IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 10, 2018, accepted September 7, 2018, date of publication September 17, 2018, date of current version October 12, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2870684

EOD Edge Sampling for Visualizing Dynamic Network via Massive Sequence View

YING ZHAO[1], YANMIN SHE[1], WENJIANG CHEN[1], YUTIAN LU[1], JIAZHI XIA[1], WEI CHEN[2], JUNRONG LIU[3], AND FANGFANG ZHOU[1]

[1]School of Information Science and Engineering, Central South University, Changsha 410083, China
[2]State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China
[3]Key Laboratory of Network Assessment Technology, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Corresponding author: Jiazhi Xia (xiajiazhi@csu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0904503, in part by the National Science Foundation of China under Grant 61672538, Grant 61772456, Grant 61872388, and Grant 61872389, in part by the Natural Science Foundation of Hunan Province under Grant 2017JJ3414, in part by the Open Project Program of the State Key Lab of CAD&CG, Zhejiang University, under Grant A1812, and in part by the Open Research Fund of the Key Laboratory of Network Assessment Technology, Institute of Information Engineering, Chinese Academy of Sciences.

ABSTRACT Dynamic network visualization is crucial to understand network evolving behavior. Massive sequence view (MSV) is a classic technique for visualizing dynamic networks and provides users with a fine-grained presentation of time-varying communication trend from both node pair and global network levels. However, MSV is vulnerable to visual clutter caused by overlapping edges, failing to show clear patterns or trends. This paper presents an edge sampling method, using the edge overlapping degree (EOD) concept, to reduce visual clutter in MSV while preserving the time-varying features of network communication. Referring to accept–reject sampling, we use kernel density estimation to characterize the time-varying features between node pairs and generate EOD probability density functions to accomplish sampling in a bottom-up manner. To enhance the sampling effect, we also consider the edge length factor and streaming processing. The case studies on two dynamic network data sets demonstrate that our method can significantly improve the overall readability of MSV and clearly reveal the temporal features of both node pairs and global network. A quantitative evaluation comparing with two other sampling methods using three real-world data sets indicates that our method can well balance visual clutter reduction and temporal feature preservation.

INDEX TERMS Dynamic network visualization, massive sequence view, graph sampling, visual abstraction.

I. INTRODUCTION

Networks (or graphs) are often formed gradually and evolve continuously in real-world domains, such as phone calls, email communications, and Twitter posts. Effective analysis of these dynamic networks is crucial to understand time-varying network behavior. Dynamic network visualization [1] is an active research field that provides intuitive diagrams and rich interactions to involve users in making sense of networks' evolving nature. Massive sequence view (MSV) is a classic dynamic network visualization technique [2], widely used for analyzing dynamic social network [3] and program execution traces [4]. Fig.1(a) and Fig.1(b) demonstrate its visual encodings. Nodes (or vertices) of a dynamic network are represented by horizontal lines, which are equally spaced along the vertical axis. The horizontal axis represents the time when the network exists. If there is an instant relation (edge) from nodes a to b at time t_i, then a vertical line with start and end points at the vertical positions of a and b, respectively, is drawn at horizontal position t_i. This step is repeated for all edges in the observation period.

Two advantages make MSV easy for users to observe the communication trend from the perspectives of node pair and global network. First, MSV enables arbitrarily fine-grained visualization as edges are drawn onto a continuous timeline. Second, MSV can preserve users' mental maps because of its fixed node positions. However, MSV is vulnerable to visual clutter. In a dynamic network, multiple edges may occur at (approximately) the same time, leading to overlapping edges or failing to show distinguishable edges due to insufficient horizontal pixels, as shown in Fig.1(b). MSV's overall readability is thus compromised. Furthermore, users may misunderstand the time-varying trend of network communication. For example, the MSV in Fig.2(a) presents the dynamic network of internal emails of Enron, a former energy

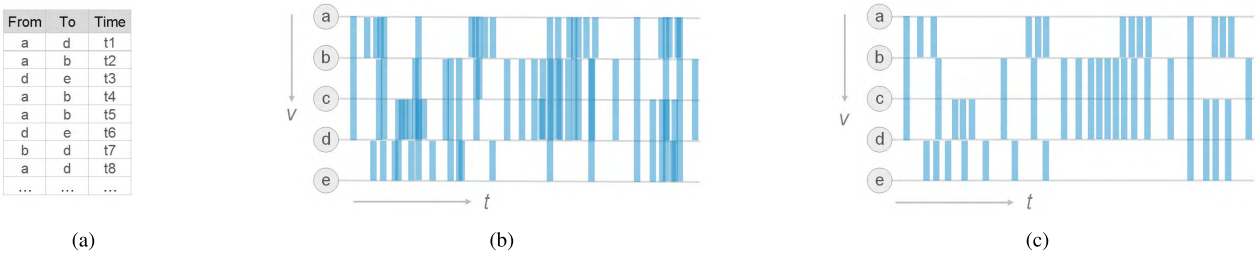


FIGURE 1. Use of MSV to visualize the dynamic network in table (a) before sampling and (b) after sampling (c). The overall readability of MSV is improved while the time-varying features of network communication is preserved after sampling.

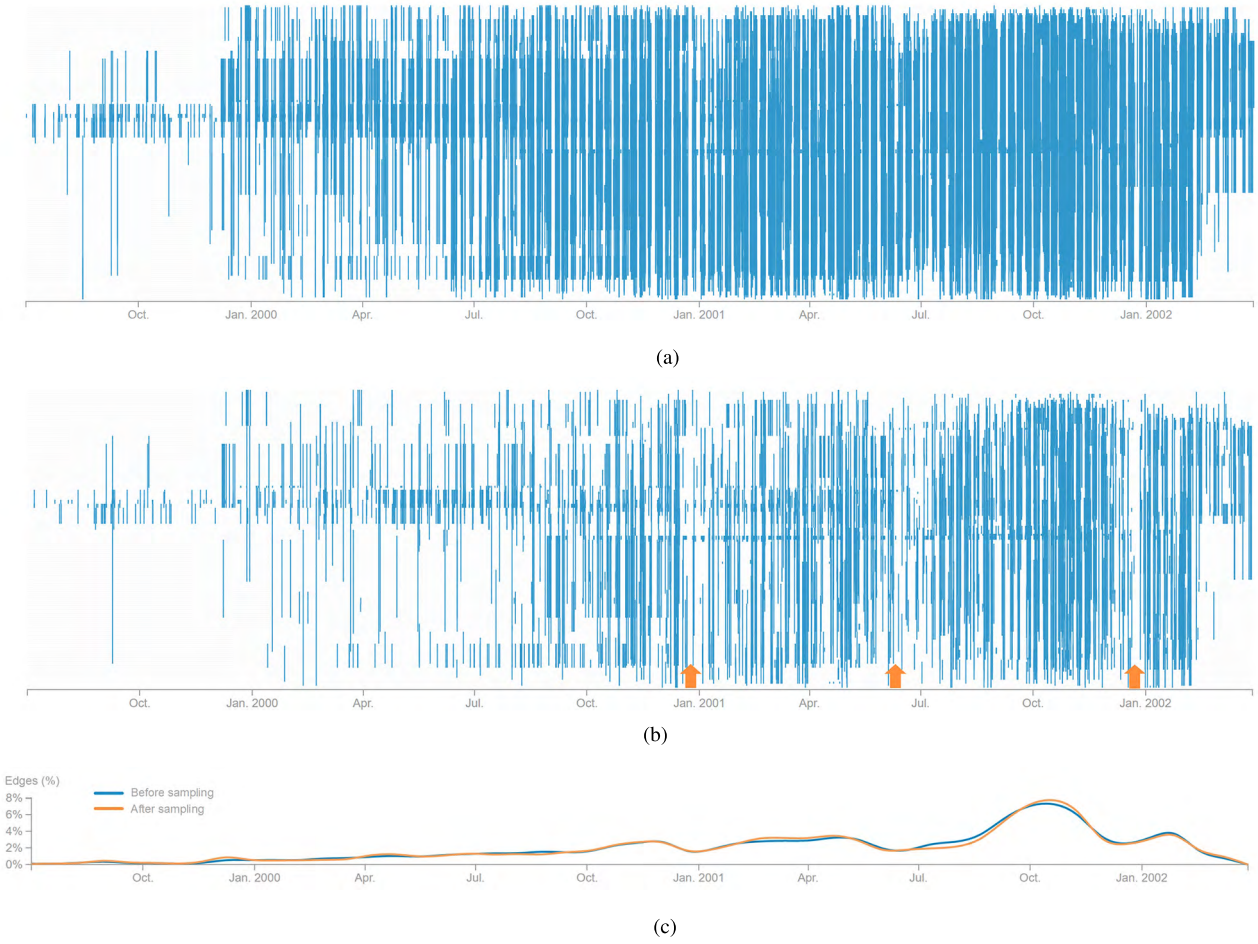


FIGURE 2. Enron email dataset containing 24,705 email communications (edges) between 150 employees (nodes) from 1999 to 2002. The nodes in the two MSVs are sorted by the node reordering strategy of minimizing edge length [3]. There is serious visual clutter in MSV (a). After sampling (b), the overall readability of MSV is improved, clearly showing three sudden drops of email communications (three orange arrows), indicating the events of two CEO replacements and bankruptcy petition of Enron company. Two distribution curves in (c) stick together most of the time, thereby indicating that the time-varying trend of email traffic is well persevered after sampling.

