This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3423721

DBNetVizor: Visual Analysis of Dynamic Basketball Player Networks

Baofeng Chang, Guodao Sun, Sujia Zhu, Qi Jiang, Wang Xia, Jingwei Tang, and Ronghua Liang

***Abstract*—Visual analysis has been increasingly integrated into the exploration of temporal networks, as visualization methods have the capability to present time-varying attributes and relationships of entities in an easy-to-read manner. Visualization techniques have been employed in a variety of dynamic network datasets, including social media networks, academic citation networks, and financial transaction networks. However, effectively visualizing dynamic basketball player network data, which consists of numerical networks, intensive timestamps, and subtle changes, remains a challenge for analysts. To address this issue, we propose a snapshot extraction algorithm that involves human-in-the-loop methodology to help users divide a series of networks into hierarchical snapshots for subsequent network analysis tasks, such as node exploration and network pattern analysis. Furthermore, we design and implement a prototype system, called DBNetVizor, for dynamic basketball player network data visualization. DBNetVizor integrates a graphical user interface to help users extract snapshots visually and interactively, as well as multiple linked visualization charts to display macro- and micro-level information of dynamic basketball player network data. To demonstrate the usability and efficiency of our proposed methods, we present two case studies based on dynamic basketball player network data in a competition. Additionally, we conduct an evaluation and receive positive feedback.**

***Index Terms*—Dynamic basketball player network, dynamic network analysis, network visualization, hierarchical snapshot.**

I. INTRODUCTION

VISUALIZATION has been one bridge to connect people's requirement of data analysis with raw, complex, and large-scale data, especially in current background of information explosition. Visualization has a capacity of transforming structured/non-structured data into visualization charts, which can convey data information effectively and sensibly in vision channel [6]. Utilizing visualization charts to convey the information extracted from raw, complex, and large-scale data can markedly reduce the burden of people's memory, cognition, and computation in the process of data analysis. In recent years, visualization technique has been utilized in many domains, e.g., social media analysis [9], academic citation analysis [59], and financial transaction analysis [27]. In particularly, visualization technique plays a significant role in analysis of dynamic network data, since that the numerical networks, time-varying network topology, and multidimensional attributes deepen people's analysis burden.

Dynamic network data is a type of data model, regarding as a temporal networks to describe time-changing nodes and their relationships/connections [22]. In a dynamic network, a node can be encoded as any entity flexibly while a link can describe connection/relationship between two nodes. With the development of data collection and extraction techniques, analysis of dynamic network data has emerged in many realistic domains, e.g., the management of traffic network [67], hot spots analysis of social media network [30], and patterns exploration of financial transaction network [44]. To facilitate the analysis of dynamic network data, many researchers have integrated visualization techniques into this data. Dynamic network visualization techniques are researched in many aspects, e.g., animating network topology [15], [38], encoding network elements [3], [65], providing an overview [8], [23], and comparing topology structures [24].

In recent years, dynamic network visualization plays an important role in sport analysis such as evaluate player performance [11] and analyze team tactics [60], [62]. It is integrated into dynamic multi-player network data to present overview patterns and detailed information in soccer domains [49], [62]. Visualizing and analyzing dynamic basketball player network data is becoming an effective tool to obtain the player tactics and mobility patterns. However, current visualization works perform poorly in analysis of dynamic basketball player network data, since it consists of the following data features including numerical networks, intensive time stamps, subtle changes, multi-dimensional attributes, and domain background knowledge. The numerical networks makes it hardly possible to visualize and analyze all the networks peremptorily. The intensive timestamps and subtle change make it unnecessary to visualize two networks which are neighbouring and similar. In addition, how to effectively derive and present the multidimensional attributes of dynamic network data is still an under-researching point. In summary, the visual analysis of dynamic basketball player network data needs to be researched comprehensively and deeply.

To fill this gap, we present this work for visual analysis of dynamic basketball player network data. To improve the analysis efficiency of dynamic basketball player network data, we propose an interactive snapshot extraction algorithm to extract hierarchical snapshots of networks. The algorithm integrates a Human-in-the-Loop operation flow to extract snapshots of dynamic basketball player network data by considering the network change indicators, i.e., node change, link change, time gap, and users-defined parameters. In addition, we propose

Baofeng Chang is with Zhejiang University of Technology and Zhejiang Airport Innovation Institute, China. Email: baofeng.chang@zjut.edu.cn. Guodao Sun, Sujia Zhu, Qi Jiang, Wang Xia, and Jingwei Tang are with Zhejiang University of Technology, China. Email: guodao@zjut.edu.cn, zhusujiashiny@gmail.com, jiangqi@zjut.edu.cn, xiawang@zjut.edu.cn, and jwtang@zjut.edu.cn. Ronghua Liang is with Zhejiang University of Science and Technology, China. Email: rhliang@zjut.edu.cn. Guodao Sun are corresponding author.

Manuscript received xx xx, 20xx; revised xx xx, 20xx.

a complete visual analysis workflow, including three steps: feature computation, snapshot extraction, and visualization. To help users analyze dynamic networks expediently, we design a visual analysis prototype system, named DBNetVizor, for visualizing the extracted dynamic networks and snapshots of dynamic basketball player network data. DBNetVizor is equipped with multiple visualization diagrams to display dynamic networks including macro-level network overview, micro-level network details, and hierarchical snapshot extraction, etc. DBNetVizor offers many efficient interaction methods to support interactive analysis of dynamic basketball player network data. To illustrate the usability and efficiency of DBNetVizor, we introduce the case study based on dynamic basketball player network data in a competition. The main contributions of this work are summarized as follows:

- We propose an interactive snapshot extraction algorithm, which involves the degree of network change and user-defined threshold to extract hierarchical snapshots of dynamic basketball player network data.
- We present a complete workflow in the visual analysis of dynamic basketball player network data, regarded as a three-step pipeline including feature extraction, snapshot extraction, and visualization.
- We design a visual analysis prototype system (DBNetVizor), which help the analysis of dynamic basketball player network data by providing interactive snapshot extraction operation flow and displaying macro- and micro-level network information.

The rest structure of this paper is as follows: Section II introduces the related works about dynamic network visualization, temporal attribute visualization, and sport data visualization. Section III describes the dynamic basketball player network data, requirements, and pipeline of this work. Section IV illustrates the approaches of this work including network feature extraction and snapshot generation. The visual design of DBNetVizor is elaborated in Section V and the case study based on dynamic basketball player network data is presented in Section VI. The Section VII demonstrates the evaluation of this work. The discussion and future work constitute Section VIII, and the conclusion is written in Section IX.

II. Related Works

A. Dynamic Network Visualization

Dynamic network visualization has been researched for many years to facilitate the analysis of dynamic network data. Many surveys are already conducted by researchers from many aspects such as visual designs [26], task taxonomy [27], community discovery [37], and topology structures [19], [50]. However, to refer to enlightening visualization methods for dynamic dynamic basketball player network data, we conduct a comprehensive review of dynamic network visualization.

Dynamic network visualization techniques are utilized to analyze many kinks of dynamic network datasets, e.g., social networks [7], [9], traffic networks [51], [53], academic networks [31], [59], [65], and sport player networks [49], [62]. The traditional visualization method can be expanded from two basic diagrams: node-link diagram [6] and adjacency matrices [46], [66]. Visualizing dynamic network data with a series of node-link diagrams [16], [17], [47] and matrix-based diagrams [3], [49], [64] facilitate people to analyze data hierarchies, evolution, and patterns. With the increasing scale of dynamic network data, researchers propose many visualization techniques network animation [2], [15], [38], parallel-based node-link diagram [7], [18], network snapshots [8], [47], network navigation [24], [29], network projection [17], set-based network [40], network clustering [22], [51], and hyper network visualization [35], [45]. These methods present dynamic network data in new forms or visual encodings. For example, Bach et al. animate the networks to reduce display space, transferring time for space [2]. Dang et al. propose to display networks in parallel axes to visualize large-scale networks in limited visualization space [18]. In addition, many researchers employ an overview [16] or hyperdiagram [45] of dynamic networks to improve analysis efficiency. The network analysis methods and theories, such as link prediction [72], [74] and community detection [8], [9], [73], and network layout [75], are considered in the dynamic network visualization to improve the effectiveness of analysis.

However, current dynamic network visualization techniques cannot cover dynamic dynamic basketball player network data with features like large scales, subtle changes, dense timestamps, and abundant temporal attributes, even though it has been researched for many years. To fill this gap, we propose a complete workflow and design a prototype system to visualize dynamic basketball player network data.

B. Dynamic Network Snapshot Extraction

The traditional visualization technique for large-scale dynamic network data is to extract and display snapshots to present important information about dynamic networks to reduce the burden of perception and analysis. To help us extract snapshots of dynamic basketball player network data, we comprehensively summarize the snapshot extraction methods for dynamic network visualization.

The snapshots of dynamic network data can be classified based on the hierarchies: single-time granularity snapshots (STGS) [47] and multiple-time granularity snapshots (MTGS) [8]. Sampling or aggregating raw dynamic network data uniformly to construct STGS is too simple to present important information accurately. To fill this gap, nonuniform time-slicing methods are proposed to generate snapshots based on the change degree of nodes or links. For example, selecting the network when its nodes/topology changes as the snapshot of raw dynamic network data [71]. Wang et al. proposed a nonuniform time-slicing method to generate snapshots with balancing visual complexity of dynamic network data [70]. In addition to the aforementioned methods, selecting dynamic networks interactively to build snapshots [23], [47] is another method to extract STGS. However, compared with MTGS, STGS can not help users analyze dynamic network data from different hierarchies. Consequently, in recent years, many researchers have proposed novel methods to generate MTGS of dynamic network data. For example, Cakmak et al. propose a recursive combination to form multiple granularity snap-

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3423721

shots [8]. Arleo et al. proposed a multilevel method to present event-based dynamic networks in different granularity [69].

To generate rational snapshots of dynamic basketball player network data, we propose an interactive and hierarchical snapshot extraction algorithm considering the degree of network change and integrating the Human-in-the-Loop principle. Users can customize the network change thresholds freely and our proposed algorithm will generate the hierarchical snapshot of dynamic network data based on the customized thresholds.

C. Temporal Attribute Visualization

Dynamic network data has lots of temporal attributes, which are visualized to help users access data insights. In dynamic basketball player network data, there are many temporal attributes. And the experiment data is actuall a kind of temporal data. To draw lessons from excellent temporal attribute visualization methods, we conduct a review of the temporal attribute visualization.

