MoNetExplorer: A Visual Analytics System for Analyzing Dynamic Networks with Temporal Network Motifs

Seokweon Jung, DongHwa Shin, Hyeon Jeon, Kiroong Choe, and Jinwook Seo

Abstract—Partitioning a dynamic network into subsets (i.e., snapshots) based on disjoint time intervals is a widely used technique for understanding how structural patterns of the network evolve. However, selecting an appropriate time window (i.e., slicing a dynamic network into snapshots) is challenging and time-consuming, often involving a trial-and-error approach to investigating underlying structural patterns. To address this challenge, we present MoNetExplorer, a novel interactive visual analytics system that leverages temporal network motifs to provide recommendations for window sizes and support users in visually comparing different slicing results. MoNetExplorer provides a comprehensive analysis based on window size, including (1) a temporal overview to identify the structural information, (2) temporal network motif composition, and (3) node-link-diagram-based details to enable users to identify and understand structural patterns at various temporal resolutions. To demonstrate the effectiveness of our system, we conducted a case study with network researchers using two real-world dynamic network datasets. Our case studies show that the system effectively supports users to gain valuable insights into the temporal and structural aspects of dynamic networks.

Index Terms—Visual analytics, Dynamic networks, Temporal network motifs, Interactive network slicing

1 Introduction

A DYNAMIC NETWORK is a data structure that captures evolving relationships between entities over time. For example, it is used to represent the evolution of social interactions among community members [1], [2] or citations between published papers [3]. Due to the temporal nature of a dynamic network, many structural patterns within a network emerge and disappear over time. Thus, it is challenging to effectively capture the patterns by naively using numeric metrics (e.g., edge density, node degree) [4].

One way to alleviate such an issue is to use network slicing, which partitions dynamic networks over time into slices (i.e., snapshots) and explores the structural patterns within each slice. The main idea behind this technique is that by independently analyzing slices at different time points, it can capture structural patterns within a particular time interval that may have disappeared over time.

Selecting an appropriate time interval (i.e., time-window size) of snapshots is crucial for the success of network slicing, as the observed patterns in the resulting snapshots can drastically vary depending on the size of the windows [5]. For instance, partitioning the data into hourly intervals may reveal differences between patterns during daytime and nighttime, while daily slicing may highlight variations between weekdays and weekends. An appropriate size of the window can differ based on both the data and tasks [6], [7], and multiple appropriate window sizes may exist. Therefore, it is essential to facilitate the exploration of different candidate time-window sizes for snapshot-based analysis.

To address this need, we utilize the concept called temporal network motifs (TNMs), which are defined as induced subgraphs on sequences of temporal edges [8]. Network motifs, including temporal network motifs, have been widely used to discover structural patterns of network data [9], [10]. We propose measures that leverage TNMs to extract the structural features of a dynamic network by considering the distribution of TNMs in a given network. These measures provide a grounded basis for finding appropriate time-window sizes.

However, although quantified measure-driven approaches [11], [12], [6], [7] exist for determining and validating adaptive window sizes outside of the visualization domain, there is still a need for human evaluation due to the inherent loss of information resulting from the numerical abstraction of networks [13]. Visual aids can effectively capture evolving structural patterns by providing a detailed depiction of structures and a high-level overview with network motifs [14], [15]. Nevertheless, existing visualization systems have shown limited interest in selecting and validating window sizes, often relying on predetermined or fixed options [16], [14].

We introduce MoNetExplorer, the first interactive visual analytics system that leverages TNMs to facilitate selecting and validating window sizes for slicing dynamic networks, enabling the identification of time-evolving structural patterns. Using our proposed measures using TNMs, MoNetExplorer recommends several window size options and depicts the advantages and disadvantages of each option. MoNetExplorer also suggests time window sizes that may contain changes in structural patterns, utilizing both TNM-based similarities between snapshots and changes in TNM-

Seokweon Jung, Hyeon Jeon, Kiroong Choe, and Jinwook Seo are with Seoul National University. e-mail: {swjung, hj, krchoe}@hcil.snu.ac.kr, jseo@snu.ac.kr
DongHwa Shin is with Kwangwoon University. e-mail: dhshin@kw.ac.kr
(Corresponding author: DongHwa Shin and Jinwook Seo.)

based measures. MoNetExplorer details the structural patterns for specified snapshots of interest, including the composition of TNMs. By analyzing the composition of motifs in the details, users can gain insights into the structural characteristics of the network. At last, users can further explore the data using familiar node-link diagram visualization. With visualization and interaction techniques designed to explore both temporal and structural aspects of a dynamic network, MoNetExplorer enables users to perform the whole process of dynamic network analysis, from selecting the window size for snapshots to making detailed observations of snapshots of interest.

To validate the efficacy of MoNetExplorer, we conduct case studies with two real-world dynamic network datasets. The results demonstrate that the system is capable of handling both directional and unidirectional network data, and can effectively detect changing patterns in the temporal occurrence of actual historical events by adaptively exploring multiple network slices.

The major contributions of this paper are as follows:

- We propose **TNM-based quality measures** for selecting and validating window sizes for network snapshot analysis.
- We design and develop **MoNetExplorer**, a visual analytics system supporting the process of dynamic network analysis from window size selection to the identification of evolving structural patterns in dynamic networks.
- We evaluate the efficiency of **MoNetExplorer** in window size selection and dynamic network analysis with two case studies and heuristic evaluation metrics.

2 RELATED WORK

2.1 Window Sizing for Slicing Dynamic Network

Selecting the proper window size of time slices, also known as the temporal resolution of a dynamic network, is crucial for understanding the evolving structure of a dynamic network [17]. Depending on the size of the window, the structure of the observed network varies [5]. Quantifying network characteristics has been an effective approach to determining the optimal window size for slicing dynamic networks [11]. Classical network structure measures capture the partial structural information of a network as numerical scores. Sulo et al. [12] proposed a TWIN (Temporal Window In Networks) algorithm for identifying window sizes with network structure measures. By balancing the reduction of noise with the loss of information on a particular measure, the algorithm can determine the best window size that reveals critical changes in the network structure. Uddin et al. [6] defined positional dynamicity as a structural change of nodes of a longitudinal network. The minimal variance of the dynamicity, which can be interpreted as minimal noise in distribution, ensures that the suggested window size is neither too large nor too small for discovering changes in nodes.

Another approach for quantifying a dynamic measure is to measure the distance between two network snapshots. Jaccard similarity is a widely used metric to calculate the distance between consecutive snapshots. Chiappori et al. [7] proposed stability and fidelity, which respectively measure the reduction of noise and the preservation of noise between two consecutive snapshots. The window size that balances the two metrics is supposed to be optimal. Orman et al. [18] applied a Jaccard similarity between three different elements of dynamic networks: node similarity, link similarity, and neighbor similarity, and utilized the noise and compression ratio proposed by Sulo et al. [12] to each similarity to determine the proper window size. However, reducing snapshots into a few numerical scores results in information loss and provides limited support for comprehending the evolving structure.

From the perspective of visualization, various window sizing has been tried to lessen the cognitive burden of users. In common, the distribution of events, which can be transferred into edges of dynamic networks, has been widely used to determine the size of windows. Arleo et al. proposed MultiDynNoS [19], which calculates a proper static window size: *autotau*, for their dynamic graph visualization technique based on the distribution of events. Wang et al. [20] and Ponciano et al. [21] proposed nonuniform time-slicing methods based on the distribution of events. Their approaches are in opposite directions: Wang et al. tried to lower the variance of events in every snapshot, which results in short window sizes in dense time periods. In contrast, Ponciano et al. consider that high-activity periods contain too much visual information, so they represented them with large window sizes. The aforementioned research aims for an optimal time-slicing strategy for visual analysis of dynamic networks. However, events are the most basic graph concept [22], therefore cannot fully represent the structural and temporal features of dynamic networks.

We propose TNMs [8] as an effective method to represent evolving structural patterns in dynamic networks. The robustness of the TNM as a measure effectively captures the structural information of evolving networks. Moreover, the interpretability and simplicity of these motifs enable human-in-the-loop exploration. A detailed description of TNMs is shown in Section 3.2.

2.2 Validation of Snapshots

Once an appropriate window size is determined, it is also crucial to evaluate the resulting snapshots to ensure their successful representation of the evolving patterns of dynamic networks. Several validation measures have been proposed to assess sliced networks in terms of minimizing noise [12], [6], [7] and information loss [12], [6], [7]. An alternative approach involves utilizing distance-based clustering algorithms to summarize temporal patterns as a sequence of clusters to which each snapshot belongs. Masuda et al. [23] tested five different distance measures and ran hierarchical clustering of snapshots with them, where clusters of snapshots are treated as states. The authors compared the state extraction performance of different distance metrics by showing state changes. Still, quantified metrics capture only partial information about network structures. Even the distance measures do not account for similar structures composed of different sets of nodes.

The process of generating and validating network snapshots involves exploring multiple potential options, which requires a visual analytics system that includes a human examination. Regarding window size selection, it is commonly accepted that there is no optimal window size for all analysis purposes [6], [7]. Similarly, validating the results obtained from slicing requires a qualitative approach using visualizations, as task-dependent effectiveness (e.g., outlier detection) is insufficiently supported by quantitative measures[12], [23], [7]. Building upon the capability of TNMs to robustly represent network structures, we developed snapshot validation measures inspired by existing research. Moreover, the interpretability and simplicity of these motifs enable human-in-the-loop exploration of various sliced results in our system. We provide a detailed discussion of snapshot validation measures in Section 4.

2.3 Visual Analysis of the Dynamic Network

Understanding the evolving patterns of dynamic networks involves comparing multiple snapshots. To visualize multiple snapshots, animation-based and timeline-based are the two common approaches employed [24]. Crnovrsanin et al. [25] proposed staged animation strategies that can visualize temporal changes of dynamic networks effectively. However, the animation-based approach, which relies on human visual memory, is not suitable for tasks that require users to compare multiple networks [26], [27].