service company. This dynamic network contains a series of evolving events related to the biggest American accounting scandal [5]. With serious visual clutter, the MSV cannot depict the sudden changes in email traffic caused by the events. Users may even misunderstand that the email communication remains intense and frequent through the latter half of the timeline (see Section 5.1 for more detailed case analysis).

Several techniques have been proposed to improve standard MSV. Filtering and zooming [6], [7] provide detail-only and pixel-increased views to show selected periods and nodes, but a clear overview is lost. Anti-aliasing techniques [8] can reduce the visual clutter caused by overplotting edges but not by overlapping edges. Curved links [9] avoid edge overlapping but result in edge crossings. Node ordering strategies are by far the best techniques to

improve MSV [3], [10], but achieving satisfactory optimization results is particularly difficult when communications between mass nodes occur densely and irregularly. Improving the overall readability of MSV remains a challenging problem.

The main cause of visual clutter in MSV is that too many edges are drawn on a screen with a limited size. Empirically, not all edges in a dynamic network are significantly related to network evolution, and some may be regarded as background noise. Removal of noise edges can reduce the amount of edges to be displayed on the screen, as shown in Fig.1(c). This idea coincides with that of graph sampling technique for generating highly readable graph visualization on complex network. However, existing graph sampling methods [11]–[13] largely concern static network analysis. To our knowledge, no method has been designed for dynamic network analysis.

To facilitate dynamic network analysis with MSV, this paper proposes an edge overlapping degree (EOD) concept for edge sampling to reduce visual clutter in MSV while preserving the time-varying features of a network. EOD is designed to quantitatively measure the degree of an edge in MSV overlapping with its neighboring edges, i.e., the amount of visual clutter in the corresponding area of the edge. As an edge level indicator of visual clutter, EOD facilitates fine-grained sampling examination and supports other edge operations, such as interactive local exploration. To preserve the time-varying features of network communication in clutter-reduced MSV, we first use kernel density estimation (KDE) to generate probability density functions (PDFs) for characterizing such features between node pairs. Then, referring to accept-reject (AR) sampling, a type of Monte Carlo method known for the capability of sampling arbitrary target PDFs if given suitable proposal PDFs, we generate EOD-based proposal PDFs to achieve our expected sampling to balance visual clutter reduction with feature preservation in a bottom-up manner. To further enhance the sampling effect, we consider the influences of edge lengths and the discreteness of edges on visual clutter reduction and temporal feature representation.

To evaluate our method, we apply it to two well-known real-world dynamic network datasets. The two case studies demonstrate that the overall readability of MSV can be effectively improved after applying our sampling method. Consequently, the time-varying features of both node pair and global network levels presented in sampled MSV are clear and accurate, revealing many visual patterns hidden in the original MSV. We also perform an evaluation with three quantitative indicators by comparing our method with AR sampling and random sampling. The first indicator, KS distance, measures the sampling performance on the time-varying feature preservation of a dynamic network. The other two indicators, edge overlapped rate and edge hidden rate, measure the ability of reducing MSV's visual clutter. Results show that our method achieves good performance to feature preservation, and generally outperforms AR sampling and random sampling with respect to visual clutter reduction. The proposed sampling method can be considered a novel attempt to improve MSV and the first step to extend graph sampling to dynamic network analysis.

The main contributions of this paper are summarized as follows:

- We propose a quantitative indicator called EOD for measuring edge-level visual clutter in MSV;
- We propose a flexible edge sampling method that can achieve the trade-off between visual clutter reduction in MSV and time-varying feature preservation of a dynamic network.

The remaining parts are organized as follows. Section 2 summarizes related work. Sampling considerations are described in Section 3. Section 4 first details design challenges and then presents EOD edge sampling. Two case studies are showed in Section 5, followed by a quantitative evaluation. Parameters, limitations and future work are discussed in Section 6. Finally, we conclude this paper in Section 7.

II. RELATED WORK

A. DYNAMIC NETWORK VISUALIZATION

Dynamic network visualization is an active research field in visualization and visual analytics community. A recent survey [1] provides a systematic review of the growing number of dynamic network visualization techniques by classifying the existing techniques into two main categories: animation-based and timeline-based. Animation-based techniques typically use animated node-link diagrams to show transitions across individual snapshots of dynamic networks [14], [15]. These techniques have a common drawback that it is hard for users to track various network changes in animation. Timeline-based techniques draw a network at each time step and simultaneously display the complete set of time steps along a timeline in a static way [16]–[18]. This time-to-space mapping is able to provide a better evolution overview and facilitate insightful interactive exploration. It is, however, generally hard to determine an appropriate number of time steps to divide the entire period of time, particularly for continuously changing networks. Excessive time steps would reduce the readability of visualization in a limited screen space, while insufficient time steps may eliminate important information. Among existing timeline-based techniques, Massive Sequence View (MSV) [2], [3] could present dynamic network at a fine-grained level without the time step issue as detailed below.

B. MASSIVE SEQUENCE VIEW

MSV's history may trace back to Message Sequence Chart (MSC) [19], a standardized and widespread technique to visually describe the communication behavior between components within complex systems. Execution mural [8] is the early prototype of MSV, which extends MSC for visualizing object-oriented program executions. Later, Cornelissen *et al.* [4] and Holten *et al.* [20] designed a multi-view visualization system to analyze large program

execution traces, with a view called MSV. Recently, Van *et al.* [3] noticed a striking similarity between execution traces and network evolutions. They first introduced MSV into the dynamic behavior analysis of email networks and social networks.

MSV in these years has undergone a series of technical improvements. Linked-views [21], hierarchical navigation [21] and rich interactions [6] (i.e., filtering, brushing and zooming) enable users to explore industrial-sized MSCs and large program execution traces at different levels of detail. Density-based [4] and importance-based [8] anti-aliasing techniques make the compressed representations of the entire dynamic networks more accurate. Node reordering [3], [10], radial layout [2] and curved link [9] could reduce visual clutter. However, in difficult cases (e.g., communications between mass nodes occur densely and irregularly), these techniques may be unable to provide an effective global overview of MSV. In this paper, we propose to utilize sampling-based strategy to increase the overall readability of MSV.

C. SAMPLING FOR VISUAL CLUTTER REDUCTION

In the taxonomy proposed by Ellis and Dix [22], techniques for visual clutter reduction can be divided into three categories: spatial distortion, temporal and appearance. As one of the appearance techniques, sampling is relatively effective, low-cost and easy-to-implement, and consequently becomes popular [23]–[25] with various visualization techniques to improve readability. Chen *et al.* [26] employed a hierarchical multi-class sampling technique to declutter scatterplots. Ellis and Dix [27], Johansson and Cooper [28] and Bertini and Santucci [29] investigated visual clutter reduction in parallel coordinates with sampling. Cui *et al.* [30] measured the abstraction performance of sampling in parallel coordinates and scatterplots with their proposed metrics. Liu *et al.* [31] developed a blue-noise sampling scheme to reduce visual clutter in massive timeline visualization. To our knowledge, no work has been done to study sampling technique for reducing MSV's visual clutter.