Temporal attribute visualization is already researched in many scenes such as spatio-temporal data visualization [41], [42], social media visualization [43], [57], and event sequence visualization [32], [60], [65]. For example, visualizing the temporal attribute of urban traffic networks by sequential diagrams can help users access the temporal traffic patterns [25], [53]. In social media visualization, researchers propose to visualize temporal attributes with time-varying density diagrams for helping users figure out and compare the massive movement patterns [28]. In addition, visualizing temporal attributes with a line chart can provides implicit evolution patterns for many analysis requirements such as biology system evolution analysis [34], network structure evolution analysis [16], [23], and group dominance analysis [14].

Visualizing temporal attributes with a sequential diagram such as the line chart, density network, and circular diagram is a traditional method [10]. As a result, in this work, we visualize temporal attributes and indicators of dynamic basketball player network data to facilitate the patterns exploration and insights anslysis.

D. Sport Data Visualization

The experimental data, dynamic basketball player network data, is full of sport data features. To present the sport information rationally and help users analyze dynamic basketball player network data effectively, we conduct a review in sport data visualization.

Existing sport visualization works mainly focus on sport data analysis, event data mining, and strategy exploration, etc [12]. For instance, Chen et al. design a visual analysis system GameFlow by presenting diverse data for analyzing game of National Basketball Association (NBA) [11]. ForVizor, a system integrating multiple diagrams, is proposed to visualize spatio-temporal team formation changing information [60]. SoccerStories is proposed to analyze the kick-off events in the soccer court [36]. In addition, a graph-based Sankey plot is proposed to visualize the evolution patterns of player's movement in a soccer game [49]. In racket sport, researchers have conducted many works in recent years. Many visualization methods are proposed to analyze racket sport data such as mine tactic pattern of table tennis [52], [56], [58] and explore badminton tactics with an immersive technique [13].

However, current sport visualiztion works is challenging in visualizing dynamic basketball player network data intuitively and interactively. To fill this gap, we design a visual analysis system, which integrates an multiple interactive visualization diagrams and many data maining algorithms, to visualize dynamic basketball player network data.

III. Requirements Analysis and Pipeline

A. Requirements Analysis

To strengthen the usability of this work, after several turns of discussion and reviews, we have summarized the system requirements based on the suggestion of visualization experts, our visualization experience, and data features. In summary, 4 requirements (**R1-R4**) are summarized as follows.

- **R1 Providing overview of dynamic basketball player network data.** Displaying the whole networks burden the people's memory, cognition, and computation [61]. In particular, when dynamic network data consists of numerical networks and multi-dimensional attributes, providing overview to support the analysis of dynamic network data is important. Thus, providing an overview of dynamic basketball player network data is necessary.
- **R2 Combining network and snapshot details to support collaborative analysis.** Dynamic basketball player network data contains time-varying topology structures and attributes information. Just displaying topology structures of dynamic basketball player network data, ignoring the node and link attributes, is inefficient for obtaining data insights [34]. Therefore, combining extra network details (e.g., temporal attributes) is vital for analyzing dynamic basketball player network data.
- **R3 Supporting Human-in-the-Loop principle to extract hierarchical snapshots.** To improve the analysis efficiency, extracting hierarchical snapshots of dynamic basketball player network data is essential. Fixed snapshots cannot satisfy the analysis tasks of dynamic basketball player network data [8]. For example, focusing topology evaluation analysis may only based on node and link change instead of time-varying attributes. The human subjective factor is very important in the process of analyzing complex tasks. As a result, integrating the Human-in-the-Loop principle in hierarchical snapshot extraction is significant in improving the usability of this work.
- **R4 Integrating rational visual design and interactive operation for analysis.** In almost scenarios, visualization techniques need to be rationally designed considering the tasks and data features. For example, employing traditional node-link diagrams performs poorly in helping users obtain insights of dynamic basketball player network data when the data has a large number of nodes and links [35], [45]. Consequently, integrating rational visualization techniques is a vital requirement for analyzing dynamic basketball player network data.

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3423721

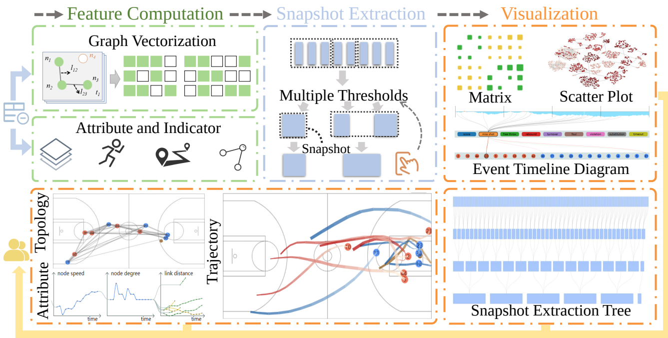


Fig. 1. The pipeline of this work consists of three steps: Feature Computation, Snapshot Extraction, and Visualization.

B. Pipeline

In response to aforementioned requirements **R1-R4**, we propose a visual analysis pipeline (Fig. 1) for visualizing dynamic basketball player network data. The pipeline consists of **1) Feature Computation**, **2) Snapshot Extraction**, and **3) Visualization**.

1) Feature Computation. In this part, we collect the dynamic basketball player network data in a competition as the dynamic network data with data features including numerical networks, intensive time stamps, subtle changes, and multi-dimensional attributes. After data processing, we construct a combined vector consisting of node vector and link vector for each network (Section IV-B1). We also extract network attributes and indicators (e.g., player's class, position, speed, degree, link's distance, stability, and network stability in dynamic basketball player network data) to help users access the data insights (Section IV-B2).

2) Snapshot Extraction. In this part, we propose an interactive snapshot extraction algorithm considering the multiple aspects of network change (Section IV-C) and user-defined thresholds. At first, the change degrees of network are computed based on the combined network vector. Then, the computed change degrees of network is compared with user-defined threshold to judge merging snapshots or not. In the end, users are allowed to preserve, delete, re-extract, and extract next-level snapshots interactively. By this algorithm, hierarchical snapshots are generate interactively to support focus+context visual analysis of dynamic network data.

3) Visualization. In this part, we design an interactive web-based prototype system, named DBNetVizor, for the visual analysis of dynamic basketball player network data (Section V-B). Compared with traditional or purely manual analysis methods (e.g., viewing video playback), the system can improve the efficiency of analysis, process a large amount of basketball player network data at one time, and display detailed information at the same time. As shown in Fig. 6, DBNetVizor integrates multiples diagrams to offer users the macro-level overview and micro-level details of dynamic basketball player network data. In addition, DBNetVizor are equipped with several algorithms and interaction techniques to help users obtain insights into basketball dynamic networks.

IV. Approaches

A. Data Description

A static network can be modeled as $N = (V, E)$, where V (vertices/nodes) represents objects while $E \subseteq V \times V$ (relations) represents edges/links. Dynamic network data can be modeled as $\Gamma = [N_1, N_2, ..., N_n]$, where the $N_i = (V_i, E_i)$ represents the network in timestamp t_i [5].

In this work, the dynamic basketball player network data is based on an NBA competition which was played by New Orleans Pelicans (NOP) and Golden State Warriors (GSW) on October 27th, 2015. It contains 22 nodes ($V_1 - V_{22}$) and 231 links ($E_1 - E_{231}$). Every node symbolizes a player, and every link denotes a relationship between two nodes. When the following guidelines are applied to links: (1) Every player in the data has two fixed links that are used to connect him to his closest opponent and teammate on the court; (2) A link is established between two players if their Euclidean distance is less than 1.0 meters; (3) A link is established between the ball and the closest player on the court.

The dynamic basketball player network data consists of 8243 networks ($\Gamma = (N_1, N_2, ..., N_{8243})$) over time leading to a time- and labor-consuming analysis work. The *intensive timestamp* and *subtle change* are represented in that its sampling time is about 0.35 seconds and the variation between two sequential networks is subtle leading that analyzing similar sequential networks is ineffective. The *multidimensional attributes* illustrate that it contains many attributes such as the class of nodes and the position of the nodes in the dynamic basketball player network data.

B. Feature Computation

1) Network Vectorization: Each network/snapshot consists of the nodes and links with a time information. To qualify the network, we construct a combined vector, which includes node vector and link vector, for each dynamic basketball player networ data.

The vector of network should have two functions: (1) it indicates the network topology, i.e., the state of nodes and links, and (2) it is explainable and computable. As a result, we propose a network vectorization method to heuristically construct a vector of a dynamic basketball player network data. As shown in Fig. 2 (a), a dynamic network data has four nodes (n_1, n_2, n_3, and n_4). In the network (t_1), n_1, n_2, and n_3 are *existent* while the n_4 is *nonexistent*. To describe the state of node in this network, we construct a node vector ($Node_{vec}$) with the encoding rule: *0*→nonexistent and *1*→*existent*. As a result, the $Node_{vec}$ of this network is $[1, 1, 1, 0]$. The links l_{12} and l_{23} are existent and the links in this network can be represented as a symmetric matrix by the same encoding rule of nodes. To reduce the memory space, we just flatten the upper triangle matrix to form a link vector ($Link_{vec}$) in this network. Thus, the $Link_{vec}$ of this network is $[1, 0, 0, 1, 0, 0]$. In the end, the vector of this network can be formed by combining $Node_{vec}$ and $Link_{vec}$ to be $[1, 1, 1, 0, 1, 0, 0, 1, 0, 0]$. The vectorization method can be used for a snapshot, since it can be regarded as a static network. For a dynamic basketball player network data, the

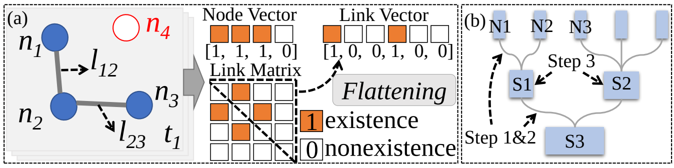


Fig. 2. (a) A combined vector (node vector and link vector) of a network. (b) Three steps of hierarchical snapshot extraction.

value 1 in the network vector indicates that one player or one relation between two players is exist.

Discussion: We heuristically propose a vectorization method to construct feature vector for each graph as well as snapshot. The constructed feature vectors is utilized for subsequent snapshot extraction and visual analysis. This method describes and qualifies the status of nodes and links in certain timestamp, following the inspiration from rule-based graph vectorization methods [34], [47]. This heuristic vectorization method may be restricted by the increasing scale of dynamic graphs. Because the larger graph correspond with the larger vectors. In addition, comparing with modal-based vectorization method, our rule-based method may ignore the deeper features and insights of dynamic graphs. However, to better describe and qualify the status of dynamic basketball player network data, we utilize this vectorization method to generate the vectors of nodes and links respectively.