In the timeline-based approach, a common method is to use small multiples of elementary visualizations [28]: node-link diagrams and adjacent matrices. However, their ability to convey detailed structural patterns becomes limited as the number of networks and their complexity increase the recognition of intricate structural patterns between a limited set of nodes, which can be impeded by high edge density or the prevalence of other arbitrary patterns [16], [29].

Several studies in the visualization field have attempted to address issues related to visual scalability when representing dynamic networks; however, the investigation of diverse window sizes has not been a primary focus. To address the challenge of scalability, Lee et al. [1], [30] designed an interactive visual analytic system utilizing a large display. In general, however, visual analysis takes place within a limited small display space. Researchers extracted high-level features from dynamic networks in limited display space to visualize the temporal overview. For example, GraphFlow [31] summarizes the temporal evolution of graph metrics over time. To show a structural overview, Cakmak et al. [32] utilized the distribution of 13 different static network motifs in snapshots. LargeNetVis [33] provides three information taxonomies—temporal, structural, and evolution—to achieve the visual exploration of network communities. However, most visualization approaches slice dynamic networks with fixed window sizes based on background knowledge [16], [31], [14]. Although some studies have provided a set of a few window size options [34], [35], [32] or algorithms to determine data-driven window size [33], the human role of exploring various window sizes is not sufficiently considered by design.

MeasureFlow [36] provides recommendations for window sizes based on the measure called connected nodes and its Fourier transform. It also provides visualizations and interactions to explore dynamic networks through a time series of various network measures. This approach seamlessly joins the selection of the proper window size and the visual analytics of a dynamic network. However, solely using the connected nodes measure is limited in reflecting all the complex information within the structure of the network. Further, the design for window size selection does not provide the reasoning for the recommendation.

We designed a visual analytics system capable of determining the proper window size of a dynamic network and performing validation using our novel snapshot validation measures. Furthermore, we enhanced the system to support temporal pattern recognition and detailed pattern analysis, features commonly available in other visual analytics systems. A description of our visualization design can be found in Section 5.

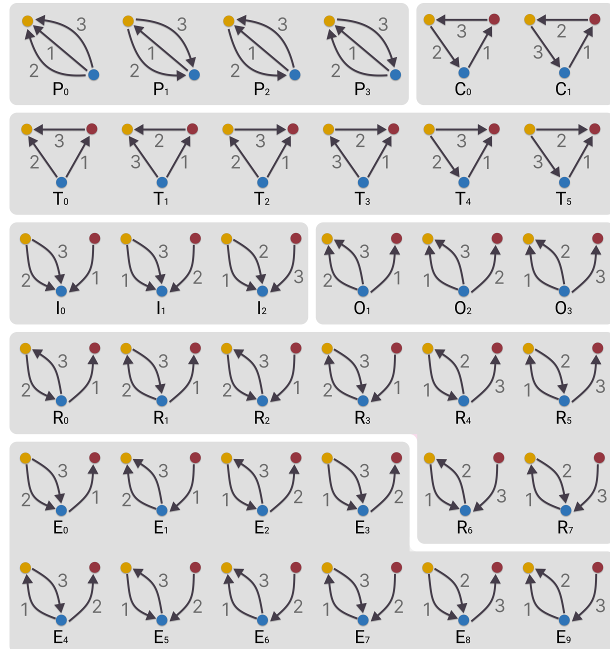

Fig. 1. The illustration of 36 different 2-node, 3-node, and 3-edge Δw motifs. Motifs can be divided into seven groups: pair (P), cycle (C), triangle (T), in-burst (I), out-burst (O), ping-pong (R), and others (E). Detailed definitions and descriptions can be found in Table 1

3 Backgrounds

3.1 Dynamic Networks

We represent a dynamic network as $N = \{V, E, T\}$, where V is the collection of nodes, E is the collection of edges between the nodes, and T is the collection of timestamps. Each edge $e^t_{ij} \in E$ is a timestamped directed edge from node v_i to v_j, denoted by (v_i, v_j, t) where $v_i, v_j \in V$ and $t \in T$. From T, we extracted a number of timestamps from T; $(t_0, t_1, ..., t_k, ..., t_{end}) \in T$ with a uniform interval w: $t_{k+1} = t_k + w$. The first timestamp begins from the minimum value of T: $t_0 = \min(T)$, and the last timestamp is equal to the maximum value: $t_{end} = \max(T)$. As an exception, the last timestamp may have an interval between 0 and w: $t_{end} = t_{end-1} + w', 0 < w' \leq w$.

A dynamic network N can be divided into a series of dynamic sub-networks $N = (S_1, S_2, ..., S_k)$. Where each sub-network S_k at the timestamp t_k consists of a set of vertices and nodes $V_{[k-1,k)}$ and a set of directed edges $E_{[k-1,k)}$ exists between timestamp t_{k-1} and the next timestamp t_k. Building upon previous studies [12], [6], [7], we treated each sub-network as a distinct dynamic network and utilized temporal network motifs to capture the relationships between them.

3.2 Temporal Networks Motifs

Temporal network motifs (TNMs) are predetermined interconnection patterns in dynamic networks considering temporal order [37], [8]. Motifs work as basic building blocks of networks [38] so that their occurrences can represent the structural and temporal characteristics of the original network. TNMs have been shown to be a viable method for classifying dynamic network data in various domains, such as social networks [39], collaboration networks [40], [41], [9], biological networks [42] and travel patterns [10]. Further, Crawford et al. [43] showed that topological similarity based on TNMs can be used for the clustering of dynamic networks.

There exist two different notions about the time duration of TNMs: Δc motifs [37] and Δw motifs [8]. The former defines TNMs as a set of events whose time difference between consecutive events is less than the threshold Δc. The latter defines TNMs as a set of events whose time difference between the last and the first events in a motif is less than the window size Δw. Δc and Δw motifs are known to have complementary features; the former fails to bound time spans, whereas the latter introduces bias for the occurrence of intermediate events [44]. We adopt Δw motifs because Δw is more suitable for use as the window size of snapshots, and the bias can be overcome with data processing.

The definition of Δw motifs can be denoted as *k*-node, *l*-edge, Δw-temporal motif is a sequence of *l*-edges that are time-ordered within a Δw duration, such that induced static graph from the edges is connected and has *k* nodes. Additionally, we limited our scope to 2- and 3-node, 3-edge motifs, limiting the number of network motifs to 36 different types. This is because extending the size of the motifs may reveal new insights, but it causes the data to have too high dimensions and increases computational complexity. For example, including 4-edge motifs would result in considering about 700 motifs.

According to Paranjape et al. [8], 2- and 3-node, 3-edge motifs are composed of three groups: pair(P), triangle, and star-shaped motifs. We divided triangle motifs into two subgroups: triangle (T) and cycle (C), according to the consistency in the orientation of the edges. We also divided the star motifs into four subgroups: in-burst (I), out-burst (O), ping-pong (R), and the others (E). The former three subgroups are distinguished according to the more detailed internal structure of motif [44]. The rest of the star-shaped motifs are classified as the others. The definition of each group can be found in Table 1, and their appearances in Figure 1.

The distribution of TNMs varies greatly depending on how nodes inside the network relate to each other. For instance, the pair and ping-pong motifs have been detected significantly in network data for social media platforms such as Facebook and SMS-A [8], [44], showing that a large number of social media users repeatedly interact with each other through retweets or replies. Bitcoin network data, which consists of consecutive transactions between crypto wallets, are rich in cycle motifs [8].

TABLE 1
The seven types of temporal network motifs and their descriptions. Images of all 36 motifs are shown in Figure 1.

Type	Not.	Description
Pair	P_i	motifs consist of two nodes and three edges
Cycle	C_i	triangle motifs whose relationships are continuous in one direction
Triangle	T_i	motifs that have relationships between all three nodes
In-burst	I_i	motifs whose three edges come into a certain node.
Out-burst	O_i	motifs whose three edges go out from a certain node.
Ping-pong	R_i	motifs whose two consecutive edges make a round trip between two nodes.
Others	E_i	motifs that are not classified.

3.3 Task Design

Based on the TNMs definition and the examples of TNMs distribution varying depending on the network structure, we determined that TNM would be available as a numerical value reflecting the structural feature of the snapshot. From this insight, we figured out four tasks in slicing and analyzing dynamic networks. The first task is grounded on the window sizing of dynamic networks, while the other three tasks are based on task taxonomy for dynamic network visualization defined by Kerracher et al.: lookup, comparison, and relation seeking [45].

- Task 1: (Window sizing) Select the appropriate window size for the analysis of dynamic network data.
- Task 2: (Comparison) Observe temporal changes in network structures of snapshots and find relations between them.
- Task 3: (Look up) Find a structural pattern to describe the snapshots.
- Task 4: (Relation seeking) Find the subsets of graph elements in snapshots related to a given structural pattern.

4 Snapshot Validation Measures

We propose three snapshot validation measures—**Motif Fidelity**, **Motif Stability**, and **Motif Clusterness**—assessing the overall suitability of window sizes. These measures leverage TNMs to validate and select appropriate window sizes that effectively partition dynamic networks.