Sampling is commonly used in graph visualization. Graph sampling randomly picks a subset of vertices and/or edges to construct a subgraph of original unfiltered graph. The usage includes graph drawing and graph mining to achieve highly readable network visualization and efficient analysis of large scale graphs. Graph sampling has three types of strategies: node-based, edge-based and traverse-based [11], [12], [32]. A number of graph sampling methods have been proposed to preserve various properties of the original graph in its sampled subgraph as far as possible, such as degree distribution, the number of triangles and cluster coefficients [33]. Graph sampling is closely associated with our work as MSV is a technique for analyzing dynamic network (graph). However, most existing graph sampling techniques are designed for static graph analysis without considering the temporal aspect.

Ahmed *et al.* [34] recently proposed a family of streaming graph sampling methods with consideration of the time dimension of network. They assumed that large scale graphs cannot fit in the main memory of computers at once for subsequent sampling operations. Therefore, their proposed methods adopted the streaming data input and continually update the sampled subgraph. In other words, the subgraph is still a static representation of the original graph in a streaming environment. Our sampling method emphasizes on maintaining the time-varying characteristics of the network. To some degree, our work is the first step to extend graph sampling to dynamic network analysis.

III. SAMPLING CONSIDERATIONS

Our goal is to achieve a highly readable overview of dynamic network in MSV. We employ the idea of removing the overlapping edges visualized on the screen while fulfilling feature preservation for dynamic network analysis. The idea can be formulated as a graph sampling problem. Given an initial dynamic network, we seek to find a subset of edges to construct a sampled dynamic network that preserves the evolving properties of an original dynamic network. Before doing so, we must consider a series of questions.

A. DATA MODEL

The data model of dynamic network is defined as directed graph $G = (V, E)$, where V denotes the set of all vertices, $E \subseteq V \times V \times T$ is the set of all edges, and $T = [t_{min}, t_{max}]$ denotes the entire period of time. Each edge $e \in E$ occurs at a specific time point $t_e \in T$ and consists of a vertex tuple (v_{src}, v_{sink}), which gives the source and sink vertex of edge e respectively. For two arbitrary vertices v_p and v_q, we define an edge sequence of vertex pair (v_p, v_q) as $E(v_p, v_q) = [e_0, \ldots, e_i, \ldots, e_n]$, where $e_i \in E \wedge \left(\left(v_{i\_src} = v_p \wedge v_{i\_sink} = v_q\right) \vee \left(v_{i\_src} = v_q \wedge v_{i\_sink} = v_p\right)\right) \wedge \left(t_{e_0} \leq \ldots \leq t_{e_i} \leq \ldots \leq t_{e_n}\right)$, that is, the sequence of all edges between the two nodes is arranged chronologically.

B. FEATURE PRESERVATION

A dynamic network has various features. Empirically, no single sampling method can preserve all the features [13], [35]. In this sense, the first step toward our desirable goal is to determine which features to preserve. Through the Gestalt principle, Van *et al.* [3] pointed out that MSV is an expert at presenting the tendency changes of network communication traffic, such as increase or decrease. Our sampling method is therefore expected to preserve such temporal features.

We notice that the features may behave differently or even oppositely at node and global levels. For example, the number of contacts between some nodes may decrease over time, whereas the communication traffic of the entire network may increase. We suggest the preservation of the features initially from the node level and then achieve the preservation of aggregated features at the global level. A single node and a node pair are both at the bottom level of a dynamic network. We are interested in node pairs because of MSV's block effect. Leveraged by the closure, proximity, and similarity of the Gestalt principle, many closely positioned edges between

two nodes are perceived as a solid block. Groups of sequential blocks will produce the block effect that helps users easily identify the temporal features. In summary, we plan to adopt a node-pair-wise and bottom-top manner to achieve feature preservation.

How can such temporal features be quantitatively measured? KDE is a well-studied statistical tool that can create a continuous density scalar field for all edges between two nodes to depict their time-varying contact frequency. In this work, we use KDE to generate a PDF for edges between a node pair. Specifically, KDE with Gaussian kernel is used. Gaussian kernel can depict the time-varying trend of communication frequency between nodes, and the resulting PDF is endowed with smoothness. Consider an edge sequence $E\left(v_p, v_q\right)$ with n edges between nodes v_p and v_q. The probability density of an edge e occurring at the time point t can be computed by using the following PDF:

$$f\left(v_p, v_q, t\right) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{i=1}^{n} e^{-\left(t-t_{e_i}\right)^2/2\sigma^2}, \tag{1}$$

where σ is a parameter called bandwidth controlling the smoothing degree of the density scalar field.

C. SAMPLING STRATEGY

The next consideration is to determine the basic sampling strategy. Graph sampling has three types of typical strategies: node-based, edge-based, and traverse-based [13]. The traverse-based strategy is appropriate for preserving network structural features. Nodes in MSV may stay active throughout the whole observation period. Edges only occur instantly at specific time points. Therefore, the edge-based strategy is relatively simple and flexible with consideration of only a specific local period.

After choosing the edge-based strategy, we wonder whether certain classic sampling methods can inspire us. Accept-reject (AR) sampling, a type of Monte Carlo methods, is known for its capability of generating random samples from arbitrary target probability distributions (PDFs). It applies to easy PDFs and difficult ones if given suitable proposal distributions (i.e., proposal PDFs) [36]. Such an advantage is highly appropriate to deal with various and unpredictable PDFs generated from numerous node pairs in a dynamic network. The idea behind AR sampling is as follows. AR sampling generates random samples from a target probability distribution $f\ (x)$ by using a proposal distribution $g\ (x)$. It first demands a suitable proposal distribution from which candidate samples can be drawn. These candidate samples are then either accepted or rejected depending on a test involving the ratio of the target and proposal densities as shown below:

$$u \leq f\ (x)\ /Fg\ (x), \tag{2}$$

where u is a 0-to-1 random variable generated for each candidate sample, and F is a pre-defined constant that controls the overall sample size. As F increases, the sample size decreases, indicating fewer accepted edges.

IV. EOD EDGE SAMPLING FOR MSV

Though detailed consideration is given, designing a desired sampling algorithm is still non-trivial. In this section, we first present design challenges. Then, we address them one by one and finally outline the complete sampling algorithm.

A. DESIGN CHALLENGES

After determining the sampling strategy and the features to preserve, we now have a clear picture of our sampling method. Referring to the basic idea of AR sampling, we sample a dynamic network following the probability density of edges from various node pairs. The goal is to preserve the time-varying features of network communication in a bottom-up way (from node pair level to global level) while improving the overall readability of MSV. Our goal and idea become clearer, but how to realize them is not immediately apparent. We still face many challenges coming from three aspects.

The first challenge is balancing feature preservation with visual clutter reduction. In general, edge dense areas contain much visual clutter in MSV. AR sampling only strives to ensure the density consistency before and after sampling. It does not provide any guarantee of visual clutter reduction, especially in high-density areas where readability may not be improved. In fact, this is a problem concerning how to generate a suitable proposal PDF for the relevant target PDF. In many scenarios, using a uniform distribution as proposal PDF is practical due to excellent computational efficiency and acceptable density retention. Using node pair *(b, d)* in Fig.3 as an example, since the peak area of the target PDF is close to the uniform distribution, the edges in this area are likely to be accepted in AR sampling. This characteristic is beneficial to maintain visual patterns in sampled MSV because the Gestalt principle indicates that high-density areas in MSV contain easy-to-perceive visual information, namely, the block effect. However, numerous edges among (a, b), (c, e), and (a, e) run across (b, d) in later part of MSV, causing

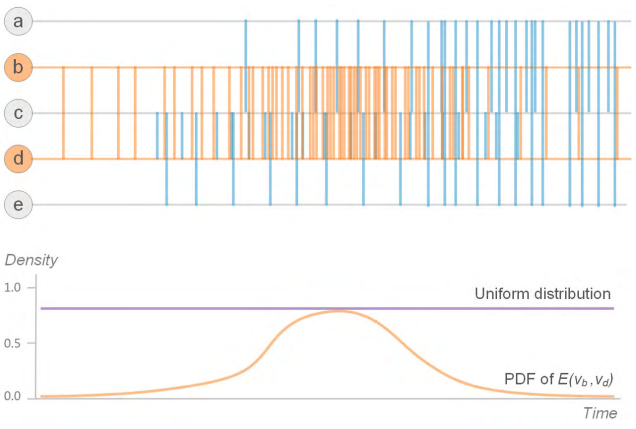


FIGURE 3. Illustration of target PDF and proposal PDF (uniform distribution). The orange curve is the target PDF of the highlighted node pair (*b, d*). The purple line is the proposal PDF in the form of a uniform distribution drawn using the maximum value of the target PDF.

severe visual clutter. The uniform distribution does not take into account this situation, and MSV's readability is not improved after sampling. In summary, the first challenge is to design a method for generating a suitable proposal PDF to realize the trade-off between feature preservation and visual clutter reduction.