2) Attribute and Indicator Computation: Computing attributes and indicators of dynamic networks are significant for revealing the information of dynamic network data [50]. In this work, we extract attributes (e.g., player position, speed, degree, and link distance) and indicators (e.g., link stability and network stability) to facilitate the analysis of dynamic basketball player network data (**R2**).

Player Position and Speed describe player's real-time geographic location and movement trend [36], [39]. **Player Degree** indicates the number of players' links in the dynamic basketball player network data. **Link Distance** presents the Euclidean distance [55] between two players if they form a link. **Link Stability** describes the stability of a link in networks during dynamic changes, which is determined by the node speed and link distance in dynamic basketball player network data. For instance, when two players as nodes form a link, the speed of players is faster or the distance between them is farther, the more unstable the link is. As a result, link stability is inversely related to players' speed and distance, and it can be computed as the following formula.

$$L_{stability} = \frac{1}{P^1_{speed} + P^2_{speed} + L_{distance} + \varepsilon} \qquad (1)$$

where P^1_{speed} and P^2_{speed} are the speed of two players who form the link, $L_{distance}$ is the link distance, and ε is a constant to avoid the divisor is zero.

Network stability presents the stability of one network. In this work, network stability has a specific definition. 1) if all the players do not move (i.e., speed is 0), the player network is more stable. 2) If the size of links is very large, the influence of adding or deleting a link to network stability can be ignored. 3) If the link distance is shorter, the link is more stable, which leads to the network being more stable, too. Based on the definition, it can be computed as follows.

$$N_{Stability} = \frac{\sum_{i=1}^{n} P_i^{speed}}{n} \times m \div \frac{\sum_{j=1}^{m} L_j^{distance}}{m} \qquad (2.1)$$

$$= \frac{m^2 \times \sum_{i=1}^{n} P_i^{speed}}{n \times \sum_{j=1}^{m} L_j^{distance} + \varepsilon} \qquad (2.2)$$

where m is the size of players' links, n is the size of players, P_i^{speed} is the speed of player P_i, $L_j^{distance}$ is the distance of link L_j, and ε is a constant to avoid the divisor is zero.

Based on the network vector, attributes and indicators, we propose an interactive hierarchical snapshot extraction algorithm and design a prototype system to visualize dynamic basketball player network data.

Discussion: We define and extract attributes and indicators for the visual analysis and insights exploration of dynamic basketball player network data. Fundamental attributes (e.g., position and speed) are based on realistic scenario while abstracted indicators (e.g., degree and stability) are proposed and computed from the fundamental attributes. Both of them are useful for visualizing and analyzing dynamic basketball player network data. Including aforementioned attributes and indicators, many attributes and indicators are proposed to describe dynamic network data [16], [33], which should be focused and utilized. Additionally, comparing with our method, complex modal may offer more comprehensive and insightful attribute and indicators [17], [63]. However, to better present basketball scenario-dependent information, we utilized the attributes and indicators, introduced in above, for the visual analysis of basketball player network data.

C. Snapshot Extraction

Dynamic basketball player network data contains thousands of networks, leading to displaying all the data for users simultaneously would be unsuitable because of visual clutter (perception burden) and analysis burden. To improve the efficiency of analyzing dynamic basketball player network data, we propose a snapshot extraction algorithm considering the degree of network variation and user-defined threshold. The following is a comprehensive description of the algorithm.

Based on the review of current network snapshot extraction algorithms, we propose three considerations to guide the snapshot extraction algorithm. The first consideration is *aggregating similar and sequential networks*. If the networks are similar and sequential, they can be regarded as one snapshot to be analyzed, which improves the analysis efficiency. The second consideration is *supporting user-defined network change threshold*. The extracted snapshot based on the thresholds can help users customize their interested snapshot for analysis, which can help users find diverse insights into data. The last consideration is *extracting hierarchical snapshot interactively*, which follows the **R1&R3**. Following these considerations, we propose a snapshot extraction algorithm. As shown in Fig. 2 (b), the algorithm can be divided into three steps based on dynamic networks $[N1,\ N2,\ N3,\ ...]$.

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3423721

Step 1: Computing the degrees of network change. To qualify the network changes comprehensively, the degrees of node changes ($Node_{change}$), link changes ($Link_{change}$), and time gap ($Time_{gap}$) are computed by following three formulas based on the node and link vectors (see Section IV-B1). Here, we employ network Editing Distance [54] to quantify the degree of node changes and link changes.

$$Node_{change} = \frac{||Node_{vec}^{N2} - Node_{vec}^{N1}||_1}{Node_{num}^{N1} + \varepsilon} \tag{3}$$

$$Link_{change} = \frac{||Link_{vec}^{N2} - Link_{vec}^{N1}||_1}{Link_{num}^{N1} + \varepsilon} \tag{4}$$

$$Time_{gap} = |Time_{start}^{N2} - Time_{end}^{N1}| \tag{5}$$

where $Node_{vec}^{N1}$ and $Node_{vec}^{N2}$ are the node vectors of $N1$ and $N2$, $Link_{vec}^{N1}$ and $Link_{vec}^{N2}$ are the link vectors of $N1$ and $N2$, $Node_{num}^{N1}$ and $Link_{num}^{N1}$ are the size of nodes and links in $N1$, $Time_{end}^{N1}$ and $Time_{start}^{N2}$ are the end time and start time of $N1$ and $N2$ respectively, and ε is a teeny constant to avoid that the divisor of the formula is zero.

Step 2: Comparing the change degrees with user-defined thresholds. The change degrees of nodes ($Node_{change}$), links ($Link_{change}$), and time gap ($Time_{gap}$) between $N1$ and $N2$ are computed in Step 1, then these values will be compared with user-defined thresholds of node change (λ_{node}), link change (λ_{link}), and time gap (λ_{time}). If all the computed values are lower than the user-defined thresholds simultaneously, which can be described by $Node_{change} \leq \lambda_{node}$ and $Link_{change} \leq \lambda_{link}$ and $Time_{gap} \leq \lambda_{time}$, the merge condition will be regarded as **True**, and $N2$ will be spliced to $N1$ to form a new snapshot $S1$. Then, our algorithm will loop the Step 1-2 to merge $S1$ with subsequent network (i.e., $N3$). If the merge condition is **False**, the $N1$ will be regarded as a snapshot $S1$. Then, the algorithm will loop Step 1-2 to attempt to merge $N2$ and $N3$.

Step 3: Preserving, deleting, or re-extracting hierarchical snapshots. Once the snapshot extraction algorithm is excuted, the networks will be extracted to be snapshots. Then, attributes and indicators of extracted snapshots (Section IV-B2) are computed and visualized to help users judge to preserve, delete, or re-extract the snapshots (**R3**). If users want to extract more high-level snapshots, the extracted snapshots will be reagrded as networks in our algorithm. By using this algorithm, users can focus on the POI (points of interest) of snapshots and conduct a further focus+context analysis of dynamic basketball player networks. This algorithm is integrated in an interactive diagram, as a snapshot extraction method, to help users operate snapshot extraction freely and intuitively, which improve the analysis efficiency of such large-scale dynamic basketball player network data.

Discussion: We propose a snapshot extraction algorithm (method) integrating Human-in-the-Loop principle and network change degrees to generate hierarchical snapshots dynamic basketball player network data. The method contains three main steps as well as supports a visual and interactive operation flow. However, the method is heuristic leading to its limitation on all the types of dynamic network data. For example, when users are unfamiliar with the experimental data, it is possible for users to conduct many attempts to set rational thresholds for available snapshots. In addition, the running time of our method is positively correlated with the number of networks, leading to a terrible user friendliness when extracting snapshots based on large-scale dynamic network data. For this data, which is introduced in Section IV-A, we utilize this heuristic method to help users extract hierarchical snapshots for visual analysis. In the future, we will improve this algorihtm to improve its effectiveness and validation.

V. Visual Design

A. Design Goals

In response to the requirement analysis (Section III-A) and data features, we summarize four concise design goals (**G1-G4**) to guide the implementation of visual analysis system.

G1 Visualizing hierarchies of extracted snapshots. The system should visualize the hierarchies of snapshots to help users perform snapshot analysis in multiple time granularity. During the snapshot extraction, preserving, deleting and re-extracting the dynamic basketball player networks into snapshots can be supported based on the snapshot indicators and analysis tasks (**R1-R4**).

G2 Providing macro-level overview for network data. The system should provide users with a macro-level overview of dynamic basketball player networks. In the overview of dynamic network data, users can concentrate on the POI. Then users can conduct a snapshot extraction for further exploration (**R2&R4**).

G3 Displaying micro-level details of dynamic network data. The system should display the micro-level details of the dynamic basketball player networks, i.e., network attributes indicators to help users in analyzing the hierarchical snapshots (**R3&R4**).

G4 Supporting rational interaction techniques. The visualization should integrate multiple rational interaction techniques to visualize the interested snapshots, network attributes, and network indicators, that are useful to help users access the insights of dynamic basketball player network data (**R1-R4**).

B. System Design

To facilitate the analysis of dynamic basketball player network data, we design a visual analysis prototype system, named DBNetVizor. The system is designed as a web-based prototype system and implemented by D3.js and Django framework based on the **G1–G4** which are summarized in Section V-A. As shown in Fig. 6, DBNetVizor is equipped with multiple linked visualization diagrams. The visual encoding of each diagram is introduced specifically as follows.

1) Network Matrix: Following the guideline of Utilizing a matrix as an overview to display the whole dynamic basketball player network data [6], we design a matrix diagram to display macro-level overview of networks **(G2)**. As shown in Fig. 3 (left), the nodes (players) are mapped to circles at the top and left sides, while the links (relations) are mapped to rectangles in the matrix. The colors of circles correspond to the class of nodes (i.e., players' team), while the colors of rectangles correspond to that the link occurs in the same node class (green) or the different node class (yellow). The node label, which represents the jersey number of players, are overlaid on the circles to distinguish the players. For example, as shown in Fig. 3 (left), the blue color of circles indicates the team of the players is GSW while the orange color indicates the team of the players is NOP. Specifically, the blue circle represents No.23 (GSW) while the orange circle represents No.23 (NOP). The width of the rectangle is encoded as the number of the link. As for interaction in this diagram, users can hover on each circle or rectangle, then the corresponding matrix row and matrix column are highlighted. Users can also click on each circle or rectangle, then the corresponding networks in the scatter plot (Section V-B2) are highlighted to help users obtain corresponding networks **(G4)**.

