Eiffel: Evolutionary Flow Map for Influence Graph Visualization

Yucheng Huang, Lei Shi, Yue Su, Yifan Hu, Hanghang Tong, Chaoli Wang, *Senior Member, IEEE*, Tong Yang, Deyun Wang, and Shuo Liang

Abstract—The visualization of evolutionary influence graphs is important for performing many real-life tasks such as citation analysis and social influence analysis. The main challenges include how to summarize large-scale, complex, and time-evolving influence graphs, and how to design effective visual metaphors and dynamic representation methods to illustrate influence patterns over time. In this work, we present Eiffel, an integrated visual analytics system that applies triple summarizations on evolutionary influence graphs in the nodal, relational, and temporal dimensions. In numerical experiments, Eiffel summarization results outperformed those of traditional clustering algorithms with respect to the influence-flow-based objective. Moreover, a flow map representation is proposed and adapted to the case of influence graph summarization, which supports two modes of evolutionary visualization (i.e., flip-book and movie) to expedite the analysis of influence graph dynamics. We conducted two controlled user experiments to evaluate our technique on influence graph summarization and visualization respectively. We also showcased the system in the evolutionary influence analysis of two typical scenarios, the citation influence of scientific papers and the social influence of emerging online events. The evaluation results demonstrate the value of Eiffel in the visual analysis of evolutionary influence graphs.

Index Terms—Influence graph, dynamic visualization, citation analysis

1 Introduction

MAKING sense of the evolutionary influence of elements in interconnected information space is a crucial task in many domains. In citation analysis, understanding the development of follow-up topics from a seminal paper helps junior researchers identify cutting-edge opportunities. In social influence analysis, analyzing the dissemination of fake news on Twitter via people's distributed retweeting behavior provides the clue to potentially contain the rumor. Because the analysis questions in these tasks are often unclear to domain users, visualization of the influence hierarchy of key information elements, known as the influence graph, becomes an important tool for users to support their tasks. As an example, Fig. 2a shows a visualization of citation influence graph triggered by a scientific paper.

More often than not, influence graphs grow to very large sizes over time. Some landmark research papers have accumulated more than 10,000 citations. Celebrity gossip tweets on Twitter have been forwarded millions of times. Such large sizes prohibit the use of traditional layout algorithms for influence graph visualization due to their poor scalability [1]. Although clustering and compression methods can be integrated with multiscale visualizations to reveal the community structure of large graphs [2], [3], [4], these methods have been shown to be inappropriate for influence graphs in which flow-based influence propagation patterns are more salient than community structure. Recently, an influence graph summarization method has been proposed which aims to maximize the overall flow rate in a clustered graph representation [5]. However, this latest study is limited to a static influence graph summarization, and does not consider the evolution of influence over time or the potential edge clutter on dense graph summarizations. Also, influence graphs can be filtered to show only the landmark propagators on the graph, e.g., the highly cited papers or the most-retweeted messages. The filtering approach focuses on important details of influence graphs but fails to reveal the overall influence graph hierarchy.

In this study, we consider the problem of visualizing large-scale evolutionary influence graphs. Three domain user's requirements in their influence analysis tasks should be met. First, the visualization should be a compact summary of influence graph while revealing the key nodes, edges, and influence flows on the graph. Second, the

- *Y. Huang is with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100049, China, and also with the Peking University, Beijing 100871, China. E-mail: huangyc96@gmail.com.*
- *L. Shi is with the Beijing Advanced Innovation Center for Big Data and Brain Computing, School of Computer Science and Engineering, Beihang University, Beijing 100083, China. E-mail: leishi@buaa.edu.cn.*
- *Y. Su and D. Wang are with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences and UCAS, Beijing 100049, China. E-mail: {suyue, wdy}@ios.ac.cn.*
- *Y. Hu is with the Yahoo Labs, New York, NY 10036. E-mail: yifanhu@yahoo.com.*
- *H. Tong is with the School of Computing, Informatics, Decision Systems Engineering, Arizona State University, Tempe, AZ 85281. E-mail: hanghang.tong@gmail.com.*
- *C. Wang is with the Department of Computer Science & Engineering, University of Notre Dame, Notre Dame, IN 46556. E-mail: chaoli.wang@nd.edu.*
- *T. Yang are with the Department of Computer Science, Peking University, Beijing 100871, China. E-mail: yang.tong@pku.edu.cn.*
- *S. Liang is with the Academy of Arts & Design, Tsinghua University, Beijing 100084, China. E-mail: liangshuo919@163.com.*

Manuscript received 15 Oct. 2018; revised 17 Feb. 2019; accepted 6 Mar. 2019. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Lei Shi.)
Recommended for acceptance by J. Heer.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2019.2906900

visualization should support an interactive analysis of temporal dynamics of influence graphs, including the evolution of graph structure and certain node/edge groups, and their pace of evolution. Third, the visualization should allow to drill down to the detail of individual elements in the influence graph and link these details to the context of influence such as human factors.

Solving the evolutionary influence graph visualization problem is challenging. In static settings, a specialized matrix decomposition on the influence graph has been shown to approximate the influence flow maximization objective and provide compact node summarizations [5]. Regarding evolutionary influence graphs, it remains an open question whether node summarization alone can effectively reduce the visual complexity of influence graphs. On visualization design, the use of node-link metaphor might be appropriate for our analysis scenario as users can conduct several influence path related tasks (e.g., ST3 in the user study presented in Section 6.2), in which the node-link representation is reported to perform the best [6], [7]. However, the authors of existing works have concluded that for most other graph analysis tasks, node-link representation performs worse than the adjacency matrix on graphs that are large and dense. Again, this calls for the application of effective edge summarization algorithms before influence graph visualization. In addition, the flow map visual metaphor [8] adopted in our design was initially applied to graphs with a single source node and with all the other nodes directly linked to the source. In comparison, the influence graphs studied here have many more hierarchies, which brings challenges to the flow map layout algorithm. Last but not least, the display of time-varying graphs remains an open problem for the visualization community. However, in our case, the single source and mostly single directional nature of the influence graph has narrowed the design space for visualization.

We present Eiffel, an evolutionary flow map for influence graph visualization. Our contributions are summarized as follows:

- We propose new edge summarization algorithms, based on the node summarization method reported in [5], to reduce the visual complexity of evolutionary influence graphs. The temporal summarization method is also introduced to improve analysis efficiency when the number of time frames is large (Section 4). We quantitatively validate the proposed triple summarization framework in both data-driven experiments that compare with standard graph clustering and edge pruning algorithms, and in a user study about the soundness of summarization result (Section 6.1).
- We adapt the flow map metaphor to the visualization of influence graph summarizations. A new flow map layout method is proposed to reveal both hierarchical influence structure and flow-based patterns. Two evolutionary visualization modes (i.e., flip-book and movie) are introduced to illustrate the dynamics of influence graphs over time (Section 5). The flow map and evolutionary visualization design are evaluated in separate, controlled user studies. The result demonstrates the advantage of Eiffel over the baseline design using node-link and single-mode evolutionary visualization (Section 6.2).
- We apply Eiffel to the study of citation influence networks and retweeting influence networks. Case studies on real-world data sets were conducted. The study result shows the usefulness of Eiffel in deriving new and detailed insights from evolutionary influence graphs (Section 6.3). An online Eiffel prototype is deployed, which enables the retrieval and visualization of citation influence evolution within the visualization community (Appendix C, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2019.2906900).

2 Related Work

2.1 Influence Graph Visualization

We discuss the study of influence graph visualization in two application domains: citation network analysis and social influence graph analysis.

Citation Networks, as a subset of bibliometric networks [9], describe the citation relationship among scientific documents (e.g., papers, patents). Analyzing the citation networks has been a regular topic in the visualization community [10]. The CiteSpace II system [11], [12] was built to delineate the concept of research front and intellectual base using node-link style citation network visualization. Each document in the research front is represented by a tree-ring node metaphor, which shows its citation information. The links between documents indicate a co-citation relationship [13], [14], [15], i.e., both have been cited in at least one other document. The historiograph in HistCite [16] also supports the node-link visualization of citation networks. In particular, the citation information flow among scientific documents can be displayed. Maguire et al. extracted and visualized the egocentric citation network of a document to reveal its publication impact [17]. In non-node-link designs, VOSviewer [18] projected citation networks onto 2D space using dimensionality reduction methods; CircleView [19] was proposed to arrange the citation context of a document in a circular layout. There are many other citation network visualization tools, e.g., CitNetExplorer [20], Citeology [21], and the general-purpose network visualization toolkits such as Pajek [22], Gephi [23], Tulip [24], and NodeXL [25].