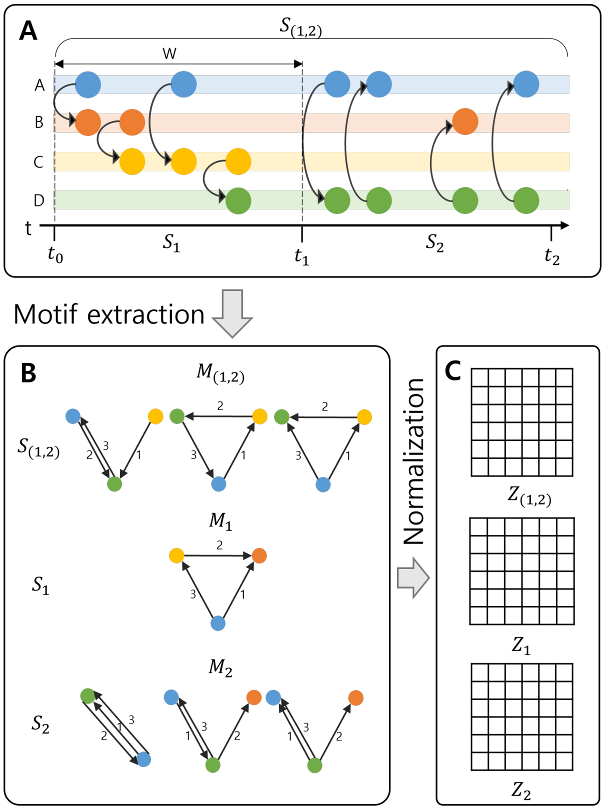

Fig. 2. Depiction of data processing. (A) Dynamic network N is sliced into snapshots: S_1 and S_2 which have the same window size (w). The concatenated result of consecutive snapshots can be denoted as $S_{1,2}$. (B) Then, motif vectors of TNMs are extracted from each snapshot. (C) Finally, the motif vectors are normalized and transformed into Z-scores. The detailed description of the process is described in Section 4.1.

4.1 Data Processing

Given dynamic network data N and snapshots $S_1, S_2, ..., S_t$ divided based on a time window Δw (Figure 2:A), we extract a global motif vector M corresponds to N, and local motif vectors $M_1, M_2, ..., M_t$, where M_k corresponds to S_k (Figure 2:B). $M_{(k,k+1)}$ refers motif vectors extracted from consecutive two snapshots S_i and S_{k+1}. We use the fast motif extraction algorithm [46] to extract motif vectors. As mentioned before, Δw motifs have a problem in that biases occur severely due to an imbalance between each vector element. In addition, the composition of motifs is not very informative in their raw form [38]. Therefore, we utilized Z-scores to overcome those obstacles (Figure 2:C). The Z-score of i-th snapshot is defined as:

$$Z_i = \frac{L1_{norm}(M_i) - \mu}{\sigma}$$

Cosine similarity has been used to calculate the similarity between z-scores of motif vectors [14].

4.2 Definitions

We propose three snapshot validation measures. Following previous research (Section 2), we design a measure estimating the amount of information preserved while slicing [12], [6], which we call **Motif Fidelity**. We also implement a measure evaluating the stability of snapshots (**Motif Stability**). At last, we propose **Motif Clusterness** representing the degree to which snapshots form clusters. The detailed design of each measure is as follows:

Motif Fidelity evaluates how much information has been preserved while slicing. High fidelity means that the information loss that occurred by slicing is small; thus snapshots well reflect the structural and temporal characteristics of the original network.

Motif Fidelity can be computed both in a global and local manner. Local similarity measures the information preservation between two consecutive snapshots. It equals the similarity between the motif composition of the sum of the snapshots and the sum of the motif composition of the snapshots. Global fidelity is defined as the weighted average of local fidelity, where weight equals the number of motifs included in local snapshots. Since the result of cosine similarity ranges from -1 to 1, we adjust the range of the result values from 0 to 1 through normalization. Formally, local fidelity between two consecutive snapshots $MF(S_i, S_{i+1})$ and global fidelity MF is defined as:

$$MF(S_i, S_{i+1}) = \frac{1 + \cos(Z_i + Z_{i+1}, Z_{(i,i+1)})}{2}$$

$$MF = \frac{\sum\{MF(S_i, S_{i+1}) \cdot |M_i + M_{i+1}|\}}{\sum |M_i + M_{i+1}|}$$

Motif Stability measure validates the quality of snapshots by measuring how stable the snapshots are. By minimizing the variance in motif composition for each snapshot, it's possible to reduce noise that occurs over time, thereby allowing for an evaluation of stability.

As with **Motif Fidelity**, **Motif Stability** is defined both globally and locally. Local stability is the similarity between two consecutive snapshots. Global stability is defined as the weighted average of local stability scores. Weight is computed based on the motif counts of each snapshot. This allows normalization with respect to the number of motifs in each snapshot. Since the result of cosine similarity ranges from -1 to 1, we adjust the range of the result values from 0 to 1 through normalization. Formally, local stability between two snapshots $MS(S_i, S_{i+1})$ and global stability MS is defined as:

$$MS(S_i, S_{i+1}) = \frac{1 + \cos(Z_i, Z_{i+1})}{2}$$

$$MS = \frac{\sum\{MS(S_i, S_{i+1}) \cdot \min(|M_i|, |M_{i+1}|)\}}{\sum \min(|M_i|, |M_{i+1}|)}$$

Note that without mentioning global or local, referring to MF and MS means the global version of the measure.

Motif Clusterness evaluates how well snapshots can be clustered. The clustering results can be regarded as states of snapshots [23]. If the states of the snapshots are well distinguished, it is easy to determine the temporal evolution of dynamic networks. Therefore, **Motif Clusterness** can be used as a proxy for performance in temporal pattern-discovering tasks.

We applied the HDBSCAN algorithm to cluster snapshots. We used HDBSCAN as it is known to be applicable to our 36-dimensional data because of its robustness to noise and high clustering performance with high-dimensional data [47], [48]. Then, we validated the clustering result with s_dbW [49] measure, as it is one of the most robust clustering validation measures [50]. The closer the s_dbW score is to 0,

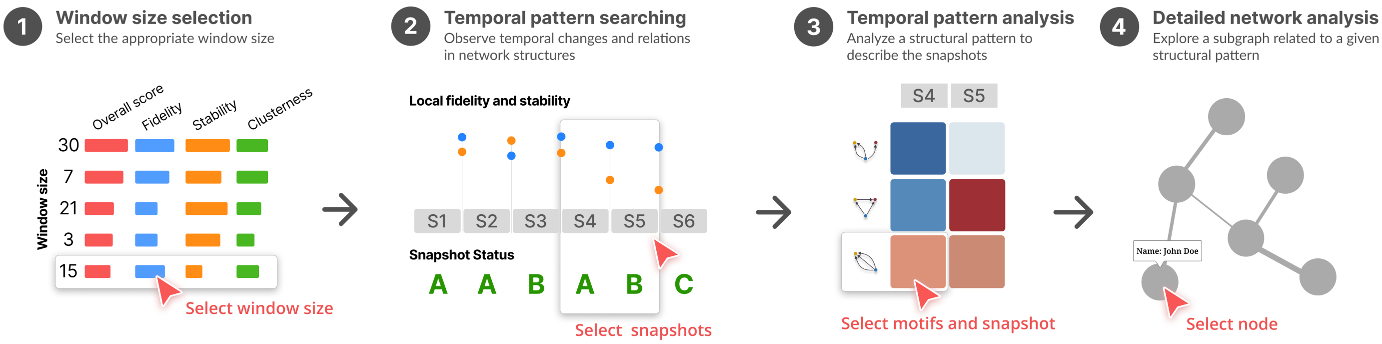

Fig. 3. The interactive workflow for identifying evolving structural patterns in a dynamic network. Users can select window size, temporal filter, and structural filter to explore multiple network slicing options. (1) Users select window size through rank-based recommendations for window size using three measures we designed from TNM. (2) Users can observe the temporal evolution of the local version of measures and states of snapshots and identify patterns of them. (3) Users undergo structural analysis of selected snapshots by choosing a specific TNM and nodes through its distribution. (4) The resulting node-link diagram provides a detailed visualization of the filtered network.

the better the performance. Therefore, we adjust the range of the result values from 0 to 1 through normalization and set the result higher the better. Formally, **Motif Clusterness** MC is defined as:

$$MC = max(1 - s\_dbW(\text{HDBSCAN}(M)), 0)$$

The suggested three measures are designed to evaluate the quality of slicing from different perspectives. Therefore, they can be an effective solution for evaluating the quality of snapshots, working as criteria for selecting the proper window size for dynamic network datasets (T1). Also, as they are based on the cosine similarity between snapshots, they can be used to figure out the relations between snapshots (T2).

4.3 Computational Complexity

Most of the computational cost of the system arises from the extraction of TNMs from the network and the clustering of snapshots. For the i-th node of the dynamic network V_i, the degree of V_i can be denoted as d_i. The average number of edges connected to the i-th node within time interval w as d_i^w, The scalable TNM counting algorithm suggested by Gao et al. [46] assumes d_i^w of all nodes is approximately equal, denoted d^w, and defines the upper bound of motif counting time complexity as $O(2d^w|E|)$ for counting pair/triangle-shaped motifs and $O(2(d^w)^2|E|)$ for star-shaped motifs. The motif extraction algorithm can be found in the following link [1]. As HDBSCAN has $O(N \log N)$ computational complexity when dynamic network data is sliced into S snapshots with w window size, the upper bound of computational complexity would be the following:

$$O(S \log S) + \sum_{i=1}^{S} O(2d^w|E_i| + 2(d^w)^2|E_i|)$$

5 System Design

In this section, we describe the workflow and tasks of our visual analytics system, MoNetExplorer, as well as its visual components: Slicing Navigation View (Figure 5.A), Temporal Measure View (Figure 5.B), Temporal Status View (Figure 5.C), Motif Composition View (Figure 5.D), and Network View. (Figure 5.E).

5.1 Workflow

One of our primary objectives is to enable the interactive selection and validation of window sizes to partition dynamic networks (T1) and identify relations between snapshots based on proposed validation measures and states (T2). Then, searching for structural patterns in snapshots with TNMs (T3) and graph attributes related to the patterns (T4) should be possible.

Before running the system, the base time unit should be selected. The base time unit is a default unit of window size, predetermined in consideration of the properties of the data and the task to be performed. Candidate window size is determined by the base time unit, which is provided in the range from the single base time unit to the next unit of base time unit. For instance, if the base time unit is determined as a month, then the candidate window sizes range from a month to twelve months.