The second challenge is related to edge length. The length of an edge is the distance between the positions on MSV's *Y*-axis of the source and sink vertices of that edge. All the vertices of a dynamic network are equally spaced on MSV's *Y*-axis, and their positions on this axis are fixed. As a consequence, the lengths of edges between different node pairs vary from one to another, whereas the edges between the same node pair have an identical edge length. The node-pair-wise sampling strategy we planned ignores the diversity of edge length, resulting in two problems. First, long edge is more likely to cause visual clutter than short edge because the former spans more nodes on the *Y*-axis of MSV. Second, the amount of edges of different lengths are often unevenly distributed, especially after applying node reordering strategies [3]. For example, minimizing edge length, a type of node reordering strategy, can increase the number of short edges and decrease the number of long edges. In this case, most visual patterns are presented by short edges, and the overall readability of MSV is improved. We should think about this challenge in our sampling process.

The third challenge concerns the discreteness of a dynamic network. AR sampling can only generate continuous candidate samples from continuous PDFs, but edges of dynamic networks are all discrete. Additionally, AR sampling randomly examines samples from the whole sample space, causing some samples never to be examined. Therefore, we should consider discrete sampling and full sample examination.

B. EDGE OVERLAPPING DEGREE

To solve the first challenge, we introduce the concept of edge overlapping degree (EOD). EOD is an indicator that quantitatively measures the overlapping degree of an edge in MSV by its neighboring edges, i.e., the amount of visual clutter in the area where the edge of interest is drawn. Considering the conceptual and computational complexity of EOD, we first introduce two auxiliary concepts: indistinguishable pixel area (IPA) and edge overlapping set (EOS). We then present an auxiliary operation used for easy calculation of EOD: edge decomposition (ED). Finally, we describe how to calculate EOD in detail.

1) INDISTINGUISHABLE PIXEL AREA

Empirically, when two edges occur at (approximately) the same time in a dynamic network, it is hard to visually distinguish them due to insufficient horizontal pixels. We thus define indistinguishable pixel distance (IPD) within which users cannot completely distinguish two separate edges:

$$IPD = ceil\left(\frac{W_{edge}}{2}\right) + \rho, \tag{3}$$

where W_{edge} is the width of an edge in pixel, $ceil(\cdot)$ is the rounding up function, and ρ ($\rho \geq 1$) is a user-defined parameter that is used to adjust the size of IPD and set to 1 by default. For an edge e, we use function $HP(\cdot)$ to obtain the central position of the edge on the *X*-axis:

$$HP(e) = \frac{W_{MSV}}{t_{max} - t_{min}}(t_e - t_{min}), \tag{4}$$

where W_{MSV} is the length of MSV's timeline in pixel; t_{min} and t_{max} are the beginning and end time of dynamic network, respectively; and t_e is the occurring time of edge e.

With the preceding basis, we define that the IPA of edge e is centered around its central position, extending pixels of IPD to the left and right such that IPA forms a rectangular area with the width of two IPDs and the height as that of MSV canvas. Mathematically, IPA can be expressed as follows: $[HP(e) - IPD, HP(e) + IPD]$. As shown in Fig.4, edge e_1 occurs at time t_1. We take the t_1 time point as the center and extend one IPD left and right to obtain the IPA of e_1.

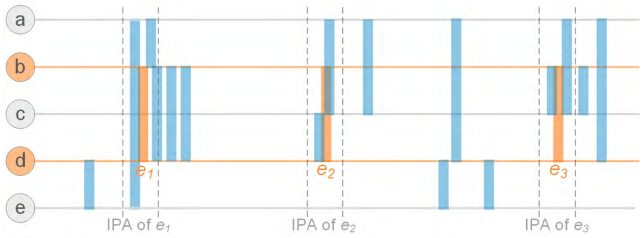


FIGURE 4. Illustration of IPA and EOD.

To conclude, if other edges fall into the range of a specific edge's IPA (considering the width of edge, those edges are also included that only a portion of their width falls into the IPA), then we are unable to visually distinguish these edges from the edge.

2) EDGE OVERLAPPING SET

For an edge, will all the edges in its IPA overlap it and result in visual clutter interfering with users' perception? The answer is negative. We need to consider the relationship between the vertices of IPA edges and the vertices of the edge from the perspective of the *Y*-axis. IPA edges can be divided into three categories based on different relationships: trivial edge, similar edge, and overlapping edge.

We first introduce how to determine the location of vertices on the *Y*-axis. All vertices of a dynamic network are equally spaced on the *Y*-axis of MSV. Every edge has two vertices, each of which has a corresponding position on the *Y*-axis. We assume that the origin of the *Y*-axis is located in the top-left corner of MSV canvas, and each edge's source vertex is closer to the origin of the *Y*-axis than the sink vertex. We define functions $VPbegin(\cdot)$ and $VPend(\cdot)$ to derive the vertical position of source vertex and sink vertex of an edge, respectively.

We can now easily classify the three categories of IPA edges according to the vertical positions of their vertices.

Assume that edge e is the edge of interest and e_i is an arbitrary edge inside the IPA of edge e. If $VPend\,(e_i) <= VPbegin\,(e)$ or $VPbegin\,(e_i) >= VPend\,(e)$, then e_i is called a trivial edge that does not overlap edge e, thereby introducing no visual clutter. If $VPbegin\,(e_i) = VPbegin\,(e)$ and $VPend\,(e_i) = VPend\,(e)$, then e_i is called a similar edge produced by the same vertex tuple as edge e. Such edges benefit the block effect in MSV that facilitates the perception of visual patterns. Except for trivial edges and similar edges, other IPA edges are all overlapping edges that really overlap edge e and lead to MSV's visual clutter. For example, in Fig.4, there are three edges in the IPA of edge e_1: edges e_{ae}, e_{ab} and e_{bd}. Edge e_{ae} is an overlapping edge, e_{ab} is a trivial edge, and e_{bd} is a similar edge. Through similar analysis, e_2 has two edges in its IPA, and edges e_{cd} and e_{ac} are all overlapping edges.

To quantify the degree of edge overlapping, we define the set of all overlapping edges within the IPA of a specific edge as $EOS\,(e)$. For example, in Fig.4, $EOS(e_1)$ is $\{e_{ae}\}$ and $EOS(e_2)$ is $\{e_{ac}, e_{cd}\}$.

3) EDGE DECOMPOSITION

Edges in $EOS\,(e)$ can overlap with edges e in different ways: some edges may overlap edge e completely, while others may overlap only a portion of edge e. For instance, edge e_1 in Fig.4 is 100% overlapped by e_{ae}, whereas e_2 is only 50% overlapped by e_{cd}. We thus introduce an operation called edge decomposition for easy calculation of overlapping degree. Considering that all vertices of a dynamic network are equally spaced on MSV's Y-axis, we can decompose an edge into a set of node pairs that are directly adjacent and equidistant. In general, for edge e with vertex tuple (v_i, v_j), edge decomposition can be defined as:

$$ED\,(e) = \left\{(v_i, v_{i+1})\,, (v_{i+1}, v_{i+2})\,, \ldots, (v_{i+n}, v_j)\right\}, \quad (5)$$

where nodes $v_{i+1}, v_{i+2}, \cdots, v_{i+n}$ represent all the nodes between nodes v_i and v_j, and the Y-axis positions of $v_i, v_{i+1}, v_{i+2}, \ldots, v_{i+n}, v_j$ increase gradually. After decomposition, the length of each small segment of the original edge is $\frac{H_{MSV}}{N_{node}-1}$, where N_{node} is the number of all the nodes in the dynamic network, and H_{MSV} is the height of the MSV canvas. For example, the vertex tuple of edge e_1 is (b, d) in Fig.4; we can decompose edge e_1 into $\{(b, c)\,, (c, d)\}$ (c is vertically located between b and d).