On citation analysis, Eiffel targets the network of highly influential papers during a long period of time. Each of these papers can influence thousands of other papers directly or indirectly. In such a circumstance, the existing visualization methods can introduce huge visual clutter when the full citation network is displayed [11] or are designed to interpret only a small subgraph of the network [17], [19]. For example, CiteWiz [26] proposed the Growing Polygons technique to visualize the citation influence networks, with focus on a detailed study of the one-hop citation relationship. In comparison, Eiffel computes a compact summary of the evolutionary citation influence graphs to well support the analysis of highly influential papers. In addition, Eiffel visualizes the citation influence graph structure and is not optimized for the display of semantic citation content. This is different from the recent work of CiteRivers [27], which illustrates evolving topics of scientific literature and the detailed content in their references.

Social influence graphs are generally constructed to characterize the influence propagation of social media users and their messages. Cao et al. developed Whisper [28], an elaborate visual sunflower metaphor to illustrate the spatiotemporal information diffusion of real-time topics on Twitter. Whisper focuses on the influence propagation in the geospatial dimension. By contrast, Eiffel is designed to visually display the influence graph structure among users or messages. G+ Ripples [29] supports the native visualization of the information propagation process of public posts on Google+. It combines the node-link metaphor with a circular treemap design to efficiently display the hierarchical sharing structure of a selected hot post. G+ Ripples scales to render a large number of sharing nodes by a space-filling design, which can highlight key users in the sharing or resharing process of the post. As a trade-off, it can only reveal the local information propagation path, but not the global influence graph structure. In comparison, Eiffel can provide an overview of the large-scale influence graph structure by a principled summarization framework. Siming et al. introduced D-Map [30], a novel map metaphor for visualizing the egocentric information diffusion on microblogs. D-Map also summarizes social influence graph structure to reduce visual clutter. Nevertheless, their influence node grouping is based on the social behavior of posting users, which is quite different from our goal of revealing influence flows in a graph summarization. There are many other visualizations designed to interpret retweeting propagation networks [31], [32], [33], [34], [35]. However, few of these designs support the summarization of large influence graphs as Eiffel does.

2.2 Flow Map Visualization

The flow map metaphor is a thematic map design originated in the cartography practice [36]. The design focuses on the display of object movements between areas, mostly on the surface of the earth. For example, human migration and the transportation of goods can be drawn as flow maps. In the GIS textbooks [37], [38], lines and points are generally used in the flow map to represent the direction and magnitude of an object's movements, respectively.

Regarding network data, Guo proposed an integrated flow mapping framework for visualizing large volumes of multivariate flow data extracted from location-to-location networks [39]. In this framework, graph partition and flow clustering methods are introduced to group spatial regions and the flows among these regions. Our work is a special case of the flow mapping method over network data when there is a single source node on the influence network being studied. The radial or distributive flow map [40] is generally used in this case. Therefore, the work by Phan et al. [8] on a distributive flow map layout comes closest to our research. They cluster node positions to generate a hierarchical tree structure, based on which a flow map can be drawn. Compared with Phan et al.'s work, Eiffel takes a directed non-tree graph as input and a backbone tree extraction method is used instead of the hierarchical clustering from node positions.

3 Problem

3.1 Analysis Scenario

In this preliminary work, we restrict the scope to the study of single-source maximal influence graphs, which illustrate the influence of one key element in the information space. Such a maximal influence graph is composed of three types of entities: an *influencer* node acting as the single source of influence, all the *propagator* nodes that are directly or indirectly influenced by the influencer, and the directed timestamped influence links from the influencer to the propagators and between the propagators. For example, in the citation analysis scenario, the maximal influence graph of a scientific paper f is composed of a set of nodes representing papers directly/indirectly citing f (including f), and the reversed citation links among these papers being the influence links. Unlike previous work [5], we consider the temporal dynamics of the influence graph. By the evolutionary setting adopted in this work, each influence link is associated with a unique time when the influence first happens from the source of the link to its target. For example, on the citation influence graph, the time of each link indicates the publication date of the target paper which cites the source paper.

Our analytical goal is to understand the evolutionary influence of the selected element (i.e., the influencer) in the information space. Achieving this goal serves as the centerpiece of many domain user's decision-making tasks, for example, to select the test-of-time paper award for a conference or to identify the key people and time frame to accelerate the spread of useful memes on social media. Because these decisions are often made by the human without rigid quantitative criteria, the effective visualization of evolutionary influence graphs allows users to raise questions, formulate hypotheses, validate and finalize their decisions.

3.2 User Requirement

We summarize three user requirements on the visualization of evolutionary influence graphs.

First, though the influence graph in many scenarios is large and complex, consisting of tens of thousands of elements organized in a non-tree structure, the visualization should be compact with appropriate visual complexity for the analysis of end users. More importantly, it should reveal the key components of the underlying influence graph, including the grouping of graph nodes and links, the critical propagators, and the salient influence flows across the entire graph. Meeting this requirement allows the user to comprehend the overall picture of the influence graph.

Second, as the influence graph is evolutionary, design efforts should be made to display the temporal dynamics of the graph in addition to its static structure. Over the potentially long evolution time span, the visualization should be able to locate the major changes of the graph while permitting the access of the influence links in a particular time frame.

Third, both the influence graph and its evolution forms under certain information context. For example, in citation analysis, each node in the influence graph represents a research paper written by a list of authors on a relevant topic. The same authors can contribute to several other influence nodes/links in the graph, on the same or separate topics. Illustrating the correlation of this context with the

TABLE 1
Notations

SYMBOLS	DEFINITION
$f, G = (V, E)$	influencer and its maximal influence graph
n, v_i, e_{ij}	# of nodes, the ith node, and the directed link from v_i to v_j in G
A, a_{ij}	G's adjacency matrix and its (i, j)th entry
T, t_{ij}, Γ	G's time matrix, (i, j)th entry, and time span
$G[\tau] = (V[\tau], E[\tau])$	evolutionary influence graph G at time τ, $G = G[\Gamma]$
$M[\tau]$, M	static abstraction of $G[\tau]$, $M = M[\Gamma]$
k, π_i, $\lvert\pi_i\rvert$, $C(v_i)$	# of clusters in M, the ith cluster and its size, the clustering function
l, ξ_i, $S(\xi_i)$, $D(\xi_i)$, $r_f(\xi_i)$	# of flows in M, the ith flow, its source and target cluster index, the flow rate
$\tau_1, \ldots, \tau_L$	L-segmentation on $(0, \Gamma]$ for IGS

influence graph can be important for users to understand the evolutionary pattern of the influence.

3.3 Task Characterization

To meet the user requirements, we design the evolutionary influence visualization for the following tasks in the typical scenario of citation analysis.

T1. Static Overview of Influence Graphs. To analyze the influence of a scientific paper (the influencer), users start from a visual summary of all the papers directly or indirectly citing the influencer and the citation structure among them. The summary provides an overview of the scale of the influence and the key component in the citation influence graph, such as the grouping of papers, highly influential papers, and the flow-based influence patterns.

T2. Interactive Analysis of Influence Evolutions. From the static overview, users go further to explore the evolutionary dynamics of the citation influence graph. This includes both a high-level viewing of the influence accumulation or fluctuation over time and an interactive visual query to analyze the fine-grained influence graph in the selected time frames. Visual comparisons over time are also conducted to identify the structural change of the influence graph.

T3. Context Correlation and Detail Viewing. After the static and dynamic analysis, users focus on the detailed contextual information of the citation influence. S/he can query the part of the influence graph contributed by a key author, filter the influence graph by the topic relevancy to the original influencer, or drill down to the topic keywords studied in a particular group of papers. Accessing the context and details helps users validate the hypothesis formed in the overview and dynamic analysis of the influence graph.

4 Evolutionary Influence Graph Summarization

In this section, we describe the analytical process to summarize evolutionary influence graphs for the proposed influence graph visualization method.

4.1 Definitions and Objectives

Table 1 lists the notations used throughout this work. We consider the maximal influence graph $G(f) = (V, E)$, or G for short, of a source node f (influencer). Fig. 1 A.i shows an example of such an influence graph. Let G have n nodes, denoted by $V = \{v_1, \ldots, v_n\}$, where $v_1 = f$ is the source node and all the other nodes are those reachable from f following the influence links in E. The structure of G is defined by its adjacency matrix $A = \{a_{ij}\}_{i,j=1}^{n}$, where $a_{ij} = 1$ indicates an nontrivial influence link denoted by $e_{ij} \in E$ from v_i to v_j and $a_{ij} = 0$ indicates an absence of influence link.