Then, we designed our workflow (Figure 3) to achieve this objective in accordance with the given tasks and the visual information seeking mantra [51]:*"overview first, zoom and filter, and details on demand."* Our system then provides a rank-based recommendation for window sizes within candidate window sizes that demonstrates a strong fit in terms of our TNM-based metrics, showing an overview of potential slicing options. Note that each window size possesses varying capabilities in revealing distinct patterns at different time window sizes, even if it is the highly-ranked option. Thus, once a user selects a specific window size to examine, our system offers a hint of the most compatible time window by showing the plot of local TNM-based metrics along the temporal axis. This approach enables users to select a window size and determine a temporal filter for further examination. Following this, the system showcases the distribution of motifs in the resulting temporally filtered dynamic network, highlighting structural patterns. Once users identify noteworthy patterns in certain motifs, they can select them as a structural filter. Lastly, with the established window size, temporal filter, and structural

1. github.com/steven-ccq/FAST-temporal-motif

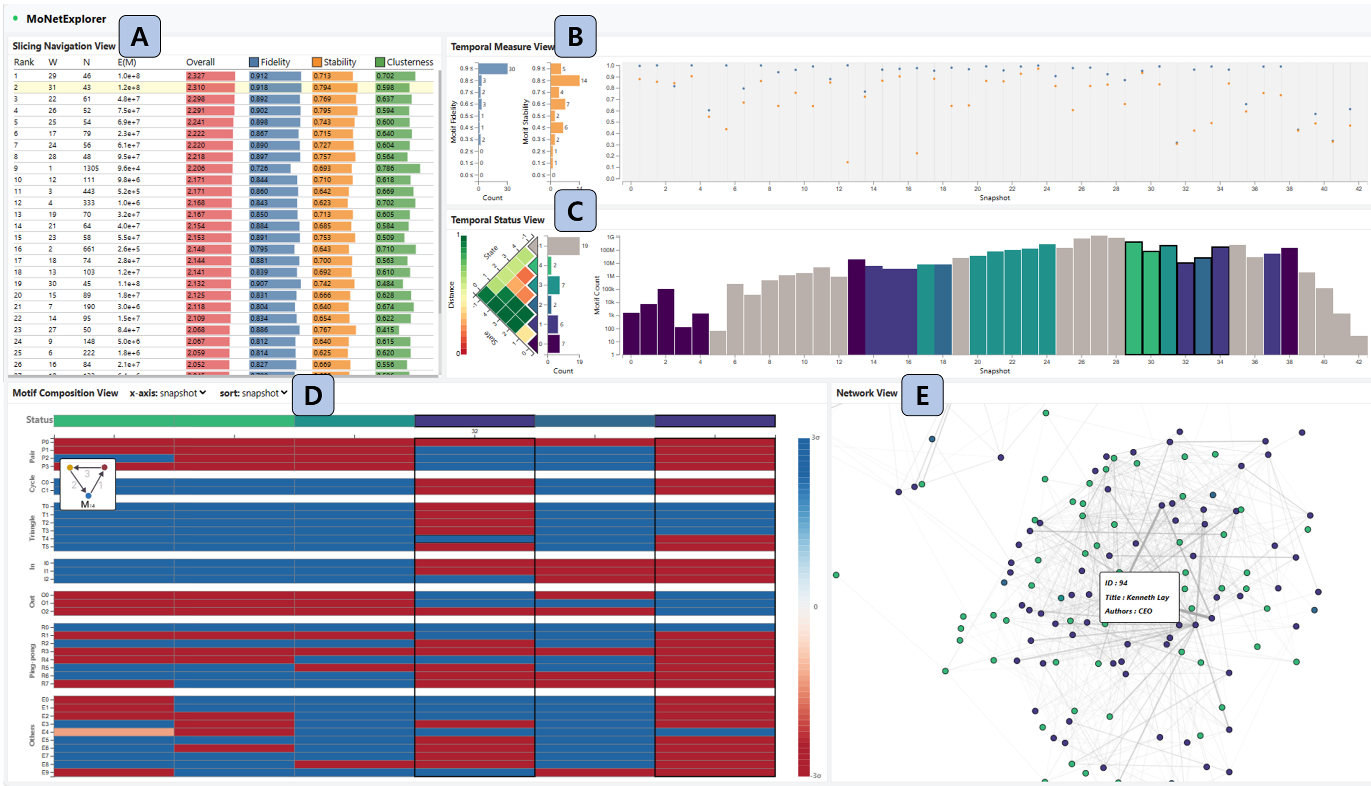

Fig. 4. MoNetExplorer is a visual analytics system designed to support the selection of appropriate window sizes for dynamic network analysis and provides a temporal and structural analysis of snapshots that are sliced according to window sizes. The system is composed of five linked components. (A) *Slicing Navigation View* supports the beginning of the workflow: selection of snapshot window sizes according to measures based on Temporal Network Motifs (TNM). (B) *Temporal Measure View* and (C) *Temporal Status View* enable validation of the quality of snapshots and identification of temporal patterns. (D) *Motif Composition View* visualizes the composition of temporal network motifs. (E) Bottom-level details of network structure are shown in *Network View*.

filter, the system presents a node-link-based view that offers comprehensive details on the dynamic network.

From now on, to describe MoNetExplorer's visualization and interaction designs, we follow the suggested workflow and demonstrate tasks with the Enron email communication network dataset [52]. Enron data are built from approximately 500,000 internal emails generated by employees of the Enron Corporation. The Federal Energy Regulatory Commission obtained it during its investigation of Enron's collapse. We utilized the network dataset constructed by Priebe et al. [4], which extracted senders, recipients, and timestamps from emails and transformed the data into a list of edges. The dataset includes 184 nodes and 108825 edges in the time range from 13 November 1998 to 21 June 2002 (44 months). The dataset can be found in the following link[2]. In order to prevent the number of time slices from becoming too large or too small, we determined the base time unit of the slicing window on a daily basis in consideration of the time period of the data. Therefore, 31 different windows became the candidate window sizes from one to thirty-one days.

5.2 Slicing Navigation View

The Slicing Navigation View (Figure 5A) facilitates proper window size selection based on three distinct measures Section 4 (Task 1). To offer recommendations based on the measures, a sortable table was created to present the window size and measure information. The table allows for metric ranking to be altered by changing the evaluation criteria through an alignment function, similar to LineUp [53] or Taggle [54]. As mentioned at the end of Section 5.1, we decided to utilize the day as a base time unit. Therefore, we have thirteen candidate window sizes: a day to 31 days. The information provided in the table includes candidate window sizes and their corresponding basic structural details, such as the number of slices and the average number of motifs per slice. Three measures—**Motif Fidelity**, **Motif Stability**, and **Motif Clusterness**—are included in the table as horizontal bar charts, with each number being mapped to a corresponding length. Each measure is partitioned into distinct colors, and this color scale is utilized in other views throughout the system. Users can select which measure to sum up to the overall score by clicking the column header of each measure (Figure 5).

Users can select the desired measures from the three provided measures and rank them based on their sums. The sum is presented as a separate bar, allowing users to select the most preferable window size overall. The number of slices and the average number of motifs per slice provide estimates of the slicing result's quality and are presented in numerical form. Adjusting the window size is crucial to ensure that these two figures are appropriate. Too many slices lead to an increased cognitive burden; conversely, too few slices make it difficult to discern notable temporal de-

2. cis.jhu.edu/∼parky/Enron/

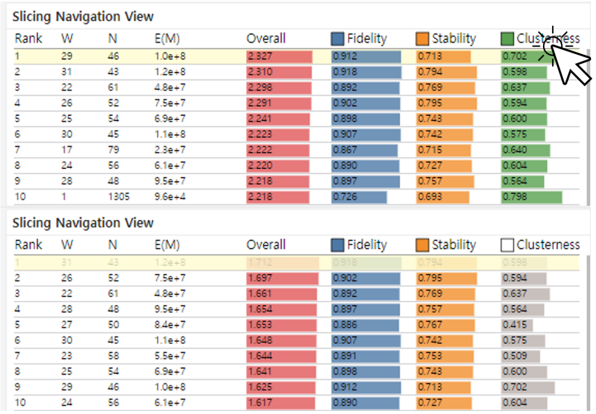

Fig. 5. The interaction technique for measure selection in Slicing Navigation View. Users can directly select which measure to use when calculating the overall score within three measures: **Motif Fidelity**, **Motif Stability**, and **Motif Clusterness**.

velopments. Likewise, when the average number of motifs is minimal, the pattern tends to be unstable, but when it's excessive, the pattern becomes overly stable.

Scenario. According to the workflow defined in Section 5.1, we first validated window sizes to determine preferable ones. When comparing ranks based on the sum of all three measures, a window size of 29 days showed the best performance (Figure 4.A). It falls to ninth place if we remove **Motif Clusterness** from the ranking criteria, and the 31-day window size comes up from second place to first place Figure 5. Therefore, we concluded 31-day is a better window size than 29-day because it ranked higher on average.

5.3 Temporal Measure View

The Temporal Measure View (Figure 4.B) provides a visual representation of the *local fidelity* and *local stability* between consecutive snapshots of a dynamic network. As the global manner of **Motif Fidelity** and **Motif Stability** can provide insight the local fidelity and stability between snapshots should remain high if the slicing is done appropriately. However, if the window sizing is inappropriate, this can result in low local fidelity and stability. Visual elements representing local fidelity and stability are colored with the color scale used in the Slicing Navigation View so that users can consistently proceed with the analysis process.

The view includes two histograms on the left side of the view, which display the snapshots' local stability and fidelity distributions (Figure 4.B). These histograms provide an overview of the quality of the snapshots, where higher values indicate greater stability and information preservation. Furthermore, the histograms can be utilized as filters, allowing users to select snapshots that fall within specific ranges by selecting the corresponding time intervals. This feature can be used to analyze the cause of low scores for particular snapshots and to adjust the window size accordingly.

On the right side, a temporal plot shows the change in both measures between consecutive snapshots. In this view, points aggregated in the histogram on the left can be observed individually. We use plots instead of a line chart to make identifying when low numbers occur easier. Users can adjust the window size in the Slicing Navigation View to explore the area more efficiently. If the local fidelity is low, it indicates that many motifs have been lost due to slicing, and the window size should be reduced. Conversely, if the local stability is low, it suggests that there is a large difference between consecutive snapshots, and the window size should be increased. For effective observation, the view supports zoom in/out and pan for exploration.