We notice that different edges in the same EOS may repeatedly overlap the same part of the edge of interest. The repeated overlapping will not always enhance visual clutter. In Fig.4, edges e_{ac} and e_{bc} both overlap the upper half of e_3, resulting in repeated overlapping. When the opacity is set to over 50%, the resulted visual effects are the same between three overlapped and two overlapped edges. In this work, we do not consider repeated overlapping when measuring visual clutter. To this end, we decompose all the edges in EOS and apply a union operation to the EOS to eliminate repeated overlapping. For edge e, the EOS that eliminates repeated overlapping can be expressed as:

$$EOS^*\,(e) = ED\,(e_{i1}) \cup ED\,(e_{i2}) \cup \ldots \cup ED\,(e_{in}), \quad (6)$$

where $e_{i1}, e_{i2}, \ldots, e_{in} \in EOS\,(e)$. For example, for edge e_3 in Fig.4, the two edges of its EOS are decomposed into $\{(b, c)\}$ and $\{(a, b)\,, (b, c)\}$. The EOS without repeated overlapping is $\{(a, b)\,, (b, c)\}$.

4) EOD COMPUTATION

We first use three examples to demonstrate how to intuitively calculate EOD. For edge e_1 in Fig.4, its EOS only contains edge e_{ae} that overlaps e_1 completely in the vertical direction. We thus define the EOD of edge e_1 as 1. For edge e_2, edge e_{ac} overlaps the upper half of edge e_2, whereas the lower half is overlapped by edge e_{cd}. Given that edges e_{ac} and e_{cd} together overlap edge e_2 completely, the EOD of e_2 is also set to 1. For edge e_3, its EOS includes edges e_{ac} and e_{bc}, both of which overlap the same upper half of e_3; the EOD of edge e_3 is set to 0.5.

Through the above examples, the calculation of EOD is concluded into two steps. The first step is to obtain the EOS of an edge. The second step is to accurately measure the degree of visual clutter introduced by overlapping between the edge and its EOS. To complete the second step, we employ edge decomposition and intersection operation. For example, in Fig.4, $EOS^*\,(e_3)$ is $\{(a, b)\,, (b, c)\}$ after elimination of the repeated overlap, and the edge decomposition of e_3 is $\{(b, c)\,, (c, d)\}$. After the intersection of the two sets, the resulting set is $\{(b, c)\}$. This means only the upper half of e_3 is overlapped by its EOS. The normalized EOD of e_3 is 0.5. In general, for edge e, we define $EOD\,(e)$ as:

$$EOD\,(e) = \frac{|ED\,(e) \cap EOS^*\,(e)|}{|ED\,(e)|}, \quad (7)$$

where $|\cdot|$ is used to count the number of set elements. The value of EOD ranges from 0 to 1. A value of 0 means an empty EOS of which no edges introduce visual clutter, and a value of 1 indicates that the edges in EOS completely overlap the edge of interest.

EOD is an edge-level indicator for visual clutter. We can use EOD to generate a new proposal PDF for replacing uniform distribution in AR sampling. We first calculate the EOD of each edge between vertex tuple (v_p, v_q) and add the maximum value of the corresponding target PDF to each EOD value. Finally, we generate a new proposal PDF by smoothly connecting these EOD values. To summarize, we can compute the proposal PDF value of an edge e at time point t by using the following formula:

$$g_F\left(v_p, v_q, t\right) = F\left(max\left(f\left(v_p, v_q, t\right)\right) + EOD\,(e)\right), \quad (8)$$

where $f\left(v_p, v_q, t\right)$ is the target PDF, and F is a user-defined constant as in AR sampling which we have introduced in Section 3.3. Here, $g_F\left(v_p, v_q, t\right)$ with subscript F indicates that F is a hyperparameter of $g\left(v_p, v_q, t\right)$.

The new proposal PDF possesses three advantages. First, it takes the maximum value of the target PDF as the benchmark and therefore can obtain a similar functionality as uniform distribution to maintain the probability density distribution of edges before and after sampling. Second, the PDF can further reduce visual clutter because it adds a corresponding EOD value to each edge on the benchmark by considering edge overlapping. Third, the overlapping information can be stored as a new property of edge for future use such as facilitating local interactive exploration in MSV.

C. EDGE LENGTH FACTOR

The new proposal PDF can achieve the balance between feature preservation and visual clutter reduction. However, sampling phenomenon related to edge length exists. When the length of an edge is large, its EOD tends to be small, and it can be easily accepted in sampling. By contrast, when the length of an edge is small, the edge is more likely to be completely overlapped, resulting in a large EOD value and a high rejection probability. In general, the influence of edge length on sampling probability is undesirable. The two main reasons have been described as the second challenge in Section 4.1.

To address this challenge, our idea is that long edges should be accepted with relatively small probability while considering the quantity distribution of edges of different lengths. We introduce edge length factor (ELF) into the design of EOD-based proposal PDF:

$$g_F(v_p, v_q, t) = F\left(max\left(f\left(v_p, v_q, t\right)\right) + EOD\left(e\right)\right) + \left(1 - w_e\right) l_e, \tag{9}$$

$$l_e = \frac{length\left(e\right)}{H_{MSV}}, \quad w_{l_e} = \frac{num\left(length\left(e\right)\right)}{N_{edge}}, \tag{10}$$

where l_e is the normalized length of the edge of interest, w_{l_e} is the weight for the corresponding edge length, $length(\cdot)$ is used to obtain the length of an edge, $num(\cdot)$ is a function for calculating the number of edges with specific length, and N_{edge} is the total number of edges in a dynamic network.

The new proposal PDF is optimized by ELF with a penalty term and an incentive term. l is the penalty term. If the length of an edge is large, then the penalty for this edge will also be large and the resulting probability of accepting it will be relatively low, and vice versa. w is the incentive term. If the weight for the edges of a specific length is large, then the accepting probability will be relatively high. In general, w can be automatically set according to the quantity distribution of edges of different lengths before sampling. From another point of view, w can also be regarded as a user-defined parameter that provides the user with free observation modes. For example, to preserve edges of moderate length, we can set a large weight for these edges so that the resulting probability of accepting them increases.

D. STREAMING EDGE SAMPLING

Prior to sampling, we need to address the third challenge stated in Section 4.1. We propose to use edge stream to discretely process every edge. The sampling process first arranges all edges between a vertex pair to be sampled in a chronological order, which can be considered an edge stream. The process generates candidate samples from the edge stream by sequentially taking each edge as a candidate sample. In this manner, the sampling process can examine every discrete edge in a dynamic network.

The sampling process is described as follows. First, a candidate sample is sequentially generated from the edge stream. For the candidate sample, a random number u between 0 and 1 is generated. The candidate sample is either accepted or rejected according to the following Boolean value:

$$u \leq \frac{f\left(v_p, v_q, t\right)}{g_F\left(v_p, v_q, t\right)}. \tag{11}$$

To conclude, sampling a dynamic network with our proposed method includes four steps: initialization, calculating target PDF, designing a proposal PDF, and streaming sampling. In step one, users input a dynamic network and the expected sample size F; we obtain the statistics of the dynamic network and set weight w for different edge lengths. In step two, target PDF $f\left(v_p, v_q, t\right)$ for node pair $\left(v_p, v_q\right)$ to be sampled is estimated by KDE. In step three, we calculate $EOD\left(e\right)$ for each edge e between the node pair and derive the proposal PDF $g_F\left(v_p, v_q, t\right)$ by using EOD, w_e and l_e. In step four, we sequentially examine every edge in the edge stream (sequence) $E\left(v_p, v_q\right)$. The pseudo-code for the entire sampling process is given in Algorithm1.

Algorithm 1 EOD Edge Sampling

Require: dynamic network $G = (V, E)$, F
Ensure: sampled dynamic work $G_1 = (V, E_1)$

```
Initialization
Obtain the statistics of E and set w for different edge lengths
for every pair of nodes (v_p, v_q) in V do
    Compute the target PDF f(v_p, v_q, t) by using KDE
    Compute the EOD(e) of each edge e
    Design proposal PDF g_F(v_p, v_q, t) by using EOD, l_e and w_{l_e}
    for e in E(v_p, v_q) do
        Generate random value u by using uniform distribution U(0, 1)
        if u ≤ f(v_p, v_q, t_e) / g_F(v_p, v_q, t_e) then
            add e to E_1
        end if
    end for
end for
```

V. EVALUATION

In this section, we use two real-world datasets to demonstrate the effectiveness of the proposed method. We then evaluate its performance on feature preservation and visual clutter reduction by using three quantitative indicators.