In the time domain, we apply an evolutionary setting on the influence graph that each link e_{ij} of G is associated with a unique timestamp t_{ij}, which forms a time matrix T for the graph G. Each timestamp t_{ij} indicates when the influence first occurs from the source of the link e_{ij} to its target. Let t_{ij} take an integer value in $(0, \Gamma]$ where Γ denotes the maximal time span of the influence graph. Using the above setting, we define the evolutionary influence graph at time τ by $G[\tau] = (V[\tau], E[\tau])$ where $E[\tau] = \{e_{ij} | t_{ij} \leq \tau\}$ indicates that the influence links occurred before τ and $V[\tau] = \{v_i | \exists v_j, \{e_{ij}, e_{ji}\} \cap E[\tau] \neq \emptyset\}$ indicates the corresponding nodes.

The final objective is to summarize the evolution of the influence graph G by computing abstractions for a series of evolutionary influence graphs $\{G[\tau]\}_{\tau \in (0,\Gamma]}$. This is known as the evolutionary influence graph summarization (IGS) problem. At time τ, we denote the abstraction of $G[\tau]$ by $M[\tau]$. $M[\tau]$ is composed of k disjoint and exhaustive node clusters: $\pi_1, \ldots, \pi_k$ with size $|\pi_1|, \ldots, |\pi_k|$, and $l \leq k^2$ flows: $\xi_1, \ldots, \xi_l$, which are link groups between k node clusters (see Section 7 for a discussion on the choice of k). The source and target node cluster indices of a flow ξ_i are denoted by $S(\xi_i)$ and $D(\xi_i)$. Examples of these abstractions are shown in Figs. 1 A.ii and 1A.iii. Computing each abstraction $M[\tau]$ over $G[\tau]$ is equivalent to defining a clustering function $C(v_i)$ that maps the nodes in $G[\tau]$ onto the cluster indices of $[1, k]$.

4.1.1 Offline versus Online Summarization

There are two strategies in setting the clustering function of an evolutionary IGS. Online summarization computes a separate clustering for each $G[\tau]$ of any $\tau \in (0, \Gamma]$. Offline summarization normally applies the same clustering function for all $\tau \in (0, \Gamma]$, by computing an abstraction $M[\Gamma]$ (or M for short) for $G[\Gamma]$ ($G[\Gamma] = G$, the maximal influence graph). In this work, we apply the offline strategy exclusively for three reasons. First, on evolutionary influence graphs, we only count the emergent dynamics of links (nodes) and therefore the clustering nature of each node is unlikely to change after its first appearance. Second, computing the node clustering only at the end of the time span yields better clustering accuracy given that the influence graph information is complete. This is similar to the online versus offline dynamic graph layout trade-off [41]. Third, the computational cost is much lower for a single-batch offline summarization than a Γ-time online summarization. The online approach also has an additional overhead to preserve clustering stability among summarizations.

Specifically, the offline IGS problem can be decomposed into two sub-problems. First, we must compute a static abstraction M ($M[\Gamma]$) of the maximal influence graph G ($G[\Gamma]$). Second, we must compute an L-segmentation $0 < \tau_1 < \cdots < \tau_L = \Gamma$ for the time span of $(0, \Gamma]$ to generate a series of evolutionary summarizations $\{M[\tau_1], \ldots,$

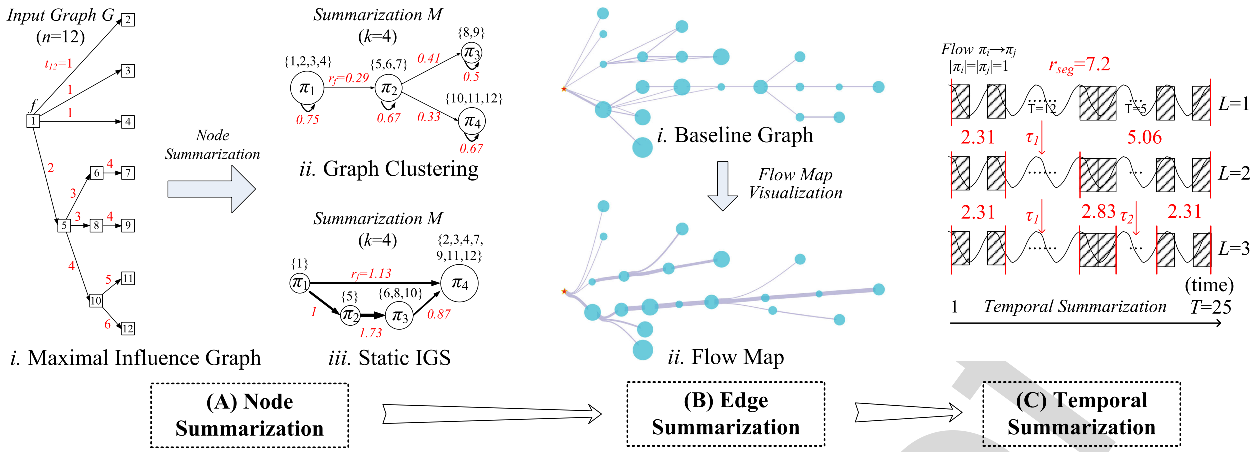


Fig. 1. The evolutionary influence graph summarization framework in Eiffel. (A) The node summarization over i maximal influence graph by ii graph clustering and iii static IGS objective. The link timestamp, clustering result and flow rate are labeled on the influence graphs. (B) The edge summarization from i baseline graph to ii flow map structure. (C) The temporal summarization. Without loss of generality, we illustrate a case summarizing the flow with the minimal rate ($|\pi_i| = |\pi_j| = 1$). Shaded boxes indicate timestamps where the influence links occur.

$M[\tau_L]\}$ for the influence graph. The latter sub-problem is known as the temporal summarization that reduces the number of time frames in the dynamic visualization. In the following, we study the objective for each sub-problem, which paves the way for the Eiffel summarization framework proposed in Section 4.2.

4.1.2 Static IGS Objective

The static IGS objective, which is built on the flow-based heuristic in VEGAS [5], governs the abstraction of M on G. The key is to define the *flow rate* $r_f(\xi)$ for any flow ξ on M as follows:

$$r_f(\xi) = \frac{\sum_{e_{ij} \in \xi} a_{ij}}{\sqrt{|\pi_{S(\xi)}||\pi_{D(\xi)}|}}. \tag{1}$$

This flow rate is exactly the sum of all links on the flow, after normalization by source and target cluster sizes. Given the flow rate, the static IGS objective is formulated as follows:

$$\max \sum_{i=1}^{l} r_f(\xi_i). \tag{2}$$

From the visualization perspective, the static IGS objective maximizes the rate of all influence flows perceivable by users in the summarization. This is essentially the same objective form applied by the classical ratio-association graph clustering algorithm [42], except that ratio-association graph clustering employs a different flow rate definition, i.e.:

$$r_c(\xi) = \begin{cases} r_f(\xi) & \text{if } S(\xi) = D(\xi), \\ 0 & \text{if } S(\xi) \neq D(\xi). \end{cases} \tag{3}$$

Here the intra-cluster flow has the same rate as that of the static IGS objective, whereas the inter-cluster flows are set to zero. By maximizing only intra-cluster flows, the graph clustering method detects communities having dense internal connections, as shown in the summarization result of Fig. 1 A.ii. However, this is undesirable for the summarization of influence graphs. If we take the citation influence graph of Fig. 1 A.i as an example, the sum of the flow rates by the IGS objective (red labels in Fig. 1 A.ii) is much lower than that by the static IGS objective (red labels in Fig. 1 A.iii, 3.62 versus 4.73), whereas the sum of the intra-cluster flow rate is higher.

4.1.3 Temporal Segmentation Objective

The second objective is to regulate the temporal summarizations $M[\tau_1], \ldots, M[\tau_L]$ by choosing L nontrivial segmentation points denoted by $0 < \tau_1 < \cdots < \tau_L = \Gamma$ $(L < \Gamma)$. The heuristic is to divide the timeline into L dense time frames in which intense influence links emerge. The resulting dynamic visualization can reveal the stages of the influence evolution from the influencer. We denote these L time frames by $W_1, \ldots, W_L$, where $W_i = (\tau_{i-1}, \tau_i]$, $\tau_0 = 0$, $\tau_L = \Gamma$. Each time frame is reduced by removing empty timestamps from the starting and ending boundaries of the frame.