Scenario. With the selected 31-day window size, we can identify fluctuations between snapshots in MF and MS. With vertical bar charts on the left, we can see that MF remained high overall, but the MS did not. To drill down to detail with temporal plots on the right, from snapshot index 5 to 13, MF are high, but MS are low. We can understand that more aggregated slicing is needed for this period. Around 32th snapshot, both MF and MS fall dramatically compared to surrounding snapshots. From this phenomenon, we can expect that the network has undergone major structural changes at this time.

5.4 Temporal Status View

The Temporal Status View (Figure 4.C) provides a high-level overview of changes between snapshots in a dynamic network. Clusters of snapshots are computed with the HDBSCAN clustering algorithm based on the cosine distance between snapshots. Snapshots with the same cluster label can be said to have similar motif distributions. This allows us to assume that snapshots with the same cluster label exhibit similar high-level structural patterns. We can then refer to these as the state of each snapshot.

On the left side, an adjacency matrix displays the distance between states, enabling users to validate the **Motif Clusterness** by evaluating how well clustering has been performed. For efficient space usage, we render only half of the adjacency matrix to represent all the information, leveraging the symmetric nature of the cosine distance. If the distance between different states is large, it can be considered good as it means they are well distinguished. Therefore, a greater distance is indicated by the green color, and a closer distance is indicated by the red. This can be verified using the legend on the left side of the matrix. The cells in the truncated boundary where the same nodes intersect are drawn as a triangle, and then the assigned colors for each state are used to fill the cells. The adjacency matrix visualizes the states by arranging them in sorted order through PCA, with colors assigned based on the Viridis color scale proportional to the order. As an exception, states identified as noise are mapped in gray. This allows for easy comparison of the similarities between different states based on their assigned colors and indicates the proportion of snapshots that are identified as noise.

Next to the matrix, a horizontal bar chart (Figure 4.B) visualizes the number of snapshots included in each cluster. The adjacency matrix and the horizontal bar chart work as not only an overview of the states of the snapshots but also filters for them. With click interaction, users can select multiple snapshots simultaneously. On the right side, a bar chart shows the change in state and motif count over time, with each bar representing a snapshot. The height of each bar represents the motif count, and the color shows the state. By examining the change in motif count, users can identify the density of each snapshot and understand how the density changes over time. The color changes also provide insight into how the state has changed significantly, allowing users to detect trends such as change points and outliers.

Scenario. Upon examining the matrix on the left(Figure 4.C), we can figure out how many states are present and the extent of their separation. The -1 category, which denotes items classified as noise, is the most numerous, suggesting a reason for the low **Motif Clusterness** observed. State 0 and 1 are notably close to each other, while states 2, 3, and 4 exhibit similarities. A significant distinction is observed between the group comprising states 0 and 1 and the group of states 2, 3, and 4. The overall trend observed in the temporal bar chart(Figure 4.C) shows a dominance of transition from 0 to 1, followed by a frequent appearance of states 2, 3, and 2, and then a shift back to 0 and 1. A particularly interesting point noted in our measure view is around the 32nd snapshot, where there's an abrupt transition from states 2, 3, and 2 to state 1, highlighting a very significant gap. Temporal Measure View

To summarize the observation result of Temporal Measure View and Temporal Status View, the time period from the 29th snapshot to the 34th snapshot shows not only low MF and MS scores but also an abrupt transition of states. According to the external information, this time period is when the Enron stock hit an all-time high price, occurred on August 23, 2000[3] [55]. Therefore, there is a need to find out what structural patterns exist and what differences exist between states through further analysis of this period.

5.5 Motif Composition View

The Motif Composition View (Figure 4.D) presents a pixel-based heatmap visualization [56] of the motif composition of snapshots chosen in the previous step [14]. A separate row above the top of the heatmap shows each snapshot's state with the color scheme. The heatmap comprises 36 rows, each representing 36 different temporal network motifs. In order to visually distinguish the seven previously defined types of temporal network motifs, the heatmap is separated into seven groups of rows. Each cell of the heatmap shows what z-score the motif corresponding to the row located in the cell has in the snapshot or node corresponding to the column through the blue-to-red color scale.

The row header displays the type and numbering of TNMs, and users can check the shape of the motif by hovering a mouse pointer on them.

Users can choose between cluster order or time order for the x-axis and select which unit to aggregate the x-axis through two drop-down menus at the top: snapshot or node (Figure 4.D). When the x-axis is set to snapshot, users can view the motif distribution of each snapshot selected in the previous step. The color scale of each snapshot is identical to the color of the state. To provide a more informative visualization, a color scale is applied using the ratio of the normalized global average as a scale. Therefore, the motif distribution of each snapshot is compared with the motif distribution of the entire network as the baseline to obtain a ratio and adjust the opacity of the cell accordingly. Sorting in chronological order enables the identification of changes over time, while sorting in cluster order enables the identification of cluster characteristics and differences between clusters.

When a node is selected as the x-axis, users can obtain the distribution of motifs, including the node for the set of existing nodes in the selected snapshots. It acts as a motif degree vector (MDV) and provides rough information about the node's role. Based on the MDV, we cluster the nodes again to obtain the relationship between nodes and map different colors to each cluster. The color scale enables users to distinguish what they are currently searching for by using a different series than when they are in snapshot alignment.

If nodes are sorted according to their clusters, they are grouped and listed according to the cluster results. This enables users to understand how network nodes can be clustered and the characteristics of each cluster. If the alignment criterion is a snapshot, the nodes are aligned by the smallest timestamp they are included. In this case, users can observe the appearance of nodes and changes in their motif compositions.

Motif Composition View allows for the interactive selection of a specific snapshot or node, or their intersection, through a click interaction. This selection can provide valuable insights into these elements' roles in the network. Specifically, nodes and edges present in the selected range can be highlighted in the Network View on the right. This functionality enables users to explore the selected elements in greater detail and gain a deeper understanding of their contributions to the network's overall structure and behavior. Further details on this feature will be provided in the following section.

Scenario. We selected six snapshots, from the 29th to the 34th, to analyze structural patterns and temporal changes. For structural analysis, we sorted snapshots in cluster order and compared their motif compositions. States 3 and 4 shared similar features: high z-score in triangle, cycle, in-burst, and some ping-pong type TNMs, as the heatmap filter in Temporal Status View forecasted. State 2 was also similar but differed in in-burst type TNMs. State 1 was distinct from others, which had low scores in cycle and ping-pong types and high scores in out-burst TNMs. As we sorted the snapshots in temporal order, we can observe that throughout time, the states of snapshots changed from other states to state 1. From this observation result, we could assume that after the stock price peaked, some Enron employees started to send emails to other employees.

5.6 Network View

The Network View (Figure 4.E) represents the structure of selected snapshots as a node-link diagram. We adopt this node-link diagram since it is a well-known and traditional representation of the network structure. Thus, users can intuitively understand and navigate the network of their interest shown in this visualization. Nodes of selected snapshots are represented as circles, and the relationships between a pair of nodes are represented with a straight

3. piratepeel.github.io/code/enrontimeline.txt

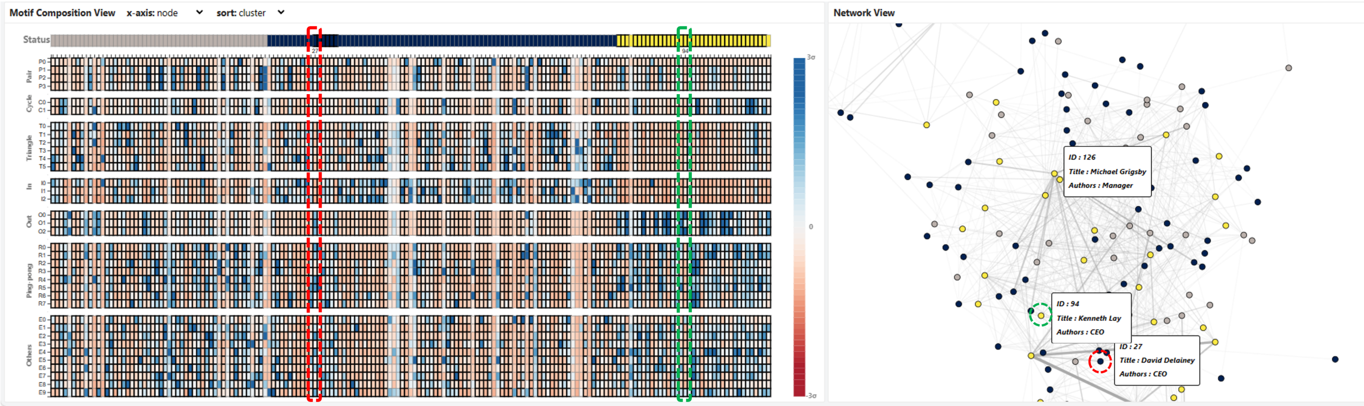

Fig. 6. Motif Composition View (left), whose domain of the x-axis is selected as the node and aligned in cluster order. The yellow cluster shows abundant out-burst, ping-pong, and other-type motifs compared to other clusters. One of the CEOs is included in this group of people: Kenneth Lay (green dotted lines). Some nodes in the light-grey cluster also show high scores in triangle and cycle motifs, one of whom is the other CEO, David Delainey. Network View (right) visualizes selected nodes. Two CEOs are neighboring a group of nodes with abundant ping-pong and cycle motifs (yellow cluster).

line. The higher the frequency of interaction between two nodes, the thicker the corresponding link rendered so that users can recognize the strength of the relationship between nodes. When the number of nodes is small, the label that can identify the node is shown together, and when the number increases, the label disappears, but detailed information can be found through mouse-hover interaction.

To enable users to inspect the detailed structure of temporal network motifs in snapshots, we reduced the scalability by only visualizing nodes and links that are contained in the selected motif types and snapshots in Motif Composition View. This feature enhances the usability of the Network View and provides a more detailed and flexible network structure analysis.