A. CASE STUDY

The first case involves the use of a popular dataset, Enron email dataset, which is widely used in dynamic network studies [3], [5], [37]. Enron, a former energy service company, is known for the biggest American bankruptcy due to accounting fraud. In February 2001, Jeffrey Skilling replaced Kenneth Lay as CEO of the company. In July 2001, Skilling resigned suddenly and Lay took over once again. In October 2001, the Securities and Exchange Commission (SEC) started an investigation into Enron. In December 2001, Enron filed for bankruptcy and many executives were sentenced to prison.

Fig.2(a) shows the whole dynamic network of the Enron email dataset. All nodes are sorted by the node reordering strategy of minimizing edge length [3]. Each of the aforementioned events can lead to a sudden change in email traffic. However, we cannot clearly identify these events due to MSV's poor readability. Fig.2(b) shows the clutter-reduced MSV by our method. With the improved readability, we can clearly observe three sudden drops in email communication, reflecting the evolution of Enron scandal. As the arrows shown in the figure, the three drops correspond to the CEO replacement from Lay to Skilling in February 2001, the sudden resignation of Skilling in July 2001, and the bankruptcy petition of Enron in December 2001.

Fig.2(c) shows the comparison of the temporal tendency of communication traffic in the two MSVs. The X-axis represents the timeline as that of the MSVs, and the Y-axis shows the percentage of the edge number in a time bin to all the edges of the entire network. The timeline is divided into 50 time bins in this case. The two curves stick together most of the time, indicating that our method effectively preserves the time-varying features of the network traffic. In addition, if users just focus on Fig.2(a), then they may misunderstand that after October 2000, the email communication peaked and lasted until February 2002. However, the actual peak only existed after October 2001, at that time the SEC started an investigation into Enron. The sampled MSV in Fig.2(b) helps us observe this correct trend.

The second case involves the high-school dataset containing time-stamped face-to-face contacts between high-school students for a week [38]. The dataset is collected by wearable sensors with 20 second recording interval. We choose the first day's data from 2012/11/19 15:00 to 24:00, and the corresponding social dynamic network has 151 nodes and 9,957 edges.

Fig.5(b) shows the sampled result on the basis of Fig.5(a). The overall readability is obviously improved, with accurate presentation of the temporal trend of the network traffic.

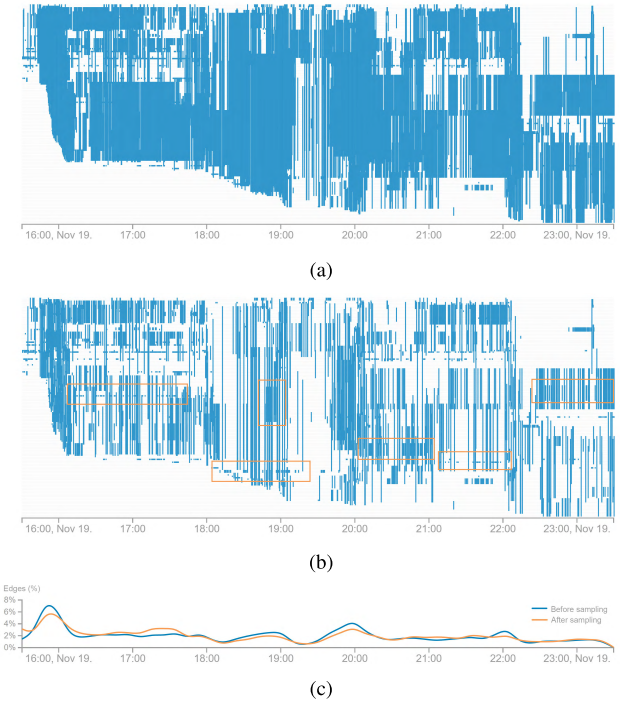


FIGURE 5. High-school student face-to-face contact dataset. The nodes in the two MSVs are sorted alphabetically. After sampling (b), many contact pairs with frequent communication stand out, some in orange boxes, which are invisible before sampling (a). (a) MSV before sampling. (b) MSV after applying our sampling method. (c) The distributions of edge counts of the high-school dataset before and after sampling.

For example, the peak of face-to-face contacts appears at around 16 o'clock, which may indicate that school was over at that time; the peak is clearly presented in Fig.5(b), instead of being identified as several possible peaks in Fig.5(a). On close inspection, as shown in the orange boxes in Fig.5(b), many contact pairs with frequent communication stand out, and these pairs are invisible in Fig.5(a). We check the specific information from the data and find that many of the contact pairs belong to the same class and are supposed to be close friends.

In summary, the first case reflects that our sampling method can make MSV clear and accurate in interpreting the time-varying features of the dynamic network from a macro global network level. The second case suggests that from a micro node pair level, our sampling method can effectively facilitate the identification of contact pairs with frequent communication.

B. QUANTITATIVE ANALYSIS

We design an indicator (KS distance) to quantify the sampling performance on feature preservation and two indicators (edge overlapped rate and edge hidden rate) to quantify the readability improvement of MSV before and after sampling.

KS distance measures the similarity of the time-varying trend of network traffic before and after sampling with values ranging from 0 to 1. The smaller the value is, the more similar the two trends of network traffic are, implying the

TABLE 1. The time spent for both preprocessing and sampling of the three algorithms on the three datasets.

Data Name	Data Description	Nodes	Edges	Time span	Preprocessing time (seconds)			Sampling time (seconds)		
					Random	AR	Our	Random	AR	Our
Calls [39]	Phone call	75	10,636	10 months	0	13.97	31.29	0.16	0.01	0.02
Sociopatterns-infectious [40]	Face-to-face contact	410	17,298	8 hours	0	17.21	58.98	0.16	0.03	0.04
Opsahl-ucforum [41]	Forum social network	899	33,720	5 months	0	23.17	673.43	0.64	0.08	0.11

better performance of sampling algorithm to preserve such trend. The calculation includes three steps. First, the entire observation period of a dynamic network is partitioned into small time bins. Then, the ratio of the edge number in each bin to the total edge number of the entire network is calculated. Finally, the Kolmogorov−Smirnov statistics [34], [33] is used to measure the similarity of the two distributions before and after sampling.

Edge overlapped rate is used to measure the degree of visual clutter caused by overlapping edges. These edges are the major factor that reduces the overall readability of MSV [2]. Thus, comparing the edge overlapped rates of different sampling results is helpful in determining which sampling methods can improve the overall readability of MSV significantly. Edge overlapped rate is defined as the ratio of the number of overlapping edges to the total number of edges of the network, with values ranging from 0 to 1. The smaller the value is, the less visual clutter appears. For an edge, we only judge whether its immediately subsequent edge overlaps with it, that is, whether the subsequent edge appears in its EOS. If the judgment is true, the edge is an overlapping edge.

Edge hidden rate measures the degree of information loss in MSV. Due to inevitable edge overlapping and/or overplotting, some edges in MSV may become unobserved. As a result, the information about the edges gets lost in the visualization result of MSV. We use the edge hidden rate to measure the percentage of unobserved edges in all edges of the entire network in MSV. The values of the edge hidden rate range from 0 to 1. The smaller the value is, the less hidden edges appear in the MSV. We assume an edge would be completely hidden by visual clutter, that is, invisible to users, if more than 50% of its entire length is overlapped by other edges. The calculation method is similar to that of EOD.

Three datasets different from the two case datasets are used in our experiment (Table 1). These datasets are commonly used in dynamic network analysis. To carry out a comparative analysis, we select two reference algorithms. The first one is random sampling, which randomly selects a certain number of edges; the second one is the classic AR sampling, which uses the uniform distribution (set to the maximum value of a target PDF) as its proposal PDF. The three algorithms are used on the three datasets at 10 different sample sizes. At each sample size, the process is repeated 10 times. We record the average of the three indicators and the average time consumption. In the experiment, the canvas of MSV is set to 1650×850 in size, the width of an edge in MSV is set to 1 pixel, the Gaussian kernel bandwidth is 5, the IPD's ρ is 1, and the weight w is automatically calculated by default. The experiment is conducted on a Dell OptiPlex 7040 desktop.

Fig.6 shows the results of the indicator analysis. For KS distance indicator, our method achieves a comparable performance to AR sampling, with random sampling being the worst. The KS value of our method is constant around 0.2 at multiple sample sizes, which can be regarded as a good KS value. In terms of the two indicators of visual clutter reduction, our method significantly outperforms the other two methods. The edge overlapped rate values of AR sampling and random sampling at most sample sizes are close to the situation without sampling (the rightmost maximum), which reflects that they almost fail to reduce the visual cluster in MSV. By comparing the three datasets, we find that the latter two indicator values increase as the data volume increases, that is, as the number of edges increases, the difficulty of visual clutter reduction increases.