2) Network Scatter Plot: Following the guideline of employing a scatter plot as an overview to illustrate the distribution patterns of dynamic networks [34], [47], we project the dynamic basketball player network data into the network scatter plot (Fig. 6 (b)) to provide a macro-level overview for pattern analysis**(G2)**. Based on the extracted vectors (see Section IV-B1), we employ t-SNE [48] to project all the networks to a 2-dimensional plane, after considering and examing MDS [68], PCA [1], t-SNE, and UMAP [4], to support network distribution pattern analysis. As shown in Fig. 3 (right), each point is encoded as a basketball player network. The color of the point corresponds to the time information. The visual mapping of color and orientation is shown at the right top of Fig. 3 (right). For example, deeper color indicates that the network occurred at later time. To help users recognize the timeline of dynamic basketball player network data clearly, we draw a link line throughout the points. The line is always from start circle to the end circle, which is shown at the right bottom of Fig. 3 (right). The scatter plot supports zoom and brush interaction to help users browse networks distribution and focus on the POI to conduct further analysis **(G1&G2&G4)**.

3) Event Timeline Diagram: To help users analyze the related information of dynamic basketball player network data, we integrate a event timeline diagram (Fig. 6 (c)) into DBNetVizor. We integrate the PBP data [11] of the basketball competition as the event data. Each event contains information of time, team score, and player. As shown in (Fig. 4 (left)), the team scores are represented by an area chart at the top of this diagram based on the difference between the two teams. In this work, the blue area represents that GSW leads the competition, while the orange area illustrates NOP leads. In the middle of this diagram, each event type is mapped to a rectangle respectively. We use the color and label to indicate the type of the events. For example, at the middle of Fig. 4 (left), the "score" event is represented by an orange rectangle. As shown at the bottom of Fig. 4 (left), each circle represents a player. The visual mapping of the circle is the same as the network matrix (Fig. 3 (a)). For example, in Fig. 4 (left), two circles represent two players NO.30(GSW) and NO.23(GSW) respectively. The link line is to indicate the event information (i.e., player, event type, and time). The red line between player and event shows that this player is a major role in this event, while the green indicates the second role. For example, in a "score" event, which is shown in Fig. 4 (left), No.30 (GSW) plays a major role while No.23 (GSW) plays a second role, indicating that "No.30 (GSW) scores with No.23 (GSW)'s assistance". Users can click the event and players to highlight the link lines between the score line chart, event types, and players to obtain the event details **(G4)**. As for interaction and application of this diagram, users can brush the timeline for focusing on the networks and further snapshot extraction, analysis, and details exploration **(G1&G3&G4)**.

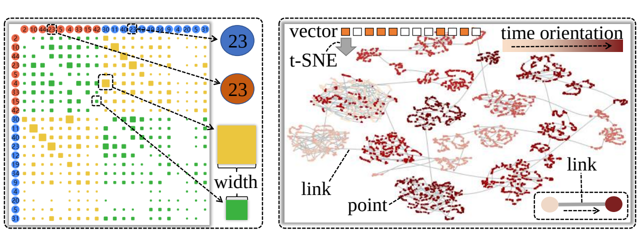


Fig. 3. The visual design of the matrix to provide an overview of dynamic basketball player network data (left) and a scatter plot to project all the networks into a 2-D plane based on network vector (right).

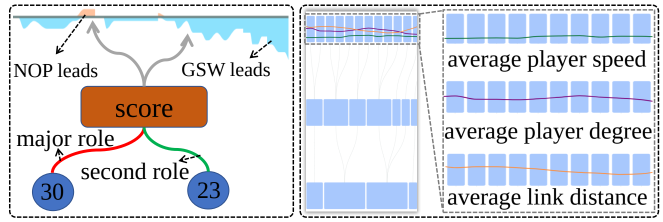


Fig. 4. A sketch of visual encoding in event timeline diagram (left) and snapshot extraction tree diagram (right).

4) Snapshot Extraction Tree: Following the guideline of visualizing the snapshot extraction, we design a snapshot extraction tree (Fig. 6 (d)) to help users extract the snapshots interactively **(G1 & G4)** to support the Human-in-the-Loop. The visual mapping of this diagram is introduced in Fig. 4 (right). As shown in the middle of Fig. 4 (right), each rectangle is encoded as a snapshot. The width of each rectangle corresponds to the size of networks, which can be regarded as the size of timestamps. The time granularity increases with the depth of the tree. The snapshot tree is interactive and the algorithm is introduced in Section IV-C. Users can extract snapshots by setting thresholds to fit different scenes. In the snapshot extraction tree diagram, users can select multi-time granularity snapshots to conduct a further snapshot analysis. In addition, we also provide the attributes and indicators visualization to help users analyze the snapshot in each layer **(G2)**. As shown in Fig. 4 (right), the green line is encoded as the average node speed of snapshots. The purple line is encoded as the average node degree of snapshots while the red line is encoded as the average link distance. The y-axis corresponds to the value of the above attributes and indicators. Users can highlight attribute and indicator lines to conduct attribute analysis and indicator analysis of snapshots. Users can select snapshots of interest for further analysis in details of snapshots in node-link diagram (Section V-B5) and snapshot

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3423721

details diagram (Section V-B6) **(G2&G4)**.

5) Node-Link Diagram: To visualize the details of dynamic basketball player network data, we design a node-link diagram (Fig. 6 (e)) to show the detailed network attribute and topology information **(G3)**. In dynamic basketball player network data, the nodes correspond to the players and the links correspond to the relationships between players. For example, the topology is shown in Fig. 5 (left), while the player's trajectory is shown in Fig. 5 (right). In this diagram, we map players to circles and the visual mapping of circles is the same as the network matrix diagram (Section V-B1). The position of a circle indicates a player's position on the field. We connect the players to show the link between the players. The width of the link is encoded as the number of this link. As shown in Fig. 5 (left), No.30(GSW) has links (i.e., l_1 and l_2) with No.23(NOP) and No.23(GSW) respectively. The width of the link is encoded as the number of this link. For example, in Fig. 5 (left), the width of l_2 is wider than l_1 indicating that the l_2 may always occur while the l_1 occurs once in this snapshot.

We show the players' mobility by segmented paths in Fig. 5 (right). One segment path is encoded as a player's movement between two merged networks. The color of the segmented path indicates the time that the deeper color corresponds to the later time in one snapshot. The width of the path indicates the speed, as the wider path corresponds to the larger speed. For example, in Fig. 5 (right), No.30 (GSW) has three-segment trajectories in the snapshot of player networks. The time order in the figure is $t_1 \rightarrow t_2 \rightarrow t_3$ and the corresponding speeds are $a \rightarrow b \rightarrow c$. He is the fastest in time t_2, while his lowest speed is in time t_1. His complete trajectory is along with $t_1 \rightarrow t_2 \rightarrow t_3$. In addition, users can highlight the path or link to avoid the visual clutter caused by showing the paths and links simultaneously **(G4)**.

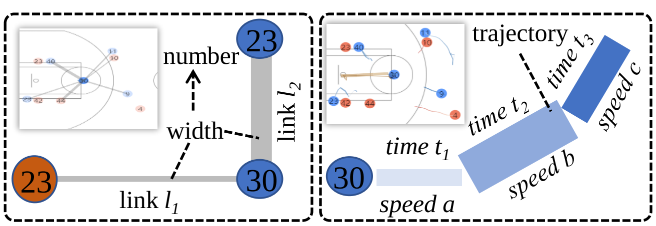


Fig. 5. A sketch of detailed link information (left) and player's movement trajectory in a snapshot (right).

6) Snapshot Details Diagram: Following the guideline of displaying the details of dynamic networks, we design a network details diagram (Fig. 6 (f)) to prensent network attributes **(G3)**. As shown in Fig. 6 (f), we map players to circles and the visual encoding of the circles is the same as the visual encoding of circles in the network matrix diagram (Section V-B1). The details of each snapshot are shown by three line charts in one row. Each line in the first and second line charts is encoded as a player while it is encoded as a link in the third line chart. All the x-axis in the line charts corresponds to the timestamps in a snapshot. The y-axis in the first line chart indicates the player's speed, while it indicates the players' degree in the second line chart. The y-axis of the third line indicates the distance of links. The colors of lines in the first and second line charts represent the player's team, while the color encoding of lines in the third line chart indicates link class, which is the same as the network matrix diagram (Section V-B1). Users can select a player to highlight the corresponding lines in all the snapshots. In addition, users can select line charts to highlight the corresponding node-link diagrams for a comprehensive analysis of dynamic basketball player network data **(G2&G4)**.

VI. Case Study

To illustrate the usability of this work, we collect a dynamic basketball player networks data based on an NBA basketball competition, which is described specifically in Section IV-A, and conduct two case studies. The case studies such as exploring the knowledge of nodes and analyzing the insights of snapshots based on this dynamic basketball player network data to validate the usability of this work.

A. Case 1: Analyzing player performance in networks

To demonstrate that our work is useful in node analysis in dynamic basketball player network data, we introduce the following study, which includes four steps, i.e., concentrating on the interesting player, selecting the targeted player networks, extracting the hierarchical snapshots, and analyzing and selecting the interesting snapshots.

Step 1: Concentrating the interesting player. As shown in Fig. 6 (c), the blue line indicates that the GSW won this competition by a big margin. Especially after the middle of second quarter, the GSW were keeping the lead until the end of this competition. We find that No.23 (NOP), that was a main star player of NOP, lost many shots, which may be one of the reasons for this failure of his team. What factors make he lose so many shots? To answer this question and help No.23 (NOP) play better, a further analysis is conducted.

Step 2: Selecting the targeted basketball player networks. In the matrix diagram (Fig. 6 (a)), the link No.23 (GSW)-No.23 (NOP) illustrates that No.23 (NOP) was mainly defended by No.23 (GSW). The highlighted points in the scatter plot (Fig. 6 (a)) present that this link often occurs. To demonstrate that the defense of No.23 (GSW) restricted the No.23 (NOP) performance, we select a part of networks through event timeline diagram. The selected player networks, which is the area shown by Fig. 6 (c1), contains intensive "miss shot" events of No.23 (NOP). If these "miss shot" were "score", the game would probably be different. Did the defense of No.23 (GSW) make him lose these shots? We conduct the following analysis by extracting hierarchical snapshots of dynamic basketball player network data.