Using these time frames, each flow ξ is divided into L continuous flow segments, denoted by $\xi^{(1)}, \ldots, \xi^{(L)}$. The *flow segment rate* of $\xi^{(g)}$ is defined as follows:

$$r_{seg}(\xi^{(g)}) = \frac{\sum_{e_{ij} \in \xi, t_{ij} \in W_g} a_{ij}}{\sqrt{|\pi_{S(\xi)}||\pi_{D(\xi)}|}} \cdot \frac{\sum_{e_{ij} \in \xi, t_{ij} \in W_g} a_{ij}}{|W_g|} \cdot |W_g|^q, \tag{4}$$

where $|W_g|$ denotes the length of the reduced time frame W_g and $q \in (0, 1)$ is the segmentation parameter. Note that the first multiplicative term in Eq. (4) is the exact flow rate definition used in Eq. (1) within the current time frame. This first term will sum to a constant value for all segments in a flow given a fixed static IGS abstraction. The second multiplicative term is a weight that prioritizes high density flow segments. The third term is a penalty for short segments (also an award for long segments) so that it does not end up with all one-length flow segments. We apply $q = 0.5$ by default as a trade-off between segment density and frame size. Finally, the temporal segmentation objective is formulated as follows:

$$\max \sum_{i=1}^{l} \sum_{g=1}^{L} r_{seg}(\xi_i^{(g)}). \tag{5}$$

If we take the minimal flow $\pi_i \rightarrow \pi_j$ ($|\pi_i| = |\pi_j| = 1$) in Fig. 1 C as an example, the initial single-segment flow ($L = 1$)

with time span $\Gamma = 25$ has a flow segment rate of 7.2. After choosing appropriate segmentation points at τ_1 and τ_2, the sum of the segment rates increase to 7.37 ($L = 2$) and 7.45 ($L = 3$), respectively.

4.2 Eiffel Summarization Framework

In Section 3, we built a three-stage framework to summarize large evolutionary influence graphs. In the first stage (Fig. 1 A), the nodes in the maximal influence graph G are clustered to maximize the static IGS objective, which leads to a smaller graph of k nodes (clusters) and a maximum number of k^2 edges (flows). In the second stage (Fig. 1 B), l flows are selected to adapt to the flow map visualization design. Lastly, in the third stage (Fig. 1 C), L flow segments are extracted from the entire timeline to optimize the user viewing experience in the evolutionary influence graph visualization.

4.2.1 Node Summarization

The work in Ref. [5] showed that the static IGS objective can be optimized by a symmetric version of nonnegative matrix factorization (SymNMF) [43]. We follow this method and propose a two-stage approach. First, we compute the topology similarity matrix A^G of the influence graph G as follows:

$$A^G = \frac{AA^T + A^T A}{2}, \tag{6}$$

where A is the adjacency matrix of G. In the context of citation influence graphs, each entry of A^G indicates the similarity between two papers by the number of commonly cited and commonly citing papers (i.e., neighboring nodes in the graph). Second, matrix decomposition is conducted to generate k node clusters from the similarity matrix A^G by SymNMF

$$\min_{H \geq 0} ||A^G - HH^T||_F^2, \tag{7}$$

where $|| \cdot ||_F$ denotes the Frobenius norm of the matrix. $H = \{h_{ij}\}$ is an $n \times k$ matrix that indicates the cluster membership of all the nodes in G. v_i will be clustered into π_c if h_{ic} is the largest entry in the ith row of H. We apply the following iterative SymNMF solver with a multiplicative updating rule [43] to compute H:

$$h_{ij} \leftarrow h_{ij}\left(1 - \beta + \beta\frac{(A^G H)_{ij}}{(HH^T H)_{ij}}\right). \tag{8}$$

Here, h_{ij} denotes the entries of H and β is set to 0.5. The iteration ends when $||A^G - HH^T||_F < \epsilon||A^G||_F$ where $\epsilon = 10^{-7}$, or a maximum number of iterations (500) is reached.

We evaluated the SymNMF-based node summarization method by comparing its performance with those of classical graph clustering methods in a series of numerical experiments. The experimental results in Appendix A, available in the online supplemental material show that the overall flow rate and the content consistency within clusters form a trade-off in IGS. SymNMF obtains the largest overall flow rate among all the algorithms tested on graphs of any size. Therefore, we selected SymNMF as the node summarization method in Eiffel. On large graphs (e.g., more than 1000 nodes), all algorithms applied to a moderate number of clusters (20 or 40) fail to detect consistent node clusterings. This calls for the development of new approaches to maintain the focus of user analyses on smaller influence graphs (Section 7). More details on the evaluation of node summarization methods are presented in Appendix A, available in the online supplemental material.

4.2.2 Edge Summarization

Influence graphs generated after node summarization can be much denser than the original graphs, and they often have complex link structures. To succinctly visualize the flow of information from the influencer to propagators, we propose to further summarize the edges of influence graphs by highlighting the most important link groups, while minimizing information loss. This means that we attempt to achieve two conflicting objectives. First, we want to maximize the overall flow rate in the summarization. Second, we want to reduce visual clutter and minimize edge crossings in the final display (i.e., flow map visualization). Below we propose three edge summarization algorithms.

Connected Top-n Flow Graph. The first edge summarization algorithm we propose uses a greedy approach. All edges (flows) are sorted by the flow rate. The first $n - 1$ edges with the highest flow rates are kept, and the other flows are removed. Here n is the number of nodes in the graph. If the resulting graph is disconnected, we incrementally add back the removed edges in decreasing order of their flow rates until the graph becomes connected. We call the final graph the *Connected Top-n Flow Graph*.

Maximum Weighted Spanning Tree (MWST). The second edge summarization algorithm computes an MWST that is rooted at the source node, which maximizes the overall flow rate of the tree edges. This algorithm guarantees that the resulting graph (a tree) is planar and can be drawn free of edge crossings.

Maximal Padded MWST. Since a tree has just $n - 1$ edges, some edges with high flow rates may be excluded when using the MWST. To preserve more information in the summarization, we propose to selectively add back non-tree edges. While it is a straightforward task to add non-tree edges to the visualization, doing so would introduce considerable visual clutter and distract users from the flow map metaphor. To reduce clutter while preserving the flow map design, we leverage the edge bundling technique and only add back non-tree edges that can be bundled onto the tree structure of MWST. Specifically, for a directed non-tree edge $e = v_i \rightarrow v_j$, if there is a path from v_i to v_j in the spanning tree, we bundle e with that path. If not, but there is a path from v_i to v_j in the current summarization (including the tree edges and non-tree edges added thus far), we bundle e with that path. Otherwise, if no path can be found, this edge is not added. All non-tree edges are tested for add-back in decreasing order of their flow rates. The final visualization largely preserves the flow map design, while maximally maintaining the influence graph information. We call the MWST with bundled edges the *Maximal Padded MWST*.

The proposed edge summarization methods were evaluated in a numerical experiment (Appendix B, available in the online supplemental material). The experimental results showed that while all the three methods could reduce the