Table 1 lists the average time consumptions of the three algorithms. We record the preprocessing time and sampling time of each sampling test. The preprocessing of AR sampling is mainly conducted to calculate the target PDFs of node pairs. For our method, the preprocessing mainly includes the calculation of the target PDFs and EOD-based proposal PDFs of node pairs. The experimental result shows that our method consumes a long preprocessing time. The calculation of EOD is the most time-consuming step in our algorithm. The worst-case complexity is $O\left(n^2\right)$, where n is the number of edges, because the EOD calculation of each edge may examine whether all edges overlap with the edge. The use of IPA can greatly reduce the amount of computation, so the actual algorithm complexity is less than $O\left(n^2\right)$. In terms of sampling time, all the three algorithms are very fast (less than 1 second).

VI. DISCUSSION

In this section, we discuss the parameter settings of our proposed algorithm and present the limitations of the study and further directions.

The most important parameter of the proposed algorithm is the sampling factor F. As F increases, the sample size decreases (i.e., fewer accepted edges). Through empirical analysis, we set F between 0 and 2. Obtaining the optimized F deserves a future study. A possible solution may be looking for inflexions of the curves of the three indicators at different

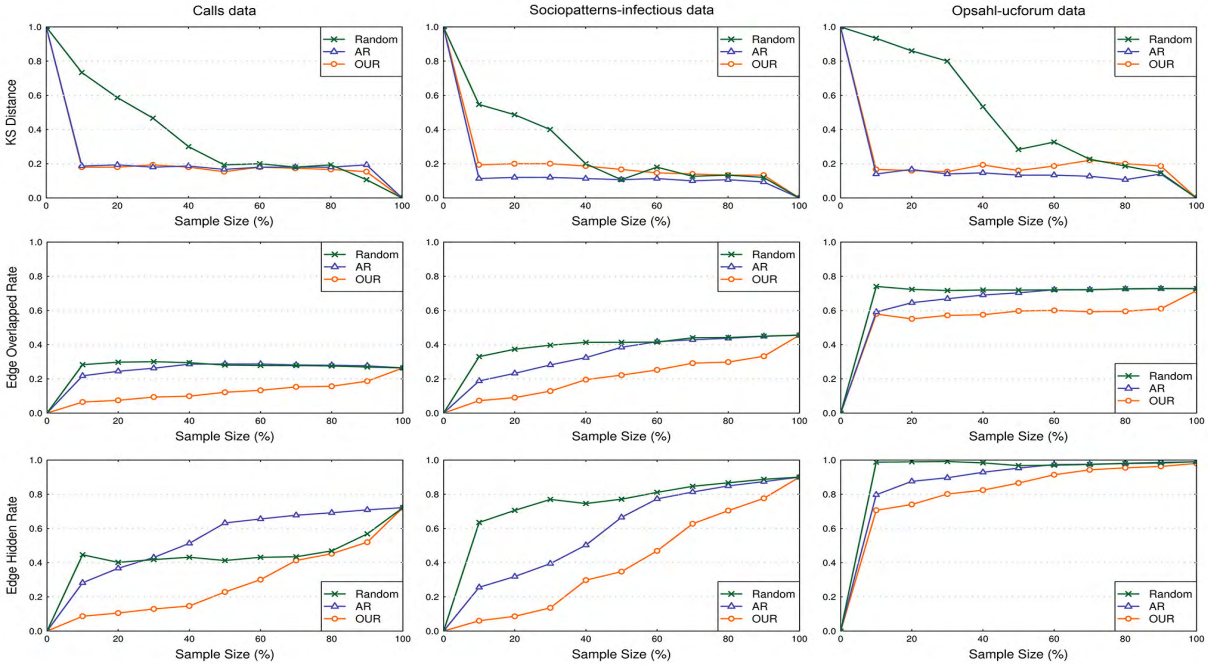


FIGURE 6. Quantitative evaluation results. The line charts of the same indicator are placed in the same row, with each chart showing the result of one dataset. The green, purple, and orange curves represent random sampling (Random), AR sampling (AR), and our sampling method (Our), respectively. The *X*-axes represent the size of the sample set. The origin of *X*-axis indicates the empty sample set, and 100% means the entire dynamic network edge set (i.e., no sampling).

sample sizes. The algorithm includes three other parameters. ρ controls the width of IPA. The larger ρ is, the more edges participate in calculating EOD. We set ρ to 1 or 2 pixels (1 by default). The second parameter is weight w for specific edge length. The default setting is calculated automatically by the quantity distribution of edges of different lengths. Users can also manually set w for observing the pattern formed by the edges of a specific length. The third parameter is the bandwidth σ of Gaussian kernel, which controls the smoothing degree of PDFs. We use the rule of thumb to select the optimal bandwidth [42].

Our algorithm has a nature of randomness, which introduces a few problems. First, applying our algorithm to a given dataset many times may result in slightly different sampling results. Second, some edges without overlapping in unsampled MSVs have a certain probability to be discarded after sampling. Last, in extreme cases, a few patterns clear in unsampled MSVs may become unclear or even lost after sampling because some edges related to the patterns are randomly rejected. We plan to reduce the randomness of our algorithm and investigate how to avoid discarding non-overlapping edges.

Our algorithm spends time mostly on preprocessing, especially for calculating EOD. We will therefore enhance the efficiency of data preprocessing. The algorithm mainly preserves the time-varying trend of network communication traffic; in fact, there are many other features in a dynamic network, such as structural evolution. We plan to investigate how to preserve numerous features in sampling and involve users in evaluating the feature-preserving aspect.

VII. CONCLUSION

This paper proposes an edge sampling method to reduce visual clutter in MSV and preserve network evolving features. EOD is carefully designed to be an edge level indicator of visual clutter. We use KDE to characterize time-varying features of node pairs and generate PDFs to realize feature preservation in a bottom-up manner. Edge length factor and streaming processing are used to enhance sampling effect. Both the case studies and quantitative analysis demonstrate the effectiveness of our method. This work provides a starting point that extends graph sampling to dynamic network visualization and visual analytics.

REFERENCES

[1] F. Beck, M. Burch, S. Diehl, and D. Weiskopf, "The state of the art in visualizing dynamic graphs," in *Proc. Eurograph. Conf. Vis.*, Jul. 2014, pp. 83–103.

[2] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, "Dynamic network visualization withExtended massive sequence views," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 8, pp. 1087–1099, Aug. 2014.

[3] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, "Reordering massive sequence views: Enabling temporal and structural analysis of dynamic networks," in *Proc. IEEE Pacific Vis. Symp.*, Feb. 2013, pp. 33–40.

[4] B. Cornelissen, D. Holten, A. Zaidman, L. Moonen, J. J. Van Wijk, and A. van Deursen, "Understanding execution traces using massive sequence and circular bundle views," in *Proc. IEEE Int. Conf. Program Comprehension*, vol. 81, no. 12, Jun. 2007, pp. 49–58.

[5] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "GraphScope: Parameter-free mining of large time-evolving graphs," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2007, pp. 687–696.

[6] S. G. Eick and A. Ward, "An interactive visualization for message sequence charts," in *Proc. Int. Workshop Program Comprehension*, Mar. 1996, pp. 2–8.

[7] D. F. Jerding, J. T. Stasko, and T. Ball, "Visualizing interactions in program executions," in *Proc. Int. Conf. Softw. Eng.*, May 1997, pp. 360–370.

[8] D. F. Jerding and J. T. Stasko, "The information mural: A technique for displaying and navigating large information spaces," *IEEE Trans. Vis. Comput. Graphics*, vol. 4, no. 3, pp. 257–271, Jul. 1998.

[9] 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.

[10] C. D. G. Linhares, B. A. N. Travençolo, J. G. S. Paiva, and L. E. C. Rocha, "DyNetVis: A system for visualization of dynamic networks," in *Proc. ACM Symp. Appl. Comput.*, Apr. 2017, pp. 187–194.

[11] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proc. ACM Knowl. Discovery Data Mining (KDD)*, Philadelphia, PA, USA, Aug. 2006, pp. 631–636.

[12] D. Rafiei, "Effectively visualizing large networks through sampling," in *Proc. IEEE Vis. (VIS)*, Oct. 2005, pp. 375–382.