Scenario. From Motif Composition View, we selected the 32nd and 34th snapshots to analyze the detailed pattern of state 1.

As triangle and cycle motifs are abundant under state 1, we can infer that a large portion of interactions are made in a closed group of nodes (Figure 4.D). Out-burst motifs are also abundant, which implies the presence of someone who is continuously sending a stream of emails. Therefore, we can assume that the CEO was selected through extensive interactions within a closed group, and the CEO intervened in the decision process. The individual with the high out-burst motifs is likely to be one of the CEOs. Interestingly, when we set an x-axis of Motif Composition View as nodes and align it according to the clusters (Figure 6), we could figure out some nodes with high scores in cycle and triangle motifs, and their position in the given snapshots in Network View (Figure 6). Two CEOs, Kenneth Lay and David Delainey, and a group of people around them are actively interacting with each other.

6 Case Studies

We conducted case studies with two participants (P1, P2) who are experts in dynamic network analysis to evaluate MoNetExplorer in terms of efficacy in practice and identify further improvement opportunities. We recruited two researchers who have at least three years of dynamic network analysis experience and who don't have any contribution to designing or developing MoNetExplorer. The logic diagrams of their workflows in case studies can be found in the appendix.

6.1 Experiment Setting

For the experiment, participants used MoNetExplorer on a 27-inch FHD monitor, equipped with a mouse as their input device. After a five-minute introduction about our research goal and a 10-minute tutorial of MoNetExplorer, a participant spent 30 minutes on their desktop playing with the system. Participants' goal was to carry out tasks: finding temporal patterns, selecting the best-fit window size for the dataset, analyzing node attributes, and studying patterns with temporal network motifs. We requested participants to share their discoveries and explain how they arrived at their conclusions. Throughout each session, we assisted participants, answering any questions they had. The experiment was fully transcribed.

For the evaluation of the system, we selected four heuristic evaluation criteria: Insight, Time, Essence and Confidence proposed by Wall et al. [57]. Although these metrics can be employed to evaluate the quality of visualizations qualitatively, the limited sample size resulting from our case study necessitates a different approach. Consequently, we gathered qualitative feedback for each category by conducting interviews. A brief description of the criteria is shown in Table 2.

6.2 Case Study: Enron Email Network

We conducted a case study with the Enron email dataset, which is also used to demonstrate the visualization design. We asked P1 to perform the aforementioned tasks with MoNetExplorer.

To find the best-fit window size for the Enron dataset (T1), P1 focused on the overall score of three measures in the Slicing Navigation View. P1 said that if the suggested three measures are reasonable, the most reasonable way is to choose the best-fit window according to the overall score. It made P1 less interested in other candidates who

TABLE 2
The definitions and brief descriptions of four evaluation criteria.

Criteria	Description
Insight	How a visualization supports intentional and incidental insights
Time	How a visualization facilitates faster, more efficient understanding of data concerning both searching and browsing of data.
Essence	How a visualization communicates the essence of the data set with respect to overview and context.
Confidence	How visualization helps a user feel confident in understanding the data set with respect to the data quality and the visualization quality.

have lower ranks. "If the first window size at the top shows the best performance in all three measures, then there is no reason not to select it", P1 said. Therefore, without any trials for other window sizes, P1 selected 29 days, which is the top-rank window size. However, P1 expressed confidence in **Motif Fidelity** and **Motif Stability** but expressed some doubts about **Motif Clusterness**. "I still don't fully understand why a cluster validation measure can be used to estimate the quality of dynamic network slicing."

After selecting the window size of 29 days, P1 performed a task to figure out relations between snapshots and detect temporal patterns (T2). P1 mainly focused on the Temporal Measure View, showing the temporal changes of **Motif Fidelity** and **Motif Stability** at a glance. P1 figured out state change points by finding snapshot indexes where drastic changes in measures occurred. Between the 33rd and the 36th snapshots, both **Motif Fidelity** and **Motif Stability** drop drastically. Note that low **Motif Stability** guarantees a change of snapshot state in general; it is a valid method for finding change points. So, P1 checked whether the state had changed with Temporal Status View and found a state transition in the 35th snapshot. P1 said, "When I look for snapshots where the fidelity and stability change abruptly, I can also see the state transitions".

With Motif Composition View, P1 tried to figure out motif compositions of the discovered temporal patterns (T3). P1 brought three snapshots, 34th, 35th, and 36th, located around the detected change point. First of all, P1 asked us about the criteria for dividing 36 motifs into seven types. Then, P1 detected severe differences in the triangle and cycle-type motifs of the 35th snapshot from others. P1 also reported that Motif Composition View not only offers a chance to detect changes in pattern over time but also analyze it in more detail. From this observation, P1 assumed that at the time period of the 35th snapshot, some huge events had happened, and employees exchanged emails a lot. We confirmed P1 that the time period was when the Enron stock hit an all-time high price(August 23, 2000).

At last, P1 tried to find CEOs of the Enron corporation from selected snapshots and discover the dominant motif with Network View (T4). However, due to the massive amount of nodes visualized in Network View, P1 suffered from scalability issues, unable to find CEOs from the data. Node clustering with Motif Composition View was not effective because motif compositions of CEOs are not distinguished from others. Also, P1 pointed out that the lack of motif composition visualization in Network View made it difficult to perform T4 solely with Network View.

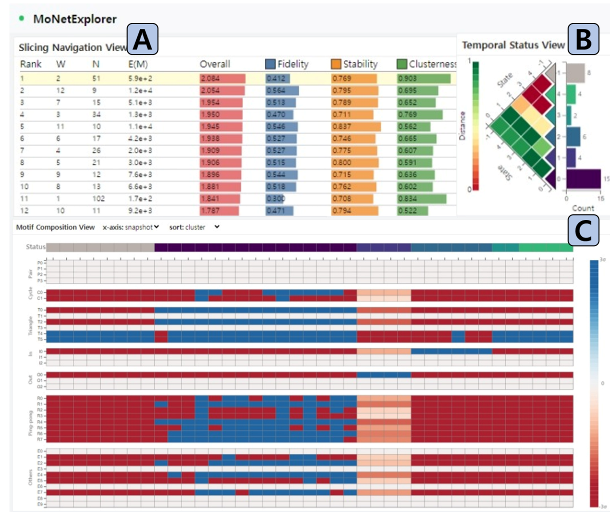

Fig. 7. Workflow of P2 validating the window size of 2 months. (A) Select the window size. (B) Observe detected snapshot states and their distances. Seven different states are detected. (C) Observe motif compositions of snapshot states. In this case, some states are not well-clustered.

In general, P1 employed the system in a manner consistent with our anticipated usage scenario in Section 5. Such results verify the validity of our user scenario as well as the effectiveness of our system design. While performing T1 and T2, P1 was able to get an insight into the best-fit window size to create snapshots and relations between them. Also, time performance in T1 and T2 was good, as P1 made decisions in a short time. P1's trust in snapshot validation measures not only supported P1's rapid decision-making but also gave him high confidence about the decision. However, there exists a little misgiving in MC. As a result, P1 was able to get the essence of the overview of the dataset throughout the workflow but had a hard time achieving the essence of detail.

6.3 Case Study: Citation Network

For P2, we conducted a case study with the high-energy physics citation dataset (HEP-PH) [3]. The data includes 34,546 papers uploaded to the HEP-PH section of arXiv and 421,578 citations between them from January 1993 to April 2003 (124 months). We decided a month as a base time unit and provided twelve candidate window sizes: one month to 12 months.

To find the best-fit window size for the data, P2 first observed Slicing Navigation View (T1). P2 mainly focused on the overall score of all three measures but did not rely entirely on it. "I don't rely on metrics when making decisions because there might be some distortions or loss of information. I make decisions after observing the data inside", P2 said. P2 utilized Slicing Navigation View as a recommendation list and excluded window sizes with low scores. From the top of the list, P2 observed whether temporal patterns are well distinguished in each window size. Then, P2 selected every snapshot with the state filter

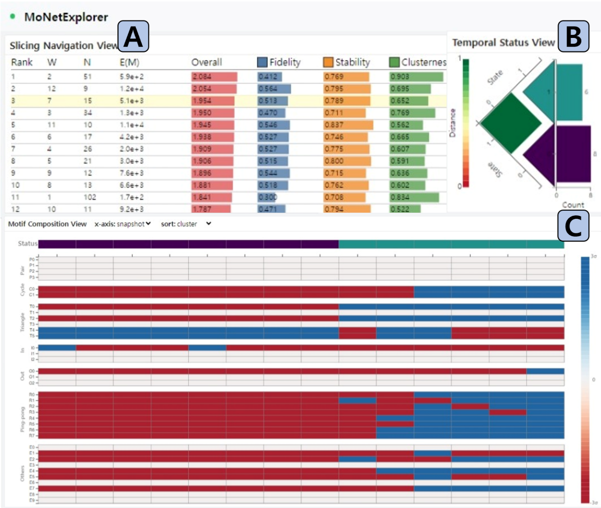

Fig. 8. Workflow of P2 validating the window size of 7 months. (A) Select the window size. (B) Observe detected snapshot states and their distances. Two different snapshot states are detected. (C) Observe motif compositions of snapshot states. In this case, the two states are well-distinguishable with their difference in ping-pong and triangle-type motifs.

of Temporal Status View and compared their motif with Motif Composition View. P2 believes that a well-executed slicing should clearly distinguish each state. Window size of 2 months, which is the top rank in overall score (Figure 7.A), was not satisfactory for P2 because some of its network states are not well clustered (Figure 7.C). However, a window size of 7 months, despite its mediocre rank (Figure 8.A), had better clustering results (Figure 8.C). Therefore, P2 concluded a window size of 7 months was the best-fit window size.