Step 3: Extraction the hierarchical snapshots. In the above exploration, we select the networks, and the brushed area is shown by Fig. 6 (c1). The specific player networks are displayed at the first layer in the snapshot extraction tree diagram, as shown in Fig. 6 (d). To reduce the number of snapshots, we merge these player networks into snapshots with an increasing time granularity according to our proposed snapshot extraction method (see Section IV-C). In the last layer of this snapshot extraction tree, we switch the graph stability of these extracted snapshots, as shown in Fig. 6 (d1).

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3423721

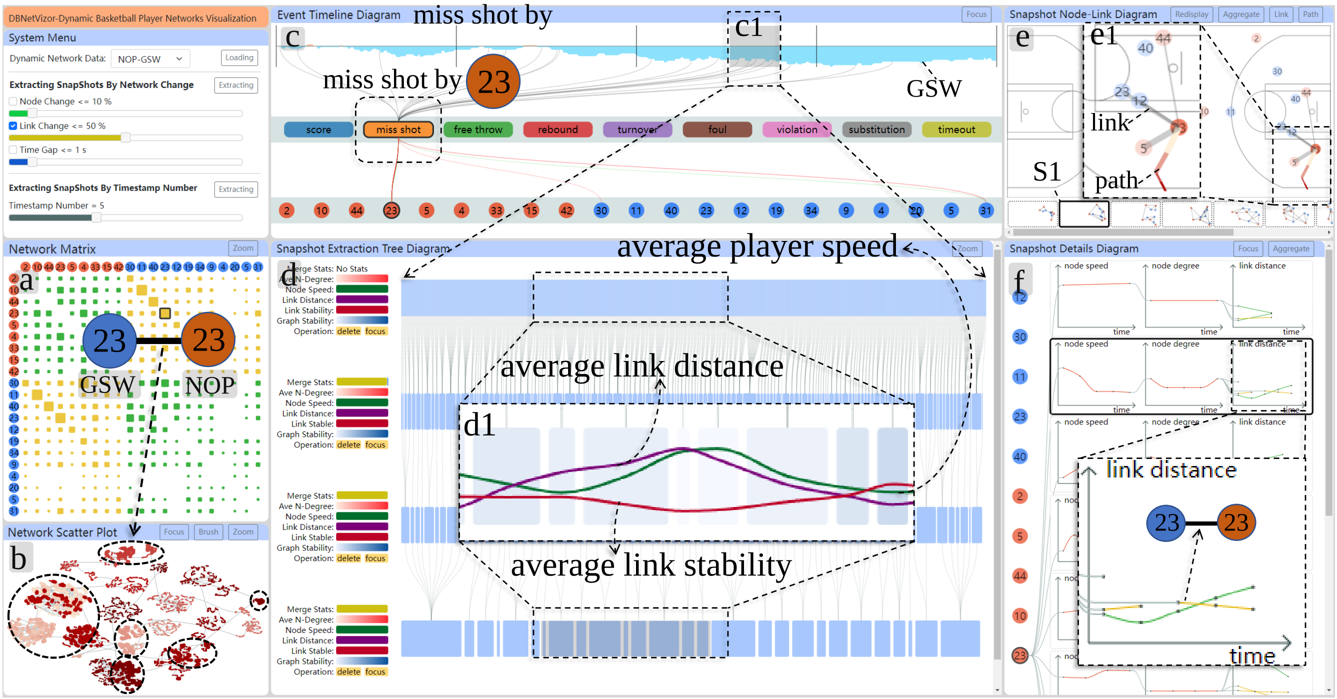


Fig. 6. Analyzing dynamic basketball player network data by utilizing DBNetVizor: (a) a matrix to provide an overview of dynamic basketball player network data, (b) a scatter plot to offer a projection based on the vector of dynamic basketball player network data, (c) a combination of the line chart and node-link diagram to integrate a additional basketball event information, (d) a tree diagram to present a snapshot extraction operation flow, (e) a node-link diagram to display the details of snapshots on the field, and (f) line charts to show players' detailed attributes.

The lighter color of the rectangles indicates that snapshots of Fig. 6 (d1) have low graph stability. The lines which overlies on the snapshots show the overall features of these snapshots.

The green line indicates that the nodes of these snapshots have an overall speed change with a low-high-low trend. The purple line indicates that the links of these snapshots have an overall distance change with a low-high-low trend. The red line indicates that the links of these snapshots have an overall stability change with a low-low trend. In addition, the green line, which is overlaid on the first layer of player networks, indicates that the nodes do not move at a fixed speed. These overall features may present that the defense of No.23 (GSW) to No.23 (NOP) is not always strong or stable. Even the defense of No.23 (GSW) against No.23 (NOP) occurs, No.23 (NOP) can get out of defense by moving. Why does No.23 (NOP) miss these shots in this period? We select extracted snapshots for detailed topology and attribute analysis.

Step 4: Analyzing and selecting the interested snapshots. The movement path and link of players are drawn with players' position data. In the selected snapshot *S1*, as shown in Fig. 6 (e1), the movement path of No.23 (NOP) indicates he has a movement to the 3-point line. At the same time, he has links with teammate No.5 (NOP) and opponents No.23 (GSW) and No.12 (GSW). The width of the link corresponds to the number of the link in the snapshots. In this snapshot, the width of the link, which represents No.23 (GSW) to No.23 (NOP), indicates that the defense of No.23 (GSW) to No.23 (NOP) does not last a long time. The detailed link distance line chart (Fig. 6 (f1)) indicates that the link between No.23 (GSW) and No.23 (NOP) has a distance change with a high-low trend in this snapshot. To conduct further exploration, we analyze players' movement path of the selected snapshot *S1* in Fig. 7 (a). We find that the movement of No.23 (NOP) in the snapshot *S1* is to catch the ball and shot it for a 3-point attempt. The No.5 (NOP) tries to delay the defense of No.23 (GSW), which is a "shield", to help No.23 (NOP) conduct a 3-point shot attempt. However, based on the players' links and movement trajectory, which is shown in Fig. 7 (b&c), we find out that No.23 (GSW) defends against No.23 (NOP), which is occurring in the subsequent snapshot *S2* of *S1*. Thus, No.23 (NOP) misses this shot at this moment.

In summary, we obtain the reason of that No.23 (NOP) loses a lot of shots in this game based on the dynamic basketball player network data. The defense of No.23 (GSW) affects the performance of No.23 (NOP). We recommend coach of NOP set tactics, e.g., "pick-and-roll", and "cuts", for No.23 (NOP) to get rid of No.23 (GSW) for easy shots.

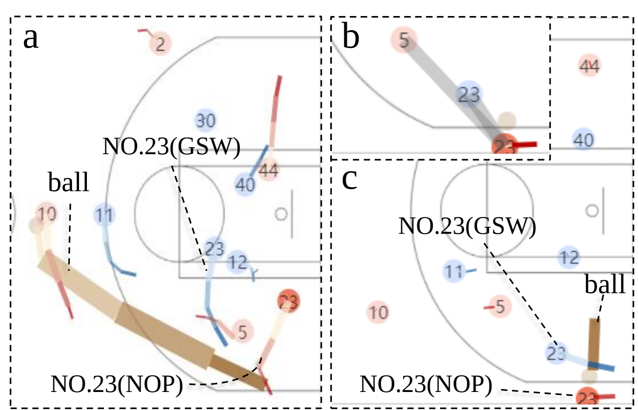


Fig. 7. (a) Players' movement path in snapshot *S1*, a detailed topology (b) and movement (c) of No.23 (NOP) in subsequent snapshot *S2*.

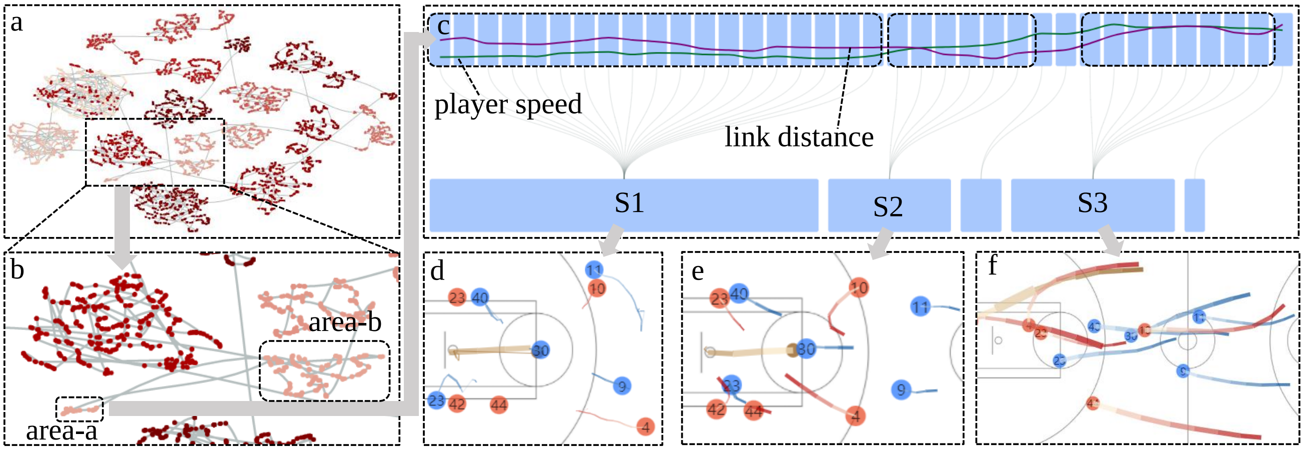


Fig. 8. Analyzing the overall characteristics of snapshots includes (a) a 2-dimensional projection of player networks based on constructed vectors, (b) an expanded view showing the *area-a* is far from the *area-b* in the projection result, (c) a tree presents the snapshot extraction result based on the player networks in *area-a*, (d-f) three players' trajectories view to introduce the overall characteristics of snapshots.

B. Case 2: Analyzing snapshot story in networks

To demonstrate that our work is helpful in analyzing the overall characteristics of dynamic basketball player network data, we introduce the following study, which includes three steps, i.e., selecting the interested networks, extracting the snapshots, and mining the overall features.

Step 1: Selecting the interested networks. As shown in Fig. 8 (a), all the player networks are projected by using t-SNE based on the constructed vectors. The color of the point is encoded as the time information of player networks. The color is deeper, the time is later. The line is drawn to link points from the first to the last basketball player network, which can indicate the time order of the networks. In the expanded projection result (Fig. 8 (b)), the circles, that is encoded as networks, in *area-a* and *area-b* has similar color indicating that they are with close time information. The line presenting the time order supports the above conclusion, too. Why are the player networks in *area-a* far away from the player networks in *area-b*? We propose a hypothesis that the player networks in *area-a* have a relatively fixed topology structure, which is different from the player networks in *area-b*. To validate this hypothesis, we select the dynamic basketball player network data in *area-a* for further analysis.

