. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):203–213.
- Liu, Y., Gu, Z., and Liu, J. (2021). Uncovering transmission patterns of covid-19 outbreaks: A region-wide comprehensive retrospective study in hong kong. *EClinicalMedicine*, 36:100929.
- Lyzinski, V., Tang, M., Athreya, A., Park, Y., and Priebe, C. E. (2016). Community detection and classification in hierarchical stochastic blockmodels. *IEEE Transactions on Network Science and Engineering*, 4(1):13–26.
- Mcauley, J. and Leskovec, J. (2014). Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data*, 8(1):1–28.
- Moody, J., McFarland, D., and Bender-deMoll, S. (2005). Dynamic network visualization. *American Journal of Sociology*, 110(4):1206–1241.
- Motalebi, N., Stevens, N. T., and Steiner, S. H. (2021). Hurdle blockmodels for sparse network modeling. *The American Statistician*, 75(4):383–393.
- Mullahy, J. (1986). Specification and testing of some modified count data models. *Journal of Econometrics*, 33(3):341–365.
- Newman, M. (2018). *Networks*. Oxford University Press.
- Opsahl, T. and Panzarasa, P. (2009). Clustering in weighted networks. *Social Networks*, 31(2):155–163.

- Panzarasa, P., Opsahl, T., and Carley, K. M. (2009). Patterns and dynamics of users’ behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology*, 60(5):911–932.
- Peel, L. and Clauset, A. (2015). Detecting change points in the large-scale structure of evolving networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pages 2914–2920.
- Pržulj, N. (2007). Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183.
- Rossetti, G. and Cazabet, R. (2018). Community discovery in dynamic networks: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–37.
- Rubinov, M. and Sporns, O. (2010). Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069.
- Schieber, T. A., Carpi, L., Díaz-Guilera, A., Pardalos, P. M., Masoller, C., and Ravetti, M. G. (2017). Quantification of network structural dissimilarities. *Nature Communications*, 8(1):13928.
- Schlosser, F., Maier, B. F., Jack, O., Hinrichs, D., Zachariae, A., and Brockmann, D. (2020). Covid-19 lockdown induces disease-mitigating structural changes in mobility networks. *Proceedings of the National Academy of Sciences*, 117(52):32883–32890.
- Sekara, V., Stopczynski, A., and Lehmann, S. (2016). Fundamental structures of dynamic social networks. *Proceedings of the National Academy of Sciences*, 113(36):9977–9982.
- Serrano, M. Á., Boguná, M., and Vespignani, A. (2009). Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences*, 106(16):6483–6488.
- Shu, K., Mahudeswaran, D., Wang, S., and Liu, H. (2020). Hierarchical propagation networks for fake news detection: Investigation and exploitation. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 626–637.



- Songdechakraiut, T. and Chung, M. K. (2023). Topological learning for brain networks. *The Annals of Applied Statistics*, 17(1):403–433.
- Tversky, B., Morrison, J. B., and Betrancourt, M. (2002). Animation: Can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262.
- Vidaurre, D., Smith, S. M., and Woolrich, M. W. (2017). Brain network dynamics are hierarchically organized in time. *Proceedings of the National Academy of Sciences*, 114(48):12827–12832.
- Von Landesberger, T., Gorner, M., and Schreck, T. (2009). Visual analysis of graphs with multiple connected components. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 155–162.
- Von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J. J., Fekete, J.-D., and Fellner, D. W. (2011). Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer Graphics Forum*, volume 30, pages 1719–1749.
- Wang, Y., Xie, X., and Qiu, P. (2024). Nonparametric online monitoring of dynamic networks. *Journal of Quality Technology*, 56(3):214–243.
- Yoghourdjian, V., Yang, Y., Dwyer, T., Lawrence, L., Wybrow, M., and Marriott, K. (2020). Scalability of network visualisation from a cognitive load perspective. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1677–1687.
- Yu, H. and Gerstein, M. (2006). Genomic analysis of the hierarchical structure of regulatory networks. *Proceedings of the National Academy of Sciences*, 103(40):14724–14731.
- Yu, M., Zhou, Y., and Tsung, F. (2023). Robust online detection in serially correlated directed network. *Naval Research Logistics*, 70(7):735–752.
- Zhao, Y., Levina, E., and Zhu, J. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292.