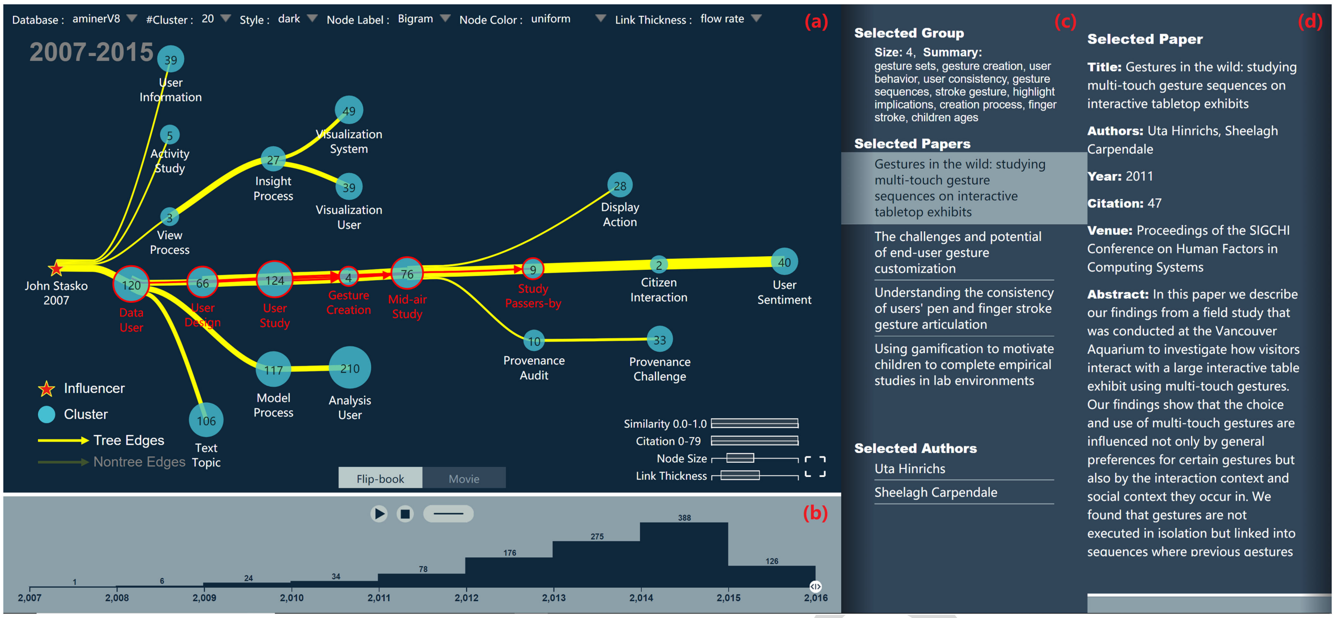


Fig. 2. Eiffel user interface: (a) Flow map for IGS; (b) Animation controller for evolutionary visualization; (c) The selected node group, which represents a list of nodes; (d) Detail panel on the selected node.

visual clutter and minimize edge crossings, the maximal padded MWST preserved a higher overall flow rate for graphs of any size after edge summarization, compared with MWST and connected top-n flow graph. Therefore, we selected maximal padded MWST as the edge summarization method in Eiffel. More details on the evaluation of edge summarization methods are presented in Appendix B, available in the online supplemental material.

4.2.3 Temporal Summarization

After the node and edge summarizations, the temporal summarization computes the best timeline segmentation to maximize the objective in Eq. (5). We propose an iterative optimization process for temporal summarization. If we take the segmentation of a single flow as an example, as shown in Fig. 1 C, the process begins by treating the entire flow as a single segment. In each iteration, the best segmentation point (τ_i) is identified by maximizing the sum of the flow segment rates in Eq. (5). Segmentation ends when all the candidate segmentation points no longer increase the sum of the segment rate. The process of a single flow can be extended to the entire evolutionary influence graph by aggregating all the flow rates onto the same timeline.

We caution that temporal summarization may introduce some side effects. When displayed as an animation, users may not recognize the fluctuating speed of the passage of time. To avoid this effect, the animation buttons in Eiffel are disabled when temporal summarization is applied. We also note that a few enabling conditions are set in Eiffel to apply temporal summarization. First, the total number of time frames should be large (based on a backend setting) so that the summarization in time can improve usability with respect to the side effect. Second, the objective in Eq. (5) should increase from that of the default setting without summarization, which indicates that the influence graph evolution is indeed staged and can be clearly perceived after the temporal summarization.

5 Flow Map Visualization

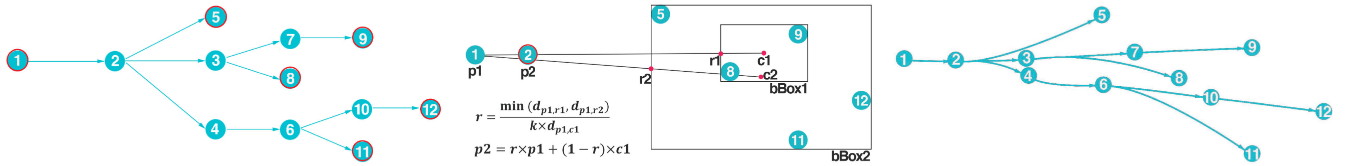
In Eiffel, we apply the flow map design [8] by observing the similarity between the influence and flow graphs (e.g., human migration). First, both types of graphs have roots, which enables the extraction of tree-based backbones. Second, in both cases, the flows among nodes are at least as important as the nodes themselves.

5.1 Static Flow Map

5.1.1 Visual Design

Fig. 2 shows a screenshot of the Eiffel visualization interface. In the main panel (Fig. 2a), the citation influence graph summarized by the maximal padded MWST (Section 4) is visualized as a flow map, which serves as the overview of the graph (T1 of Section 3.3). In the leftmost part of the flow map, the red star icon indicates the source of the influence graph, i.e., the influencer. All the other visual nodes in cyan circles indicate summarized groups of original nodes in the influence graph. The size of each circle encodes the number of nodes in the group following Stevens' power law for area perception [44]. Normalization is also applied to avoid an extreme difference in the actual size. The exact number of nodes in each group is displayed in the center of each circle and can be turned off to focus on the graph structure. The label below each group provides a summary of node content, which is produced by the keyword extraction algorithm described in Section 5.1.2 below.

The links between nodes in the flow map are represented as yellow, segmented Bézier curves, whose layout method we describe later. By default, the thickness of each segment indicates the flow rate from the source of the segment to its destination on the maximal padded MWST, including the flows passing through. This is consistent with the design rationale of the flow map. To reduce visual clutter, the arrow of the link is not visible unless the user hovers the mouse hovered, because the flow is by default from left to right. In addition to the



(a) Initial node layout by `dot` [45] (b) New layout for an intermediate node v_2 (c) The final flow map layout

Fig. 3. Eiffel flow map edge layout process.

backbone MWST in solid, curved lines, other non-tree links can be displayed on demand as half-transparent, straight lines.

In addition to the flow map of the IGS, more information is provided in the corner space of the main panel (Fig. 2a). On the top-left, a label indicates the time range of the influence graph; on the bottom-left, a legend indicates the types of graph nodes and links; on the bottom-right, two double-ranged sliders control the maximal/minimal node size and link thickness respectively to reduce the visual clutter arising from overlapping nodes/labels.

5.1.2 Interaction

The Eiffel interaction on the static flow map is designed to fulfill the context and detail viewing tasks (T3 of Section 3.3). Some of these interactions are accessed via the menu on top of the flow map (Fig. 2a), users can configure the mappings from data to the node label, color, link thickness, and the number of clusters. The node color transparency can be set to reflect the average number of citations of each paper group. This can help in the identification of important topic streams on an influence graph. The system supports different color styles. For a static display, light background and dark foreground colors are used by default. When users switch to evolutionary analysis, a dark background color is applied to provide a movie-like display.

With regard to network interactions, Eiffel offers baseline interaction methods including node drag&drop, zoom&pan, and click selection. When the mouse hovers over a node, the other nodes that have directly influenced this node or have been influenced by this node are highlighted on the graph, as well as the connecting influence flows. This helps to distinguish between direct and indirect influences. Upon the selection of a node on the graph (Fig. 2a), the group information (size, content summary, etc.) and the list of original nodes in the selected group (i.e., a list of papers in the citation case) are displayed in the panel to the right of the flow map, as shown in Fig. 2c. When users select one node from the list, details regarding this node (i.e., paper title, venue, etc.) are shown in the rightmost panel (Fig. 2d). On the citation influence graph, the authors of the selected paper are displayed below the list of papers. When users select one author, the influence of this author can be visually observed by the list of his/her co-authored papers displayed on top of the full influence graph in the main panel (Fig. 8c).

The influence graph can be further analyzed via the filtering operations by the two double-ranged sliders at the bottom-right of the main panel (Fig. 2a). If we take the citation influence graph as an example (Figs. 8a and 8b), using the top similarity filter, users can specify a minimum similarity value for the source of the influence graph, and display the distribution of nodes that match this criterion on the influence graph visualization. Using the citation filter below, the minimal #citations can be specified to show only the important papers on the visualization. In both cases, the full citation influence graph is drawn in the background and the filtered graph is shown in the foreground overlaid on the full graph.

5.1.3 Algorithm

To draw an aesthetic flow map, we designed three algorithms to realize: 1) placement of nodes; 2) intelligent edge layout; 3) node label generation.