Step 2: Extracting the snapshots. A snapshot tree is displayed in Fig. 8 (c). The snapshots of the first layer are the raw dynamic basketball player network data in *area-a*, while the snapshots of the second layer are merged from the original player networks. The green line overlaid on the first-layer snapshots indicates the average player speed in these networks while the purple line presents the average link distance. The raw player networks are extracted into three main snapshots, i.e., *S1*, *S2*, and *S3*. In these snapshots, *S1* contains the most raw player networks, while *S3* has the highest average player speed and link distance. What are the overall characteristics of these three snapshots? We conduct further analysis on these three snapshots (*S1*, *S2*, and *S3*).

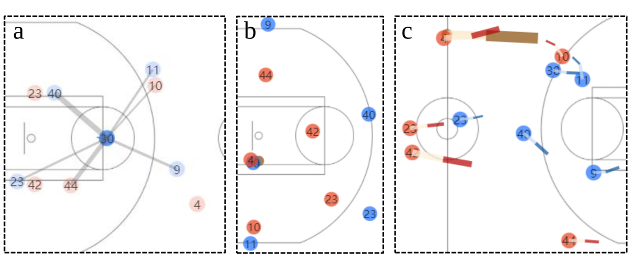


Fig. 9. (a) A star-shape links of No.30 (GSW) in the snapshot *S1*, (b) the players' trajectories before the snapshot *S1*, and (c) the players' trajectories after the snapshot *S3*.

Step 3: Mining the overall features. The detailed information of players' movement in snapshots *S1*, *S2*, and *S3* are shown respectively in Fig. 8 (d-f). In Fig. 8 (d), the players' movement path indicates that the players hardly move in the snapshot *S1*, even though *S1* has the most timestamps. We consider that this snapshot can represent a "free throw" event of No.30 (GSW). As shown in Fig. 9 (a), No.30 (GSW) has star-shaped links in the snapshot *S1*, which is a typical validation of the "free throw" event. In addition, the movement paths of the players shown in Fig. 8 (e) and Fig. 8 (f) fully demonstrate the movement state after the "free throw" event. Particularly, *S3* has the highest average player speed, illustrating the transition between the two teams after the "free throw" event. To explore the snapshot features of players' networks in the *area-b*, we find out and analyze two snapshots that exist before the snapshot *S1* and after the snapshot *S3* (Fig. 9 (b) and (c)). Fig. 9 (b) presents the players' movement before the snapshot *S1*, while the Fig. 9 (b) presents the players' movement after the snapshot *S3*. Players' position information in Fig. 9 (b) demonstrates that the "free throw" event by No.30 (GSW) may be caused by the defense of No.4 (NOP). Players' movement paths in Fig. 9 (c) are the subsequent player movement paths after snapshot *S3* (Fig. 8 (f)). This is the reason for the *area-b* are far from the *area-a* in Fig. 8 (b).

In summary, DBNetVizor is helpful to analyze a complete full "free throw" event effectively after extracting snapshots from dynamic basketball player network data and obtain the movement patterns in and after the "free throw" event.

VII. Evaluation

To evaluate this work comprehensively, we conducted a expert interview and a user study. The result validates the usability and effectiveness of this work in the analysis of dynamic basketball player network data.

A. Expert Interview

We invited three experts (E1-E3), who have researched information visualization for many years, to evaluate this work. They are experienced on dynamic network visualization and interested in basketball sports. The interview contains there main parts. Firstly, we provided E1-E3 with a comprehensive introduction to this work, including the pipeline, visual coding, case study, and interaction methods, etc. Secondly, after finishing the introduction, E1-E3 were allowed to operate DBNetVizor and propose any question freely. In the end, we asked experts for a professional evaluation. The evaluation is classified, summarized, and introduced as follows.

System Usability. Overall, experts provided positive evaluation on the usability of this work. E1 said *"The vectorization method and snapshot extraction algorithm are instructive for my tree comparison work."* He expressed that the system contained many useful interaction methods and was easy to operate when he was familiar with the system. *"The workflow is complete from the collecting basketball player network data to the analyzing it"*, E1 said. E3 expressed that the analysis flow is clear, and he would like to refer to it for his own work. E3 said *"I'm interested in the analysis of basketball networks, and I agree with the conclusions and suggestions in the case study 1."* E3 agreed with integrating the detailed information in the player basketball analysis, too. He said that the work was meaningful and should be helpful for relevant researchers such as basketball tactical analysts.

Visual Design. DBNetVizor received positive feedback on the visual design. *"The snapshot generation diagram is useful to generate the multi-layer snapshots of networks, especially with interactive extraction operation"*, E2 said. E2 presented that the overview of matrix and scatter plot made her approach to the interested dynamic basketball player network data quickly. The color mapping received a positive evaluation, too. E2 said *"Although you use so many colors to encode different information, I didn't feel confused when operating the system."*

Suggestion. Experts offered three main suggestions for this work. The first valuable suggestion was that our work should integrate more datasets and indicators into this work. E1 proposed that this work should integrate more extra basketball data such as statistic data to provide a more comprehensive analysis. In addition, E1 said *"Introducing more graph properties such as local clusters and node hierarchies will make this work more complete."* E3 suggested that we should integrate more dynamic basketball player network data, which can support users to do a deeper targeted analysis such as analyzing the performance of one certain player in different games. The second main suggestion was about networks vectorization method. E1 said *"Machine learning (ML) methods such as node2vec [21] and graph2vec [20] should be considered to construct the network vector."* In addition, E1 and E3 suggested that we should employ machine learning model to extract the snapshots of dynamic networks. E1 said *"Hot-encoding cannot express graph structure completely. And using a machine learning model can help you simplify the vector calculation."* The third mains suggestion was about the visual design. E2 concerned about the matrix view. She said *"If the dynamic network has large-scale nodes and links, the matrix and scatter plot can no longer provide an effective overview for users."* She suggested that we should employ hyper-graphs to replace the matrix and provide hierarchical clustering on scatter plots to improve the scalability of these two overviews of dynamic graph data.

Question	Mean	STD	Distribution
Q1. I think the system is useful for dynamic basketball network analysis.	5.18	0.85	4 5 6 7
Q2. I think the system is effective for dynamic basketball network analysis.	5.05	0.95	3 4 5 6
Q3. I think the visual design of this system is intuitive.	5.17	0.83	4 5 6 7
Q4. I think the visual design of this system is useful.	5.09	0.81	3 4 5 6
Q5. I think the snapshot extraction tree is easy to use.	5.05	1.21	3 4 5 6 7
Q6. I think the snapshot extraction tree is useful.	5.00	0.96	3 4 5 6

Fig. 10. The result of questionnaires in the user study.

B. Users Study

To demonstrate the usability of our system, we conduct a user study. We invited 22 participants (P1-P22), who are interested in basketball analysis, to join the users study . Eight of them are female and the others are male. The average age of them is 25.5 (Mean), while the standard deviation value is 2.6 (STD). The background knowledge of them includes data visualization, computer vision, natural language process, and control system. The whole study consists of four parts. Firstly, we introduced DBNetVizor to them detailedly. Secondly, the users were allowed to analyze dynamic basketball network with our system. Thirdly, after operating the system to analyze dynamic basketball player network data, we invited them to grade our system by six questions (Q1-Q6), which are shown in Fig. 10. Lastly, the users were allowed to provide any questions, suggestions, and criticisms freely. Participants took approximately half an hour (Mean = 38.5 minutes, STD = 5.6) to complete the user study. P15 took the longest time, 57 minutes, while P2 took the shortest time, 28 minutes.

Result. Each question (Q1-Q6) in the questionnaire evaluates our work though a score from P1-P22 according 7 points Likert Scale principle, which corresponds to strongly disagree, disagree, somewhat disagree, either agree or disagree, somewhat agree, agree, and strongly agree. After rating the system, we calculated the average score and standard deviation for every question. The result of questionnaire is shown in Fig. 10, the average score of every question is greater then or equal with 5, which indicates that the participants consider that DBNetVizor is helpful for analyzing dynamic basketball player network data. Q6 received the lowest score (Mean=5.00, STD=0.96), and the underlying reason is that the interaction flow may be not easy to handle. Similarly, Q5 received four "somewhat disagree" (score 3), indicating

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2024.3423721

that the users considered the interactive snapshot generation method might be too complex to analyze the data. In summary, the result of questionnaires illustrates that our work can help users analyze dynamic basketball player network data.

User Feedback. Overall, the participants provided positive feedback, which presented the usability and effectiveness of this work. They also proposed some valuable suggestions to help us improve the work. P3, P7, and P14 considered the snapshot extraction tree view did not help him a lot in analyzing the dynamic basketball player network data. P3 said, *"It needs me to judge the extracted snapshots and reset the parameters to re-extract snapshots. This confuses me."* Both P7 and P14 considered that the snapshot extraction tree view is not easy to use. P7 said, "It should include a recommendation modal to help us generate the snapshots." P4, P8, and P13 thought the interaction flow of the system was not easy to handle, tackling the analysis of dynamic basketball player network data. P8 suggested us to integrate more effective interaction methods in the system. He said, *"Reducing the interaction steps and integrating more effective methods may help me analyze the data."* The visual design of this system received positive feedback. However, P8 thought that the scatter and the event diagram should be equipped with more information to support a more quick analysis. P12 and P17 considered the system should be improved to be more concise to present calculated results directly. P16 and P21 considered that the system should integrated a video part to help users obtain the insights from the videos. In addition, the P19 suggested to add a function to compare the trajectories of different players in the system. Except the aforementioned feedback, the participants, who have data visualization and interactive data analysis background knowledge, said that they are interested this work. And they liked interactive snapshot extraction method because it may relax the analysis burden.

In the expert interview and the user study, our work received positive feedback and valuable suggestions for this work. However, with the help form experts and participants, we also found that many aspects of this work should be discussed and improved in the future. In Section VIII, we present a comprehensive discussion and future work of this work.

VIII. Discussion and Future Work

In this work, we propose an algorithm to help users extract hierarchical snapshots of dynamic basketball player network data. We present a complete workflow that involves a complete three-step pipeline. In addition, we design an interactive visual analysis prototype system, namely DBNetVizor, to help the analysis of dynamic basketball player network data. Nevertheless, the current work still has several aspects should be discussed and improved.