P2 mainly observed relations between snapshots with Temporal Status View (T2). P2 first checked how many states had been detected (Figure 8.B) and observed how the states of snapshots changed over time. With a window size of 7, P2 found that state one had been detected in the beginning, while state 2 was dominant in the end. However, unlike P1, P2 did not take the temporal changes in measures as an indicator of temporal patterns and did not gain much insight from Temporal Measure View. Instead, P2 utilized the bar chart at the left as an indicator of overall quality.

With Motif Composition View, P2 tried to analyze the detected inner structural patterns of states (T3). P2 first sorted selected snapshots in cluster order. Then, P2 analyzed the motif compositions of each state and compared them with others (Figure 8.C). P2 said, ”When referring to the description and image of the motif’s structure, it was possible to get a rough idea of what patterns the snapshot generally has.” P2 also said, ”The difference in the composition of motifs was clearly seen, so it was possible to grasp the difference between each state.” When we asked P2 to describe the structure of state 2, which has a high frequency of ping-pong motifs, P2 assumed there might be controversy on specific research topics that papers referring to each other have been more frequent than before.

We asked P2 to figure out the paper which seems to be most important in the data and describe its network structure (T4). Considering the type of data, the most critical node was decided as the most cited paper. P2 first tried to find relations between nodes with Network View and figured out the nodes with a high degree. Then, P2 checked Motif Composition View to determine the selected nodes’ motif composition patterns. With this process, P2 was able to find out the paper with a high degree and its relations with other papers. However, P2 said that Network View alone was not that helpful to perform the task. Cooperation with Motif Composition View was necessary. Although a node-link diagram is an effective visualization for analyzing network structure, it also has scalability issues when visualizing multi-variate network data. Augmentation of additional information, such as the degree of node or direction, should be added to make Network View effective for deriving insights.

In contrast to P1, P2 checked from the detailed motif composition and validated that the slicing was actually done properly while performing T1. For example, unlike our usage scenario, P2 utilized Motif Composition View from the very beginning of the workflow. Through this exploration, P2 gained deep insight into selecting proper window sizes and their inner structural patterns and high confidence about his decision. However, time consumption became much higher than P1. P2 also reported a prominent high-level essence of the dataset but a shortfall when it came to the details. The abundance of motifs in the dataset, which offers a general overview, works as an obstacle in examining detailed information Network View.

7 Discussion

7.1 Temporal Network Motifs

Our study has demonstrated the potential of utilizing both the structural and temporal characteristics of a network for classification purposes. Incorporating a larger variety of TNMs may enhance the performance of visual analytics systems. Although our study focused on 2-3 node, 3-edge, and Δw motifs, motifs with a larger number of nodes and edges can better convey the structural and temporal characteristics of the original network. Furthermore, selecting motifs based on both their duration and the spacing between their edges can further enhance their expressiveness [44].

However, increasing the variety of motifs can also lead to the curse of dimensionality, making it more challenging to compute the distances between motif compositions. Hence, further research is needed to extract meaningful subgroups of TNMs and explore data transformation and distance comparison methods that consider these improvements. To overcome these limitations, it is necessary to complement TNMs with other methods, such as node degree or centrality measures.

7.2 Dynamic Network Slicing and Workflow

Numerous studies have emphasized the importance of observing dynamic networks with various resolutions instead of relying on a single optimal window size. In line with this view, our study also demonstrated, through a usage scenario, that the pattern that can be observed varies depending on the window size. It also proved the necessity

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. 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/TVCG.2023.3337396

of trying diverse window sizes to find patterns from a dynamic network, and our proposed methods could help users effectively perform the process. Recommendations with visualization helped users easily find proper candidates and validate them.

However, such workflow has a limitation in that the exploration space is limited to fixed-sized windows whose sizes are regularized with base time units. In our case study, both participants answered that they could figure out the best window size from the candidates, but the best window size did not fit every snapshot. Also, P2 tried to validate the chosen window size and select another one, which was time-consuming. If we try to expand the options in window sizes and consider positions of window size, it will be unable to manage it with static, fixed-size windows. Therefore, in future work, we will research non-uniform slicing algorithms using TNMs as the basis. It will meet the requirement to consider various window sizes and positions while further reducing human cognitive load because it has the same effect of navigating with multiple customized windows simultaneously.

7.3 Visual Analytics System Evaluation

We analyzed the case study results according to four heuristic aspects: insight, time, essence, and confidence. For insight, both P1 and P2 were able to find insights into the optimal window size for their respective methods (T1). They also gained insights into the relationships between the snapshots of the network (T2) and the patterns that predominantly appeared (T3). However, they struggled to gain insights into the composition of the internal graph attributes (T4). This phenomenon was similar to essence; while participants found the Slicing Navigation View helpful for grasping an overview of the data, they reported difficulty in understanding details as they got closer to them.

Regarding time, P1 made quick decisions using the system, but P2 did not. It was because, unlike P1, who trusted the snapshot validation measures, P2 went through the process of verifying the results directly, even if the measure scores were high. Both eventually had confidence in their choices, but P1's process of gaining confidence was shorter, suggesting a higher level of confidence compared to P2. These results emphasize the flexibility of our system in exploring dynamic networks. Investigating how different users react to MoNetExplorer will be an interesting future avenue to explore.

In conclusion, for insight and essence, high scores could be given at the high level, but scores decreased as they approached the low level. For time and confidence, P1, who trusted the measures, gained high scores, while P2, who relied more on direct observation of results, gained lower scores. It might be necessary to design measures more intuitively and improve the close connection between overview and detail in the visualization system to accommodate users with tendencies like P2. Additionally, improvements in visualizations, such as showing the TNM kernel in Network View, are needed for better support of low-level tasks (T4).

7.4 Snapshot Quality Evaluation

Our study builds upon prior research on window sizing in dynamic networks and proposes three distinct measures to evaluate the quality of network snapshots. These measures offer a comprehensive approach to assessing snapshot quality and can guide the selection of appropriate window sizes. In the case studies, unlike our usage scenario, no attempt was made to sort Slicing Navigation View with the overall score of partial or other measures. Participants judged that the theoretical basis for the three measures was plausible, so it was most desirable to rank them by reflecting all three It confirmed that users could effectively use our proposed measures to evaluate slicing. However, there is a further need for a complex measure that reflects all the various elements of the dynamic network.

Nevertheless, the question remains: What makes a snapshot truly ”good”? To answer this question, many studies, including ours, have presented measures for evaluating snapshot quality and validating the efficacy of good snapshots. However, sometimes measures fail to evaluate the quality properly. In our usage scenario, for instance, the 29-day window size recorded the best score in the sum of **Motif Fidelity**, **Motif Stability**, and **Motif Clusterness**, but other window sizes provided better user experience. In fact, in the case study, P2 did not fully rely on the measures to evaluate the snapshot quality. Instead, P2 decided by observing the details of the resulting snapshots. Regardless of the slicing method, snapshots should be able to provide insights to human users.

Considering human visual cognition, some researchers tried to minimize the visual complexity of snapshots [20], [21]. However, these studies need more consideration for delivering structural and temporal evolution information. Therefore, it is crucial to find a balance between information delivery and the visual complexity of snapshots. In a follow-up study, we will search for the balance by conducting a user study to build a user-based baseline for evaluating the quality of snapshots.

8 Conclusion

We present MoNetExplorer, an interactive visual analytics system that leverages TNMs to facilitate the selection and validation of window sizes for slicing dynamic networks. Our system allows users to explore multiple candidate window sizes and identify changes in the network structure over time. By presenting usage scenarios and case studies with a real-world dynamic network dataset, we showed the effectiveness of MoNetExplorer in selecting proper window sizes and detecting changing patterns in the temporal occurrence of actual historical events. Our contributions include (1) the development of TNM-based measures and algorithms for selecting and validating window sizes and (2) the design and implementation of MoNetExplorer as a comprehensive visual analytics system for dynamic network analysis. We believe that our system is useful for researchers and practitioners who want to understand the evolving relationships between entities in dynamic networks.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2023R1A2C200520911), by the Supreme Prosecutors' Office of the Republic of Korea grant funded by the Ministry of Science and ICT (SPO2023A1202digitalB), and by the Research Grant of Kwangwoon University in 2023. The approval for the user studies was granted by Seoul National University IRB (IRB No.2310/001-001). Furthermore, we are grateful to the reviewers of our submission to VIS 2023 and TVCG for their valuable comments and suggestions, which helped us improve our article.

REFERENCES

[1] A. Lee, D. Archambault, and M. Nacenta, "Dynamic network plaid: A tool for the analysis of dynamic networks," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19, no. Paper 130. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–14.

[2] P. Moriano, J. Finke, and Y.-Y. Ahn, "Community-Based event detection in temporal networks," *Sci. Rep.*, vol. 9, no. 1, p. 4358, Mar. 2019.

[3] J. Gehrke, P. Ginsparg, and J. Kleinberg, "Overview of the 2003 kdd cup," *Acm Sigkdd Explorations Newsletter*, vol. 5, no. 2, pp. 149–151, 2003.

[4] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park, "Scan statistics on enron graphs," *Computational & Mathematical Organization Theory*, vol. 11, pp. 229–247, 2005.

[5] G. Krings, M. Karsai, S. Bernhardsson, V. D. Blondel, and J. Saramäki, "Effects of time window size and placement on the structure of an aggregated communication network," *EPJ Data Science*, vol. 1, no. 1, pp. 1–16, May 2012.

[6] S. Uddin, N. Choudhury, S. M. Farhad, and M. T. Rahman, "The optimal window size for analysing longitudinal networks," *Sci. Rep.*, vol. 7, no. 1, p. 13389, Oct. 2017.

[7] A. Chiappori and R. Cazabet, "Quantitative evaluation of snapshot graphs for the analysis of temporal networks," in *Complex Networks & Their Applications X*. Springer International Publishing, 2022, pp. 566–577.

[8] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, ser. WSDM '17. New York, NY, USA: Association for Computing Machinery, Feb. 2017, pp. 601–610.

[9] P. Liu, N. Masuda, T. Kito, and A. E. Sarıyüce, "Temporal motifs in patent opposition and collaboration networks," *Sci. Rep.*, vol. 12, no. 1, p. 1917, Feb. 2022.