Node Placement. The node layout of a flow map in Eiffel is calculated in three steps. First, a backbone tree is extracted by the maximal padded MWST algorithm described in Section 4.2.2. Second, the `dot` algorithm in the GraphViz package [45] is applied to the backbone tree (including the links padded onto the tree) to compute the layout of the root and leaf nodes on the backbone tree. The `dot` algorithm is an implementation of the Sugiyama-style hierarchical graph layout [46]. Third, the position of the intermediate nodes on the backbone tree is computed together with the edge layout process.

Edge Layout. We introduce a new edge layout algorithm in Eiffel, which is based on the work of Phan et al. [8]. The original flow map layout algorithm only works on graphs with one root and several 1-hop neighbors (i.e., star graphs). The main idea of our algorithm is to keep the aesthetic flow map layout while allowing flows to pass through intermediate nodes on the backbone tree.

We describe the algorithm with respect to a simple graph in Fig. 3. The nodes are denoted as $v_1, v_2, \ldots, v_{11}$. In the first step, the positions of the root and leaf nodes are pre-computed by `dot` (nodes outlined in red, Fig. 3a).

In the second step, the position of all intermediate nodes are computed, in the order of a breadth-first tree search. If we take the first node v_2 as an example, as shown in Fig. 3b, we first define the concept of a sub-cluster. A sub-cluster of one node includes all the nodes in one of its child branches on the tree. For example, $\{v_3, v_7, v_8, v_9\}$ is a sub-cluster of node v_2 and $\{v_7, v_9\}$ is a sub-cluster of v_3. To compute the layout of v_2, we first determine its maximal weighted sub-cluster. In this case, the node weight can be the number of papers in the group. Assume $\{v_3, v_7, v_8, v_9\}$ is the maximal weighted sub-cluster of v_2. Then two bounding boxes are considered: one to enclose all the leaf nodes in this sub-cluster (i.e., $\{v_8, v_9\}$), denoted as $bBox_1$ (centered at c_1), and the other to enclose the leaf nodes of all the other sub-clusters of v_2 (i.e., $\{v_5, v_{11}, v_{12}\}$), denoted as $bBox_2$ (centered at c_2). In cases where a node has only one sub-cluster, $bBox_1$ and $bBox_2$ become the same. Lastly, the position of v_2 is computed as follows:

$$p_2 = \frac{\min(d_1, d_2)}{k \cdot d_3} \cdot p_1 + \left(1 - \frac{\min(d_1, d_2)}{k \cdot d_3}\right) p_{c_1}. \tag{9}$$

In Fig. 3b, r_1 and r_2 are two intersection points with $bBox_1$ and $bBox_2$ when connecting the root (v_1) to c_1 and c_2, respectively. d_1, d_2, d_3 are the distances from the root to r_1, r_2, c_1, respectively. p_1, p_2, p_{c_1} are the positions of v_1, v_2, c_1, respectively. k denotes the number of hops from v_2 to its maximal weighted leaf node v_9. By this algorithm, v_2 is placed on the straight line connecting the root to the center of its maximum weighted sub-cluster. After positioning v_2, all the other intermediate nodes are placed by the same method in the order of a breadth-first tree search.

In the third step, to smoothly connect the root to each leaf node, Bézier curves are constructed, which pass through all the intermediate nodes on the backbone tree (Fig. 3c). Note that, in order to differentiate the flow rate of each link, each Bézier curve is first virtually computed and all the control points are kept. Next, each segment on the Bézier curve that connects two neighboring nodes is drawn separately using these control points.

Label Generation. The textual label beneath each node is generated by an improved TF-IDF algorithm. TF-IDF was used previously in information retrieval to rank the words from one document in the context of a text corpus. In the citation influence scenario, we extract keywords from a selected group of papers, which correspond to a single node in the influence graph summarization. Our algorithm is composed of three steps.

First, we denote the selected group of papers as C. The title and abstract of all the papers in C are merged into a single document denoted as c. Separate weights of the title and abstract are used, by default, each title is counted twice. The title and abstract of highly cited papers is also assigned a higher weight.

Second, we extract and rank tokens from c. Both the unigram and bigram schemes are applied. In the unigram, each word is counted as a token; in the bigram, each pair of two consecutive words in the document is counted as a token. The tokens in c are ranked by the metric computed as follows:

$$df\_ranking\_metric(t, c, C, D) = tf(t, c) \cdot idf(t, D) \cdot df(t, C). \tag{10}$$

Here, we denote the token to be ranked as t, the paper collection in the whole data set as D. The first two terms in the right side of Eq. (10) preserve those in the original TF-IDF algorithm, which indicate the token frequency of t in c and the inverse document frequency of t in D. We introduce a third term of $df(t, C)$ that is not used in TF-IDF. This new term represents the document frequency of t in the selected paper group C and is used to encourage the selection of tokens that appear in more papers. In other words, $df(t, C)$ is a coverage metric. For example, when comparing one token with ten occurrences in just one paper of the group and another token with one occurrence in each of all the ten papers in the group, we prefer to select the latter token.

Third, after the top-ranked tokens are selected, we extract keywords from these tokens. Due to the limited viewing space, we pick just one keyword from each token. When the bigram scheme is used, the two words in a bigram token is ranked further by the metric of Eq. (10) computed in the unigram scheme.

Our keyword extraction algorithm takes the user's input for customization. Users can switch between unigram and bigram schemes, and choose to show 1-3 keywords according to the space provided. The node layout and the node/label size can also be fine-tuned for better visualization.

5.2 Evolutionary Visualization

In addition to the static display, Eiffel supports visualization of the evolution of influence over time (T2 of Section 3.3). Depending on the summarization result, users can invoke one of two evolutionary visualization modes. In the flip-book mode (Fig. 7a), the influence graph is visualized cumulatively: once a node or flow has emerged on the timeline, it remains present forever. Users can determine an end time point to display the accumulated influence graph until this point. This is effective for analyzing the propagation of influence over time. In the movie mode (Figs. 7b and 7c), the evolutionary visualization shows only the nodes and flows in a selected time window. This window can be adjusted and scrolled along the timeline to display the temporal dynamics (invariants, changes, etc.) of influence evolution. As shown in the bottom of Fig. 2a, these modes are configured by switching between the two scented tabs located under the flow map.

To illustrate influence evolutions, we designed a smooth animation scheme for the transition between consecutive time frames in both flip-book- and movie-mode visualizations. First, for nodes that have emerged or are growing in a given time frame, silver halos are drawn around these nodes to attract the viewer's attention to these changes (e.g., in Fig. 7a, a halo is associated with node groups having high growths). The node size and numeric label inside each node circle also change with the new group size. Second, the new flows in each time frame do not appear instantly. Instead, an animated transition is displayed so that the influence link stretches gradually from the source to the destination. In the movie mode, three stereo depths are introduced to emphasize the evolution of influence over time. In the foreground, we draw the newly emerged flows and nodes in silver and fill them with halos, and we do the same with the flip-book mode. In the main display layer, other visual objects in the currently selected time window are drawn in the standard design. In the background, the complete influence graph (accumulated up to the last time frame) is displayed in high transparency, which serves as context for the current influence graph.

Previous researches by Robertson et al. [47] showed that animation-based trend visualization is the fastest technique for presentation, but performs worse than static displays (such as small multiples) regarding analysis tasks. Therefore, in our design, we support both animated evolutionary visualization and their static displays. As shown in Fig. 2b, an animation controller is designed beneath the flow map view of the Eiffel interface, which is composed of two parts. In the top row, "play" and "stop" buttons provide the same functionality as those in a classical movie player for animation. In the bottom row, a timeline slider allows flexible navigation to show the static display of influence visualization in a particular time window. In the flip-book mode, there is a single point selector on the timeline with which users can scroll to any interesting time point. In the movie mode, the selector becomes a two-ended range selector, which enables users to adjust the length of the selector and scroll it to any interesting time window. After the selection on the timeline is fixed, users can again view the influence evolution by clicking "play" and "stop" buttons. The button on the right of the top row allows users to apply variable window sizes determined by the temporal summarization. On top of the timeline, there is a line chart, which shows the change of graph size in the number of new nodes per time frame.

6 Evaluation

The Eiffel system consists of two technical components: the IGS and the subsequent flow map visualization. In this section, we evaluate each of these components based on the results of controlled user experiments and then demonstrate the utility of the whole system by its application to case study scenarios in citation and social influence analysis.