Network Vectorization. To quantify basketball player network data easily, we heuristically propose a network vectorization to describe the state of networks. It constructs a combined vector including node vector and link vector for each basketball player network. However, this vectorization method concentrates too much on topology information to describe extra features of networks such as node attributes and link attributes. In addition, it is ineffective for large-scale network such as large trading network and social media network since the constructed vector by it may lose much hidden features (e.g., node community) when the numbers of nodes and links is large. Comparing with machine learning vectorization methods, the vector by our method is explicable while it should contain more extra information and hidden features to improve its usability.

Indicator Computation. To analyze dynamic basketball player network data comprehensively, we compute several indicators such as link and network stability. The basic attributes and computed indicators help the snapshot extraction and the analysis of the dynamic basketball player network data. The attributes are used universally in many domains while the indicators are heuristically proposed for the data in this work. The indicators lacks of the support of theory, leading them to be too shallow to be useful in other domains, e.g., social media networks and academic citation networks. Thus, the indicators utilized in this work for the analysis of dynamic basketball player network data should be improved.

Snapshot Extraction. To improve the analysis efficiency of dynamic basketball player network data, which has numerical networks, intensive timestamp, subtle change, and multidimensional attributes, we propose an algorithm to extract hierarchical snapshots of networks. Our snapshot extraction algorithm integrating Human-in-the-Loop principle to help people extract hierarchical snapshots interactively and visually. The case studies demonstrate its usability. However, the time complexity of our proposed snapshot extraction algorithm is $O(nm)$ where the n is the number of the networks and the m is the scale of the network vector. If the n and m are too large, the algorithm will cost too much time to be user-friendly. In addition, the snapshot extraction result is up to the user-defined thresholds which may lead the extracted snapshots to mismatch users' analysis tasks. Therefore, the snapshot extraction algorithm should provide the suggestive snapshots and consider automatic and efficient snapshot extraction algorithm to help users extract the snapshots. Besides the automatic or semi-automatic snapshot extraction algorithms, the manual time-slicing method should be supported in this work. The comprehensive indicators of snapshots should be integrated to help users judge the extraction result, too.

Visualization. To support interactive analysis of dynamic basketball player network data, we design a visual analysis system (DBNetVizor), which consists of multiple linked visualization diagrams. It provides the macro- and micro-level information of dynamic basketball player network data and supports user to extract hierarchical snapshots interactively and visually. The case studies demonstrate the usability of DBNetVizor. However, its visual design should be improve to avoid visual clutter and support more comprehensive analysis of dynamic basketball player network data. For example, in snapshot extraction tree diagram, one rectangle represents a raw network in the first layer of the tree. If the number of the raw networks is too large, the limited space of this diagram makes the rectangles too little to be distinct. The algorithms in the visualization diagrams also need to be further enhanced to improve the scalability of the system. For example, we used t-SNE [48] with default parameters as the dimensionality reduction and projection algorithm in the network scatterplot. Both the default parameters as well as a single projection algorithm may result in the view not being applicable to other datasets. Therefore, finding adaptive dimensionality reduction algorithms based on artificial intelligence models is also an important solution that can improve the usability and scalability of the system. In addition, the analysis of player's movement on the court is a very important task in real world [36]. However, the visual design of presenting player's movement in the DBNetVizor will cause visual clutter when user analyze a long-time movement trajectory. Since the player's movement is complex and cyclic on the basketball court. As a result, improving the visual design of DBNetVizor is important. Integrating more analysis methods and theories, such as link prediction [72], [74] and community detection [8], [9], [73], should be considered in this work, cause they can help users analyze dynamic networks more efficiently. For example, link prediction can help users process the network evaluation more effectively, while community detection of nodes sharpens the focus of worthwhile analyses and reduces the burden of analysis.

Generalization. To analyze dynamic basketball player network data, we designed and implemented DBNetVizor, a visual analytics prototype system that incorporates multiple views and multiple algorithms. We also proposed a visual analytics workflow to well guide similar visual analytics efforts. Other researchers can learn from our methodology to carry out visual analysis of dynamic network data. Our proposed network vectorization approach as well as multi-layer snapshot extraction algorithms are of interest. However, one of the potential problems with this work is the weak generalization of DBNetVizor. The system may not be well suited to be used in other domains because it is tied to the data characteristics of dynamic basketball player network data. Improving the generalizability of our work is also a very important task.

Future Work. Based on the above discussions, the main future work can be summarize as follows. Firstly, a network vectorization method should be employed to cover comprehensive features of dynamic basketball player network data. Secondly, useful and logical network indicators should be utilized to present hidden information of dynamic basketball player network data. Thirdly, both automatic and manual snapshot extraction algorithm should be integrated to improve the efficiency of extracting snapshots from dynamic basketball player network data. Fourthly, improving the generalizability of this wok should be considered in the future. Last but not least, a more effective visual encoding should be designed to strengthen the system comprehensively.

IX. Conclusion

This work proposes a snapshot extraction algorithm to help users extract hierarchical snapshots of dynamic basketball player network data. A comprehensive analysis flow is presented from computing attributes and indicators to extracting hierarchical snapshots and visualizing the snapshots of dynamic basketball player network data. In addition, a visual analysis prototype system, named DBNetVizor, is designed to support to analyze dynamic basketball player network data. The case study based on dynamic basketball player network data and the evaluation illustrate the usability of this work.

Acknowledgments

The work is supported by Zhejiang Provincial Natural Science Foundation of China (LR23F020003, LTGG23F020005), Fundamental Research Funds for the Provincial Universities of Zhejiang(RF-B), National Key Research and Development Program of China (2020YFB1707700, 2022YFB3104800) and National Natural Science Foundation of China (61972356, 62036009). The authors thank all other participants for their joining the discussion, suggestion, and evaluation.

References

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] B. Bach, E. Pietriga, and J.-D. Fekete. Graphdiaries: Animated transitions andtemporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754, 2013.

[3] B. Bach, E. Pietriga, and J.-D. Fekete. Visualizing dynamic networks with matrix cubes. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, pages 877–886, 2014.

[4] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature Biotechnology*, 37(1):38–44, 2019.

[5] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The state of the art in visualizing dynamic graphs. In *Proceedings of Eurographics Conference on Visualization*, pages 1–21, 2014.

[6] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1):133–159, 2017.

[7] F. Beck, M. Burch, C. Vehlow, S. Diehl, and D. Weiskopf. Rapid serial visual presentation in dynamic graph visualization. In *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 185–192, 2012.

[8] E. Cakmak, U. Schlegel, D. Jäckle, D. Keim, and T. Schreck. Multiscale snapshots: Visual analysis of temporal summaries in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):517–527, 2020.

[9] S. Chen, S. Chen, L. Lin, X. Yuan, J. Liang, and X. Zhang. E-map: A visual analytics approach for exploring significant event evolutions in social media. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pages 36–47, 2017.

[10] W. Chen, F. Guo, and F.-Y. Wang. A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2970–2984, 2015.

[11] W. Chen, T. Lao, J. Xia, X. Huang, B. Zhu, W. Hu, and H. Guan. Gameflow: Narrative visualization of nba basketball games. *IEEE Transactions on Multimedia*, 18(11):2247–2256, 2016.

[12] X. Chen, L. Shen, Z. Sha, R. Liu, S. Chen, G. Ji, and C. Tan. A survey of multi-space techniques in spatio-temporal simulation data visualization. *Visual Informatics*, 3(3):129–139, 2019.

[13] X. Chu, X. Xie, S. Ye, H. Lu, H. Xiao, Z. Yuan, Z. Chen, H. Zhang, and Y. Wu. Tivee: Visual exploration and explanation of badminton tactics in immersive visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):118–128, 2021.

[14] D. Coelho, I. Chase, and K. Mueller. Peckvis: A visual analytics tool to analyze dominance hierarchies in small groups. *IEEE Transactions on Visualization and Computer Graphics*, 26(4):1650–1660, 2020.

[15] T. Crnovrsanin, S. Chandrasegaran, K.-L. Ma, et al. Staged animation strategies for online dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):539–549, 2020.

[16] W. Cui, X. Wang, S. Liu, N. H. Riche, T. M. Madhyastha, K. L. Ma, and B. Guo. Let it flow: a static method for exploring dynamic graphs. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 121–128, 2014.

[17] A. Dal Col, P. Valdivia, F. Petronetto, F. Dias, C. T. Silva, and L. G. Nonato. Wavelet-based visual analysis of dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(8):2456–2469, 2017.

[18] T. N. Dang, N. Pendar, and A. G. Forbes. Timearcs: Visualizing fluctuations in dynamic networks. *Computer Graphics Forum*, 35(3):61–69, 2016.

[19] M. T. Fischer, A. Frings, D. A. Keim, and D. Seebacher. Towards a survey on static and dynamic hypergraph visualizations. In *Proceedings of IEEE Visualization Conference*, pages 81–85, 2021.

[20] M. Grohe. word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 1–16, 2020.

[21] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[22] S. Hadlak, H. Schumann, C. H. Cap, and T. Wollenberg. Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2267–2276, 2013.

[23] M. Hajij, B. Wang, C. Scheidegger, and P. Rosen. Visual detection of structural changes in time-varying graphs using persistent homology. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 125–134, 2018.

[24] D. Han, J. Pan, C. Xie, X. Zhao, X. Luo, and W. Chen. A visual analytics approach for structural differences among graphs via deep learning. *IEEE Computer Graphics and Applications*, 41(5):18–31, 2021.

[25] X. Huang, Y. Zhao, C. Ma, J. Yang, X. Ye, and C. Zhang. Trajgraph: A graph-based visual analytics approach to studying urban network centralities using taxi trajectory data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):160–169, 2015.

[26] N. Kerracher, J. Kennedy, and K. Chalmers. The design space of temporal graph visualisation. In *Proceedings of Eurographics Conference on Visualization*, pages 1–5, 2014.

[27] N. Kerracher, J. Kennedy, and K. Chalmers. A task taxonomy for temporal graph visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 21(10):1160–1172, 2015.

[28] R. Krueger, G. Sun, F. Beck, R. Liang, and T. Ertl. Traveldiff: Visual comparison analytics for massive movement patterns derived from twitter. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 176–183, 2016.

[29] A. Lee, D. Archambault, and M. A. Nacenta. The effectiveness of interactive visualization techniques for time navigation of dynamic graphs on large displays. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):528–538, 2020.

[30] D. Liu, G. Yang, Y. Wang, H. Jin, and E. Chen. How to protect ourselves from overlapping community detection in social networks. *IEEE Transactions on Big Data*, 8(4):894–904, 2022.

[31] Q. Liu, Y. Hu, L. Shi, X. Mu, Y. Zhang, and J. Tang. Egonetcloud: Event-based egocentric dynamic network visualization. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pages 65–72, 2015.