[13] Y. Wu, N. Cao, D. Archambault, Q. Shen, H. Qu, and W. Cui, "Evaluation of graph sampling: A visualization perspective," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 401–410, Jan. 2016.

[14] T. E. Gorochowski, M. di Bernardo, and C. S. Grierson, "Using aging to visually uncover evolutionary processes on networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 8, pp. 1343–1352, Aug. 2011.

[15] B. Bach, E. Pietriga, and J.-D. Fekete, "GraphDiaries: Animated transitions andtemporal navigation for dynamic networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 5, pp. 740–754, Nov. 2013.

[16] M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf, "Parallel edge splatting for scalable dynamic graph visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2344–2353, Dec. 2011.

[17] Y. Wu, N. Pitipornvivat, J. Zhao, S. Yang, G. Huang, and H. Qu, "egoSlider: Visual analysis of egocentric network evolution," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 260–269, Jun. 2016.

[18] C. Vehlow, F. Beck, and D. Weiskopf, "Visualizing dynamic hierarchies in graph sequences," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 10, pp. 2343–2357, Oct. 2016.

[19] Wikipedia. *Message Sequence Chart*. Accessed: Jan. 19, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Message_sequence_chart

[20] D. Holten, B. Cornelissen, and J. J. van Wijk, "Trace visualization using hierarchical edge bundles and massive sequence views," in *Proc. IEEE Int. Workshop Visualizing Softw. Understand. Anal.*, Jul. 2007, pp. 47–54.

[21] M. Belachew, A. Gandhi, S. Gadhwala, and R. K. Shyamasundar , "MSC+: A generalized hierarchical message sequence charts," in *Proc. Int. Conf. Inf. Technol.*, Dec. 2000, pp. 183–189.

[22] G. Ellis and A. Dix, "A taxonomy of clutter reduction for information visualisation," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 6, pp. 1216–1223, Nov. 2007.

[23] A. Dix and G. Ellis, "*By chance* enhancing interaction with large data sets through statistical sampling," in *Proc. Work. Conf. Adv. Vis. Interfaces*, Jan. 2002, pp. 167–176.

[24] G. Ellis, E. Bertini, and A. Dix, "The sampling lens: Making sense of saturated visualisations," in *Proc. Conf. Human Factors Comput. Syst.*, Apr. 2005, pp. 1351–1354.

[25] E. Bertini, "A sampling approach to deal with cluttered information visualizations," *Pediatrics*, vol. 51, no. 2, p. 292, 2008.

[26] H. Chen *et al.*, "Visual abstraction and exploration of multi-class scatterplots," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1683–1692, Dec. 2014.

[27] G. Ellis and A. Dix, "The plot, the clutter, the sampling and its lens: Occlusion measures for automatic clutter reduction," in *Proc. Work. Conf. Adv. Vis. Interfaces*, May 2006, pp. 266–269.

[28] J. Johansson and M. Cooper, "A screen space quality method for data abstraction," *Comput. Graph. Forum*, vol. 27, no. 3, pp. 1039–1046, May 2008.

[29] E. Bertini and G. Santucci, "Improving visual analytics environments through a methodological framework for automatic clutter reduction," *J. Vis. Lang. Comput.*, vol. 22, no. 3, pp. 194–212, Jun. 2011.

[30] Q. Cui, M. Ward, E. Rundensteiner, and J. Yang, "Measuring data abstraction quality in multiresolution visualizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 5, pp. 709–716, Oct. 2006.

[31] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu, "Analyzing the training processes of deep generative models," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 77–87, Jan. 2018.

[32] C. Doerr and N. Blenn, "Metric convergence in social network sampling," in *Proc. 5th ACM Workshop HotPlanet*, 2013, pp. 45–50.

[33] A. S. Maiya and T. Y. Berger-Wolf, "Benefits of bias: Towards better characterization of network sampling," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 105–113.

[34] N. K. Ahmed, J. Neville, and R. Kompella, "Network sampling: From static to streaming graphs," *ACM Trans. Knowl. Discovery Data*, vol. 8, no. 2, Nov. 2012, Art. no. 7.

[35] Q. H. Nguyen, S.-H. Hong, P. Eades, and A. Meidiana, "Proxy graph: Visual quality metrics of big graph sampling," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 6, pp. 1600–1611, Jun. 2017.

[36] C. P. Robert, *Monte Carlo Methods*. Hoboken, NJ, USA: Wiley, 2004.

[37] J. Shetty and J. Adibi, "Discovering important nodes through graph entropy the case of enron email database," in *Proc. 3rd Int. Workshop Link Discovery*, Aug. 2005, pp. 74–81.

[38] R. Mastrandrea, J. Fournet, and A. Barrat, "Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys," *PLoS ONE*, vol. 10, no. 9, p. e0136497, Jun. 2015.

[39] A. Madan, M. Cebrian, S. Moturu, K. Farrahi, and A. S. Pentland, "Sensing the 'health state' of a community," *IEEE Pervasive Comput.*, vol. 11, no. 4, pp. 36–45, Dec. 2012.

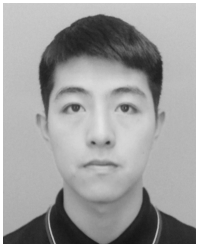
[40] J. Stehlé *et al.*, "Simulation of an SEIR infectious disease model on the dynamic contact network of conference," *BMC Med.*, vol. 9, no. 1, p. 87, Jul. 2011.

[41] T. Opsahl, "Triadic closure in two-mode networks: Redefining the global and local clustering coefficients," *Social Netw.*, vol. 35, no. 2, pp. 159–167, 2013.

[42] B. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.

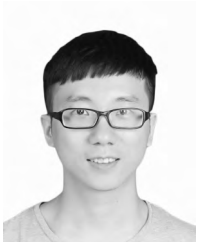
YING ZHAO received the Ph.D. degree in computer science from Central South University, Changsha, China, in 2014. He is currently an Associate Professor with the School of Information Science and Engineering, Central South University. His main research interests include visualization and visual analytics. His work received five awards at VAST Challenge 2012–2015, the Best Paper Award of the International Symposium on Cyberspace Safety and Security in 2013, the Honorable Mention Paper Award and Best Poster Award of the IEEE PacificVis Symposium 2018, and the Best Paper Award of the ChinaVis Conference 2018.

YANMIN SHE is currently pursuing the master's degree with Central South University. Her research interests are visual analytics and data mining.



WENJIANG CHEN is currently pursuing the master's degree with Central South University. His research interests are data mining and machine learning.

YUTIAN LU is currently pursuing the bachelor's degree with Central South University. Her research interests are visual analytics and data mining.



JIAZHI XIA received the B.S. and M.S. degrees in computer science from Zhejiang University and the Ph.D. degree in computer science from Nanyang Technological University. He is currently an Associate Professor with the School of Information Science and Engineering, Central South University. His research interest includes visualization and visual analytics.

WEI CHEN received the bachelor's and Ph.D. degrees from Zhejiang University, China. From 2000 to 2002, he was a Visiting Ph.D. Student in the Fraunhofer Institute for Graphics, Darmstadt, Germany. From 2006 to 2008, he was a Visiting Scholar at Purdue University, where he was involved in PURPL with Prof. D. S. Ebert. In 2009, he was promoted as a Full Professor of Zhejiang University, where he is currently a Professor with the State Key Lab of CAD&CG. His current research interests include visualization, visual analytics, and bio-medical image computing. He has performed research in visualization and visual analysis and published more than 40 IEEE/ACM Transactions and IEEE VIS papers. His Chinese books on visualization are the unique books on visualization in China. He actively served in many leading conferences and journals, such as the IEEE PacificVIS Steering Committee and the ChinaVIS Steering Committee, and paper co-chairs of IEEE PacificVIS, IEEE LDAV and ACM SIGGRAPH Asia VisSym. He is also the Associate EIC of JVLC and an Associate Editor of IEEE CG&A and JOV.

JUNRONG LIU received the M.S. degree in computer science and technology from the Beijing University of Post and Telecommunication. She is currently a Research Assistant with the Institute of Information Engineering, CAS. Her research interests include situation awareness, threat intelligence, and network security visualization.

FANGFANG ZHOU received the Ph.D. degree in control science and engineering from Central South University in 2007. She is currently a Professor with Central South University. Her research interests include visualization and virtual reality.