6.1 User Experiment on Eiffel Summarization

First, we investigate Eiffel's performance in summarizing influence graphs. In Appendices A and B, we report the quantitative results of the summarization algorithms. Eiffel is shown to achieve a better performance trade-off when the influence graph is no larger than medium in size ($\sim$1000), as compared with alternative summarization algorithms. Here, we report on user understanding of the summarization results by comparing the Eiffel visualization with that of a Google Scholar (GS) like interface implemented in our system. The GS interface displays the raw data used in the summarization. The online websites of GS and the Semantic Scholar [48] are not used for comparison because they are based on publication data sources that are not similar to ours (i.e., AMiner and CiteSeerX). The interface and the data used in this experiment are provided in Appendix D, available in the online supplemental material.

Experiment Design. We recruited 24 graduate students as subjects, most of whom were PhD candidates majoring in computer science who had a good understanding of the citation influence graph used in the experiment. The experiment involved two sessions. The first was a training session in which subjects completed a study task on a small influence graph ($\sim$100 nodes) to ensure that all participants in the test session had a good understanding of the visualization and user task. In the subsequent formal test session, each subject performed the task on two visualizations in turn. To eliminate the learning effect, we selected two influence graphs so that each visualization was applied on a different graph: a large influence graph with 29324 nodes (I) and a medium influence graph one with 1080 nodes (II). The 24 subjects were partitioned into four groups by the sequence of visualization-graph pairs tested, i.e., EI-GII, EII-GI, GI-EII, GII-EI (E=Eiffel, G=Google Scholar, I=Graph I, II=Graph II). Each subject's answer and completion time for each task was recorded in the formal test session. Measurement of the task completion time began after the subject had read the question.

Task. Each subject was asked to analyze the influence evolution of one research paper from the IGS (Eiffel) or influenced paper list (GS). After the analysis, s/he was told to write down the top three topic streams stemming from each studied paper, using two to three keywords in sequence for each topic stream. Note that these keywords can be obtained from both labels beneath each node and the extended list of tokens in the group information panel (Fig. 2c). This task ($AT1$) is designed to evaluate whether the subject correctly understands the summarization result (the overview task in Section 3.3) or the retrieved citation list.

After the subject had completed the task for each visualization, s/he was asked to answer two subjective questions based on a 0-6 Likert scale in which 6 is the best and 0 the worst.

AQ1 (Usability): How much did this visualization help you in completing the tasks?

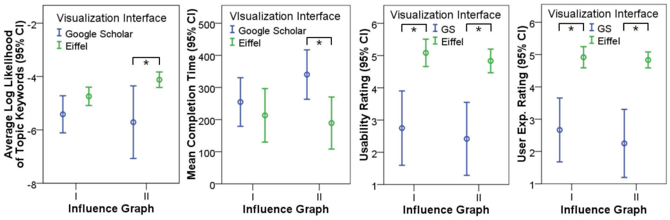
AQ2 (User Experience): How much do you like the experience of using this visualization?

Result and Analysis. We separately analyzed the experimental results of the Eiffel summarization on the two tested influence graphs, as these graph data differ significantly. As such, although it was originally designed as a within-subject experiment, the experiment then had a typical between-subject design in which each subject experienced only a single visualization for a particular graph. We set the significance level to 0.05.

First, we analyzed the user answer from task $AT1$, i.e., the topic keywords. To obtain an objective measure of the accuracy of the subjects' answers, we applied the dynamic topic model (DTM) [49], which extracts multiple evolutionary topics from text corpora with timestamps. In our study, we merged the title and abstract of each paper included in the influence graph into a document, which is used as the input to the DTM. The publication year of the paper is used as the timestamp of the document. The DTM computes a given number of topics and each topic is composed of a list of keywords in each year. Each keyword is also associated with a time-sensitive likelihood for each topic and year it is included in. We fit the topic keywords provided by each subject to the DTM model using a maximum likelihood estimation (MLE) approach. This computes a likelihood value for each topic stream answered by the subject. The average likelihood of all the three topic streams provided by each subject is then used as the measure of the answer accuracy. Note that, we tested 5, 10, 15, 20, 25, and 30 topic numbers by the DTM. Ten topic numbers achieved the highest average likelihood value for all the subject answers, which is used in the analysis of the experimental results in $AT1$.

Fig. 4a shows the distribution of this likelihood measure on a per-keyword, logarithmic scale. Next, we conducted an independent t-test to compare the mean topic keyword log-likelihood of Eiffel and GS. The study result is divided. On influence graph I, we found no significant difference between Eiffel (-4.74 ± 0.34, 95 percent CI) and GS (-5.42 ± 0.7), $t(16.1) = 1.91,\ p = 0.074$, effect size $= 0.43$. On influence graph II, Eiffel achieved a significantly higher log-likelihood (-4.12 ± 0.29) than GS (-5.71 ± 1.36), $t(12.0) = 2.52$, $p = 0.027$, effect size $= 0.59$. Note that in these t-tests, we used the Welch-Satterthwaite method to make an adjustment to the degrees of freedom using because equality of variance does not hold.

With respect to the task completion time, as the assumption of normality does not hold, we applied the Mann-



(a) (b) (c) (d)

Fig. 4. User study results comparing Eiffel with the Google Scholar like interface: (a) Relatedness of user selected topic keywords by their log likelihood in the DTM model; (b) completion time; (c) usability; (d) user experience.

Whitney test to compare the mean completion times of Eiffel and GS. The study result reveals that for influence graph I, there is no significant difference between Eiffel (213.17 $\pm$ 83.34) and GS (254.67 $\pm$ 75.73), $U = 59.0,\ p = .45$, effect size $= 0.15$, with a mean rank of 11.42 for Eiffel and 13.58 for GS (the rank value has a range of 1 to 24). For influence graph II, Eiffel achieved a significantly shorter completion time (189.25 $\pm$ 81.02) than GS (340.17 $\pm$ 76.98), $U = 22.0,\ p = .004$, effect size $= 0.59$, with a mean rank of 8.33 for Eiffel and 16.67 for GS. The completion time distributions are shown in Fig. 4b.

The subjective ratings are summarized in Figs. 4c and 4d. Again, the normality assumption does not hold for the subjective ratings, and we applied the Mann-Whitney test to compare Eiffel and GS. On all rating types and studied influence graphs, Eiffel achieved significantly higher scores than GS. With respect to usability, $U = 15.0,\ p = .001$ on graph I, and $U = 17.0,\ p = .001$ on graph II. For user experience, $U = 14.0,\ p < .001$ on both graphs.

Based on the experimental results, we can report two findings. First, in some cases (influence graph II), the Eiffel summarization helps users to understand the content and evolution of research topics, as compared with searching in raw data. The user accuracies, in terms of the likelihood in the DTM model, and their completion times, are generally better with Eiffel than GS, which shows only raw data. On influence graph I, we observed no significant difference. We hypothesize that this is due to the same reason with the result of Appendices A and B, available in the online supplemental material. The content summary by Eiffel is more consistent in small and medium graphs than in large graphs. Nevertheless, user experiments on more influence graphs are necessary to validate this hypothesis. Second, in the subjective ratings (usability and user experience), Eiffel performed better than GS regardless of the size of influence graphs. Users found Eiffel to be more effective in helping them complete the designed task and would prefer to use Eiffel than GS, although they did not realize that the two interfaces perform similarly in task accuracy given some large influence graphs.

Threats to Validity. First, the experiment result could be further validated by conducting tests on more influence graphs, albeit with the extra cost of hiring additional subjects. We observed user fatigue after they completed the test with two graphs as the study task requires considerable cognitive efforts. Second, the analysis of the accuracy result relies on the DTM model and could be improved with the use of more advanced models. Third, the subjective rating could be affected by social expectation that prefers visualization with an attractive appearance than a list-based display.

6.2 User Experiment on Eiffel Visualization

In the following, we report the results of the user experiment we conducted to evaluate the performance of the Eiffel visualization. The experiment consisted of two formal test sessions, in which the participants completed analysis tasks based on visualizations of static and dynamic influence graphs, respectively. In the static session, we compared two visualizations: a baseline approach using a straight-line node-link graph drawing with a Sugiyama-style layout (GraphVis, as shown in Fig. 1 B.i) and the Eiffel visualization (Fig. 1 B.ii). In the dynamic session, all tests were conducted using Eiffel visualizations and we compared two evolutionary visualization modes: the flip-book and movie modes. In all the approaches compared by the users, the node/edge visual settings were the same.