[32] J. Magallanes, L. van Gemeren, S. Wood, and M.-C. Villa-Uriol. Analyzing time attributes in temporal event sequences. In *Proceedings of IEEE Visualization Conference*, pages 1–5, 2019.

[33] S. Murugesan, K. Bouchard, J. Brown, M. Kiran, D. Lurie, B. Hamann, and G. H. Weber. State-based network similarity visualization. *Information Visualization*, 19(2):96–113, 2020.

[34] H. Natsukawa, E. R. Deyle, G. M. Pao, K. Koyamada, and G. Sugihara. A visual analytics approach for ecosystem dynamics based on empirical dynamic modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):506–516, 2020.

[35] V. Pena-Araya, T. Xue, E. Pietriga, L. Amsaleg, and A. Bezerianos. Hyperstorylines: Interactively untangling dynamic hypergraphs. *Information Visualization*, 21(1):38–62, 2022.

[36] C. Perin, R. Vuillemot, and J.-D. Fekete. Soccerstories: A kick-off for visual soccer analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2506–2515, 2013.

[37] G. Rossetti and R. Cazabet. Community discovery in dynamic networks: a survey. *ACM Computing Surveys*, 51(2):1–37, 2018.

[38] S. Rufiange and G. Melançon. Animatrix: A matrix-based visualization of software evolution. In *Proceedings of IEEE Working Conference on Software Visualization*, pages 137–146, 2014.

[39] D. Sacha, F. Al-Masoudi, M. Stein, T. Schreck, D. A. Keim, G. Andrienko, and H. Janetzko. Dynamic visual abstraction of soccer movement. 36(3):305–315, 2017.

[40] Y. Shi, C. Bryan, S. Bhamidipati, Y. Zhao, Y. Zhang, and K.-L. Ma. Meetingvis: Visual narratives to assist in recalling meeting context and content. *IEEE Transactions on Visualization and Computer Graphics*, 24(6):1918–1929, 2018.

[41] G. Sun, B. Chang, L. Zhu, H. Wu, K. Zheng, and R. Liang. Tzvis: visual analysis of bicycle data for traffic zone division. *Journal of Visualization*, 22(6):1193–1208, 2019.

[42] G. Sun, T. Tang, T.-Q. Peng, R. Liang, and Y. Wu. Socialwave: visual analysis of spatio-temporal diffusion of information on social media. *ACM Transactions on Intelligent Systems and Technology*, 9(2):1–23, 2017.

[43] G. Sun, Y. Wu, S. Liu, T.-Q. Peng, J. J. Zhu, and R. Liang. Evoriver: Visual analysis of topic coopetition on social media. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1753–1762, 2014.

[44] Y. Sun, H. Xiong, S. M. Yiu, and K. Y. Lam. Bitanalysis: A visualization system for bitcoin wallet investigation. *IEEE Transactions on Big Data*, 9(2):621–636, 2023.

[45] P. Valdivia, P. Buono, C. Plaisant, N. Dufournaud, and J.-D. Fekete. Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(1):1–13, 2019.

[46] N. van Beusekom, W. Meulemans, and B. Speckmann. Simultaneous matrix orderings for graph collections. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1–10, 2022.

[47] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):1–10, 2015.

[48] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008.

[49] C. Vehlow, F. Beck, P. Auwärter, and D. Weiskopf. Visualizing the evolution of communities in dynamic graphs. *Computer Graphics Forum*, 34(1):277–288, 2015.

[50] C. Vehlow, F. Beck, and D. Weiskopf. The state of the art in visualizing group structures in graphs. In *Proceedings of Eurographics Conference on Visualization*, pages 21–40, 2015.

[51] T. Von Landesberger, F. Brodkorb, P. Roskosch, N. Andrienko, G. Andrienko, and A. Kerren. Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):11–20, 2015.

[52] J. Wang, K. Zhao, D. Deng, A. Cao, X. Xie, Z. Zhou, H. Zhang, and Y. Wu. Tac-simur: Tactic-based simulative visual analytics of table tennis. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):407–417, 2019.

[53] D. Weng, C. Zheng, Z. Deng, M. Ma, J. Bao, Y. Zheng, M. Xu, and Y. Wu. Towards better bus networks: a visual analytics approach. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):817–827, 2020.

[54] Wikipedia. Eidt distance. https://en.wikipedia.org/wiki/Edit_distance.

[55] Wikipedia. Euclidean distance. https://en.wikipedia.org/wiki/Euclidean_distance.

[56] J. Wu, D. Liu, Z. Guo, Q. Xu, and Y. Wu. Tacticflow: Visual analytics of ever-changing tactics in racket sports. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):835–845, 2021.

[57] Y. Wu, Z. Chen, G. Sun, X. Xie, N. Cao, S. Liu, and W. Cui. Streamexplorer: a multi-stage system for visually exploring events in social streams. *IEEE Transactions on Visualization and Computer Graphics*, 24(10):2758–2772, 2017.

[58] Y. Wu, J. Lan, X. Shu, C. Ji, K. Zhao, J. Wang, and H. Zhang. ittvis: Interactive visualization of table tennis data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):709–718, 2017.

[59] Y. Wu, N. Pitipornvivat, J. Zhao, S. Yang, G. Huang, and H. Qu. egoslider: Visual analysis of egocentric network evolution. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):260–269, 2015.

[60] Y. Wu, X. Xie, J. Wang, D. Deng, H. Liang, H. Zhang, S. Cheng, and W. Chen. Forvizor: Visualizing spatio-temporal team formations in soccer. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):65–75, 2018.

[61] L. Xie, J. O'donnell, B. Bach, and J.-D. Fekete. Interactive time-series of measures for exploring dynamic networks. In *Proceedings of the International Conference on Advanced Visual Interfaces*, pages 1–9, 2020.

[62] X. Xie, J. Wang, H. Liang, D. Deng, S. Cheng, H. Zhang, W. Chen, and Y. Wu. Passvizor: Toward better understanding of the dynamics of soccer passes. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1322–1331, 2020.

[63] J. Xu, Y. Tao, Y. Yan, and H. Lin. Exploring evolution of dynamic networks via diachronic node embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 26(7):2387–2402, 2018.

[64] J. S. Yi, N. Elmqvist, and S. Lee. Timematrix: Analyzing temporal social networks using interactive matrix-based visualizations. *International Journal of Human-Computer Interaction*, 26(11-12):1031–1051, 2010.

[65] J. Zhao, M. Glueck, F. Chevalier, Y. Wu, and A. Khan. Egocentric analysis of dynamic networks with egolines. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, pages 5003–5014, 2016.

[66] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. Matrixwave: Visual comparison of event sequence data. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, pages 259–268, 2015.

[67] Q. Zheng and Y. Zhang. Dstagcn: Dynamic spatial-temporal adjacent graph convolutional network for traffic forecasting. *IEEE Transactions on Big Data*, 9(1):241–253, 2023.

[68] G. D. Zimet, N. W. Dahlem, S. G. Zimet, and G. K. Farley. The multidimensional scale of perceived social support. *Journal of Personality Assessment*, 52(1):30–41, 1988.

[69] A. Arleo, S. Miksch, and D. Archambault. A Multilevel Approach for Event-Based Dynamic Graph Drawing.. *Proceedings of Eurographics Conference on Visualization*. pages 103–107, 2021.

[70] Y. Wang, A. Daniel, H. Hammad, M. Torsten, Y. Wu, and H. Qu. Timeslicing of Dynamic Graphs Based on Visual Complexity. In *Proceedings of IEEE Visualization Conference*. pages 1–5, 2019.

[71] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. GraphScope: parameter-free mining of large time-evolving graphs. In *Proceedings Of ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, pages 687–696, 2007.

[72] K. Shang, M. Small, Y. Wang, D. Yin, and S. Li. A novel metric for community detection. *Europhysics Letters*, 129(6):68002, 2020.

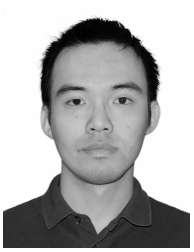
[73] J. Xu, Y. Tao, Y. Yan, and H. Lin. Exploring Evolution of Dynamic Networks via Diachronic Node Embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 26(7):2387-2402, 2020.

[74] Z. Jin, S. Guo, N. Chen, D. Weiskopf, D. Gotz and N. Cao. Exploring Evolution of Dynamic Networks via Diachronic Node Embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1343–1352, 2020.

[75] K. Steve, D. Tim, M. Kim, and W. Michael. Hola: Human-like orthogonal network layout. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):349–358.

X. Biography Section

Baofeng Chang is currently a researcher in Zhejiang Airport Innovation Institute, Hangzhou, China. He received Ph.D. degree in Computer Science and Technology and B.Sc. degree in Automation both from Zhejiang University of Technology. His main research interests are dynamic network visualization, traffic information visualization and temporal sequence visualization. Email: baofeng.chang@foxmail.com.



Guodao Sun is an associate professor at the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. He received the B.Sc. in computer science and technology, and Ph.D. degree in control science and engineering both from Zhejiang University of Technology. His main research interests are urban visualization, visual analytics of social media, and information visualization. Email: guodao@zjut.edu.cn.

Sujia Zhu is currently a PhD student at the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. Her major research interests include data mining, visual analysis and causality analysis. Email: zhusujiashiny@gmail.com.

Qi Jiang is working toward Ph.D. degree in computer science and technology in College of Computer Science and Technology, Zhejiang University of Technology. He received the B.E. degree in electrical engineering and automation from the Zhejiang University of Technology, in 2019. His research interests lie in data visualization, natural language interface for visualization, and intelligent visualization. Email: jiangqi@zjut.edu.cn.

Wang Xia is working toward Ph.D. degree in computer science and technology in College of Computer Science and Technology, Zhejiang University of Technology. Her main research interests are information visualization, image visualization, and computer vision. Email: xiawang@zjut.edu.cn.

Jingwei Tang is currently a Ph.D. student in the College of Computer Science and Technology, Zhejiang University of Technology, HangZhou, China. He received his B.E. degrees in communication engineering from Zhejiang University of Technology. His main research interests are data mining and information visualization. Email: jwtang@zjut.edu.cn.

Ronghua Liang received the Ph.D. in computer science from Zhejiang University. He worked as a research fellow at the University of Bedfordshire, UK, from April 2004 to July 2005 and as a visiting scholar at the University of California, Davis, US, from March 2010 to March 2011. He is currently a Professor of Computer Science and vice headmaster of Zhejiang University of Science and Technology, China. His research interests include visual analytics and computer vision. Email: rhliang@zjut.edu.cn.