[10] D. Lei, X. Chen, L. Cheng, L. Zhang, S. V. Ukkusuri, and F. Witlox, "Inferring temporal motifs for travel pattern analysis using large scale smart card data," *Transp. Res. Part C: Emerg. Technol.*, vol. 120, p. 102810, Nov. 2020.

[11] T. A. Schieber, L. Carpi, A. Díaz-Guilera, P. M. Pardalos, C. Masoller, and M. G. Ravetti, "Quantification of network structural dissimilarities," *Nat. Commun.*, vol. 8, p. 13928, Jan. 2017.

[12] R. Sulo, T. Berger-Wolf, and R. Grossman, "Meaningful selection of temporal resolution for dynamic networks," in *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, ser. MLG '10. New York, NY, USA: Association for Computing Machinery, Jul. 2010, pp. 127–136.

[13] H. Chen, U. Soni, Y. Lu, R. Maciejewski, and S. Kobourov, "Same stats, different graphs: (graph statistics and why we need graph drawings)," in *Graph Drawing and Network Visualization: 26th International Symposium, GD 2018, Barcelona, Spain, September 26-28, 2018, Proceedings 26*. Springer, 2018, pp. 463–477.

[14] E. Cakmak, J. Fuchs, D. Jäckle, T. Schreck, U. Brandes, and D. Keim, "Motif-Based visual analysis of dynamic networks," in *2022 IEEE Visualization in Data Science (VDS)*, Oct. 2022, pp. 17–26.

[15] J. Troidl, S. Warchol, J. Choi, J. Matelsky, N. Dhanyasi, X. Wang, B. Wester, D. Wei, J. W. Lichtman, H. Pfister *et al.*, "Vimo: Visual analysis of neuronal connectivity motifs," *bioRxiv*, pp. 2022–12, 2022.

[16] B. Bach, E. Pietriga, and J.-D. Fekete, "Visualizing dynamic networks with matrix cubes," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: Association for Computing Machinery, Apr. 2014, pp. 877–886.

[17] J. Moody, D. McFarland, and S. Bender-deMoll, "Dynamic network visualization," *American journal of sociology*, vol. 110, no. 4, pp. 1206–1241, 2005.

[18] G. K. Orman, N. Türe, S. Balcisoy, and H. A. Boz, "Finding proper time intervals for dynamic network extraction," *J. Stat. Mech: Theory Exp.*, vol. 2021, no. 3, pp. 0–0, Mar. 2021.

[19] A. Arleo, S. Miksch, and D. Archambault, "Event-based dynamic graph drawing without the agonizing pain," *Comput. Graph. Forum*, vol. 41, no. 6, pp. 226–244, Sep. 2022.

[20] Y. Wang, D. Archambault, H. Haleem, T. Moeller, Y. Wu, and H. Qu, "Nonuniform timeslicing of dynamic graphs based on visual complexity," in *2019 IEEE Visualization Conference (VIS)*, Oct. 2019, pp. 1–5.

[21] J. R. Ponciano, C. D. G. Linhares, E. R. Faria, and B. A. N. Travençolo, "An online and nonuniform timeslicing method for network visualisation," *Comput. Graph.*, vol. 97, pp. 170–182, Jun. 2021.

[22] M. Latapy, T. Viard, and C. Magnien, "Stream graphs and link streams for the modeling of interactions over time," *Soc. Netw. Anal. Min.*, vol. 8, no. 1, Dec. 2018.

[23] N. Masuda and P. Holme, "Detecting sequences of system states in temporal networks," *Sci. Rep.*, vol. 9, no. 1, p. 795, Jan. 2019.

[24] F. Beck, M. Burch, S. Diehl, and D. Weiskopf, "A taxonomy and survey of dynamic graph visualization," *Comput. Graph. Forum*, vol. 36, no. 1, pp. 133–159, Jan. 2017.

[25] T. Crnovrsanin, Shilpika, S. Chandrasegaran, and K.-L. Ma, "Staged animation strategies for online dynamic networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 539–549, Feb. 2021.

[26] B. Tversky, J. B. Morrison, and M. Betrancourt, "Animation: can it facilitate?" *Int. J. Hum. Comput. Stud.*, vol. 57, no. 4, pp. 247–262, Oct. 2002.

[27] I. Boyandin, E. Bertini, and D. Lalanne, "A qualitative study on the exploration of temporal changes in flow maps with animation and small-multiples," *Comput. Graph. Forum*, vol. 31, no. 3pt2, pp. 1005–1014, Jun. 2012.

[28] T. Munzner, *Visualization analysis and design*. CRC press, 2014.

[29] B. Bach, N. Henry-Riche, T. Dwyer, and others, "Small MultiPiles: Piling time to explore temporal patterns in dynamic networks," *Comput. Graph.*, 2015.

[30] A. Lee, D. Archambault, and M. A. Nacenta, "The effectiveness of interactive visualization techniques for time navigation of dynamic graphs on large displays," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 528–538, Feb. 2021.

[31] 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 *2014 IEEE Pacific Visualization Symposium*. ieeexplore.ieee.org, Mar. 2014, pp. 121–128.

[32] E. Cakmak, U. Schlegel, D. Jackle, D. Keim, and T. Schreck, "Multiscale snapshots: Visual analysis of temporal summaries in dynamic graphs," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 517–527, Feb. 2021.

[33] C. D. G. Linhares, J. R. Ponciano, D. S. Pedro, L. E. C. Rocha, A. J. M. Traina, and J. Poco, "LargeNetVis: Visual exploration of large temporal networks based on community taxonomies," *IEEE Trans. Vis. Comput. Graph.*, vol. PP, Sep. 2022.

[34] T. N. Dang, N. Pendar, and A. G. Forbes, "TimeArcs: Visualizing fluctuations in dynamic networks," *Comput. Graph. Forum*, vol. 35, no. 3, pp. 61–69, Jun. 2016.

[35] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic, "Time curves: Folding time to visualize patterns of temporal evolution in data," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 559–568, Jan. 2016.

[36] 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*, ser. AVI '20, no. Article 26. New York, NY, USA: Association for Computing Machinery, Sep. 2020, pp. 1–9.

[37] L. Kovanen, M. Karsai, K. Kaski, J. Kertész, and J. Saramäki, "Temporal motifs in time-dependent networks," *J. Stat. Mech.*, vol. 2011, no. 11, p. P11005, Nov. 2011.

[38] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, Oct. 2002.

[39] L. Kovanen, K. Kaski, J. Kertész, and J. Saramäki, "Temporal motifs reveal homophily, gender-specific patterns, and group talk in call sequences," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 110, no. 45, pp. 18 070–18 075, Nov. 2013.

[40] D. Jurgens and T.-C. Lu, "Temporal motifs reveal the dynamics of editor interactions in wikipedia," *ICWSM*, vol. 6, no. 1, pp. 162–169, 2012.

[41] Q. Xuan, H. Fang, C. Fu, and V. Filkov, "Temporal motifs reveal collaboration patterns in online task-oriented networks," *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, vol. 91, no. 5, p. 052813, May 2015.

[42] Y. Hulovatyy, H. Chen, and T. Milenković, "Exploring the structure and function of temporal networks with dynamic graphlets," *Bioinformatics*, vol. 31, no. 12, pp. i171–80, Jun. 2015.

[43] J. Crawford and T. Milenković, "ClueNet: Clustering a temporal network based on topological similarity rather than denseness," *PLoS One*, vol. 13, no. 5, p. e0195993, May 2018.

[44] P. Liu, V. Guarrasi, and A. E. Sariyuce, "Temporal network motifs: Models, limitations, evaluation," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2021.

[45] N. Kerracher, J. Kennedy, and K. Chalmers, "A task taxonomy for temporal graph visualisation," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 10, pp. 1160–1172, Oct. 2015.

[46] Z. Gao, C. Cheng, Y. Yu, L. Cao, C. Huang, and J. Dong, "Scalable motif counting for large-scale temporal graphs," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. ieeexplore.ieee.org, May 2022, pp. 2656–2668.

[47] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering." *J. Open Source Softw.*, vol. 2, no. 11, p. 205, 2017.

[48] H. Jeon, H.-K. Ko, J. Jo, Y. Kim, and J. Seo, "Measuring and explaining the inter-cluster reliability of multidimensional projections," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 551–561, 2022.

[49] M. Halkidi and M. Vazirgiannis, "Clustering validity assessment: finding the optimal partitioning of a data set," in *Proceedings 2001 IEEE International Conference on Data Mining*. ieeexplore.ieee.org, Nov. 2001, pp. 187–194.

[50] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE International Conference on Data Mining*. ieeexplore.ieee.org, Dec. 2010, pp. 911–916.

[51] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE symposium on visual languages*. IEEE, 1996, pp. 336–343.

[52] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004. Proceedings 15*. Springer, 2004, pp. 217–226.

[53] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, "Lineup: Visual analysis of multi-attribute rankings," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 2277–2286, 2013.

[54] K. Furmanova, S. Gratzl, H. Stitz, T. Zichner, M. Jaresova, A. Lex, and M. Streit, "Taggle: Combining overview and details in tabular data visualizations," *Information Visualization*, vol. 19, no. 2, pp. 114–136, 2020.

[55] L. Peel and A. Clauset, "Detecting change points in the large-scale structure of evolving networks," *Proc. Conf. AAAI Artif. Intell.*, vol. 29, no. 1, Feb. 2015.

[56] E. Cakmak, D. Jackle, T. Schreck, and D. Keim, "Dg2pix: Pixel-based visual analysis of dynamic graphs," in *2020 Visualization in Data Science (VDS)*. IEEE, Oct. 2020, pp. 32–41.

[57] E. Wall, M. Agnihotri, L. Matzen, K. Divis, M. Haass, A. Endert, and J. Stasko, "A heuristic approach to value-driven evaluation of visualizations," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 491–500, 2018.