Experiment Design. We invited the same set of 24 subjects described in Section 6.1. In each test session, the experiment featured a within-subject design in which every subject completed analysis tasks by the two visualizations in turn. Each visualization displayed a different influence graph to eliminate any learning effect. The two influence graphs used were of similar sizes in both the original graph and their summarizations so that the focus of the evaluation remained on the visualization method. The other aspects of the experimental design followed those described in Section 6.1.

Task. Six tasks were presented, three for the static graph analysis session ($ST1$-$ST3$, corresponding to the overview task in Section 3.3) and the other three for the dynamic graph analysis session ($DT1$-$DT3$, corresponding to the evolution analysis task in Section 3.3). All tasks were conducted on medium-sized citation influence graphs similar to those described in Section 6.3.1. For each task, four choices were provided.

ST1 (Static graph structure): Determine which paper cluster directly influences the highest number of other paper clusters.

ST2 (Static graph in-flows): Determine which paper cluster received the highest number of direct citation influences (i.e., citations of other papers.

ST3 (Static graph in/out-flows): Determine which paper cluster generated the highest number of net citation influences (i.e., citation influence sent − citation influence received).

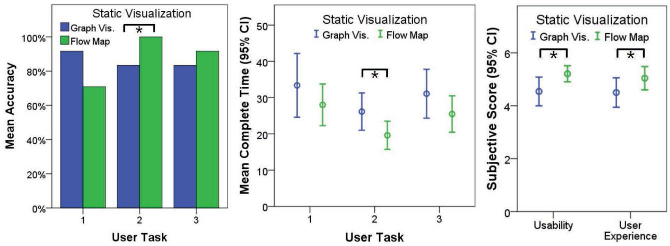
DT1 (Local dynamic graph structure): Given one paper cluster, determine which year the number of papers in this cluster increased the most.

DT2 (Global dynamic graph structure): In a given time range, determine which paper cluster increased by the highest number of papers.

DT3 (Local dynamic graph in/out-flows): Given one paper cluster, determine which year this cluster generated the highest number of net citation influences.

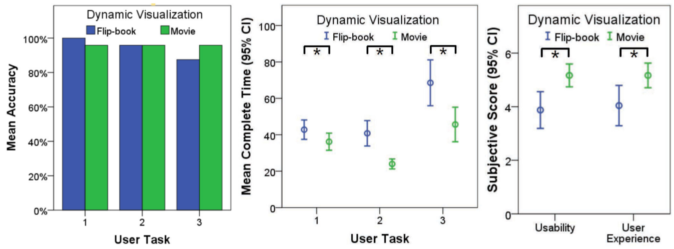
After the subjects completed all the tasks for each visualization, they responded to the subjective questions described in Section 6.1.

Results and Analysis. Static session. Figs. 5a, 5b, and 5c show summaries of the task accuracies, completion times, and subjective scores, respectively, for tasks $ST1$-$ST3$. With respect to task accuracy, the results are split. On average, GraphVis achieved a higher task accuracy than Eiffel on $ST1$ ($ST1$: 0.92 versus 0.71), but was less accurate on $ST2$ and $ST3$ ($ST2$: 0.83 versus 1, $ST3$: 0.83 versus 0.92). Based on the results of an exact McNemar's test, the differences in



(a) Task Accuracy (b) Completion time (c) Subjective

Fig. 5. User performance in static graph tasks.



(a) Task Accuracy (b) Completion time (c) Subjective

Fig. 6. User performance in dynamic graph tasks.

task accuracy were statistically significant on $ST2$, $p = .05$ (1-tailed Exact Sig.), but not on $ST1$ ($p = .063$) and $ST3$ ($p = .34$). With respect to task completion time, on average, GraphVis took longer than Eiffel for subjects to complete tasks ($ST1$: 33.38s versus 28s, $ST2$: 26.15s versus 19.6s, $ST3$: 31.07s versus 25.48s). The difference is significant, as determined by a paired t-test on $ST2$ ($t(23) = 2.21$, $p = .037$, effect size $= 0.45$). We found no significant difference for $ST1$ ($p = .46$) and $ST3$ ($p = .16$). For the subjective rating scores, in both measures, the ratings for Eiffel (usability: 5.21, user experience: 5.04) were significantly better than those for GraphVis (usability: 4.54, user experience: 4.5), as determined by the Wilcoxon test. For usability, $Z = -2.1$, $p = 0.036$, and for user experience, $Z = -1.95$, $p = 0.05$.

From the verbal feedbacks of users, we can draw two conclusions to interpret these results. First, Eiffel visualization outperforms GraphVis in its display of static influence flow patterns (i.e., significantly better accuracy and completion time in $ST2$). This is achieved using a flow map design that emphasizes flow rate quantity. Second, Eiffel facilitates the analysis of static influence graphs in a more user-friendly manner (subjective scores). Users reported that Eiffel was less complex and more visually pleasing.

Dynamic Session. Figs. 6a, 6b, and 6c show summaries of the results for $DT1$-$DT3$. With respect to task accuracy, the flip-book and movie modes achieved a similar average accuracies for all tasks with no significant difference, as determined by the McNemar's test ($DT1$: 0.96 versus 1, $DT2$: 0.96 versus 0.96, $DT3$: 0.87 versus 0.96). With respect to task completion time, the flip-book mode required significantly longer time to complete than the movie mode on all three tasks (on average, $DT1$: 42.8s versus 36.19s, $DT2$: 40.81s versus 23.97s, $DT3$: 68.53s versus 45.63s). The differences are significant, as determined by the paired t-test: for $DT1$, $t(23) = 2.22$, $p = .037$, effect size $= 0.45$; for $DT2$, $t(23) = 5.36$, $p < .001$, effect size $= 1.09$; and for $DT3$, $t(23) = 3.59$, $p = .002$, effect size $= 0.73$. For the subjective scores, in both measures, the ratings for the movie mode (usability: 5.17, user experience: 5.17) were significantly better than those for the flip-book mode (usability: 3.88, user experience: 4.04) by a Wilcoxon test. For usability, $Z = -3.67, p < 0.001$; for user experience, $Z = -3.1, p = 0.002$.

The results and the user feedback from the dynamic session indicate that: 1) On all the tested dynamic graph tasks such as the identification of changes in the node/edge size in the graph, both visualization modes can help users complete tasks correctly (especially the movie mode, with an accuracy of at least 0.96); 2). The subjects found the movie mode to be more efficient (required significantly shorter task time) and user-friendly (better subjective ratings), because this mode allows them to configure a static change view for any selected time range, whereas users must manually compare two flip-book views to perform the same task.

Threats to Validity. First, the results were statistically significant only on a few tasks with respect to accuracy and completion time. This could be due to the relatively small sample size. Second, there may have been the same social expectation bias as that described in Section 6.1.

6.3 Case Studies

6.3.1 Citation Influence Graph

We applied Eiffel to academic citation influence graphs from the AMiner V8 [51] and CiteSeerX [52] data sets. The AMiner data set contains 2.38 million papers on computer science topics up to early 2016, and there are 10.48 million citation links among these papers. Each paper's record includes its title, abstract, authors, and date of publication, etc. From the AMiner data set, we extracted a citation influence graph of 18010 papers from 37 visualization-related venues.[1] We obtained these influence graphs by recursively traversing the influence links (i.e., reversed citation links) from the initial papers. We restricted the influence graph to papers within the visualization domain by early pruning of irrelevant branches: the papers influenced outside the 37 VIS venues were included in the graph but were not traversed thereafter.

In the case studies, we first looked at a paper regarding the Jigsaw visual analytics system in the VAST'07 proceedings [50], for which Fig. 2a shows the initial Eiffel view ($k = 20$). There are five directly influenced paper groups (from top to bottom): 39 papers on user interactions, notably the Apolo CHI'11 work that combines user interaction and machine learning [53]; 5 papers on document entity analysis, including an extension of the Jigsaw paper in next year's IV journal; three seminal works on the reasoning and navigation of visualization; 120 papers on data and streams; and 106 papers on visual text analytics. By analyzing the multi-hop influence, i.e., the evolution of related research topics, we can identify two backbone topic streams. The first stems from three analytical reasoning studies (labels: "view", "process", more details available by drilling down to bigram summarizations) that seek insight into provenance and reasoning processes, and finally split into two branches: the visualization system (e.g., use of eye gaze data), and the user evaluation of the visualization and the analytical

1. We only looked at papers with more than five direct citations.

(a) 2007-2010 (b) 2011-2012 (c) 2013-2015

Fig. 7. Citation influence graph of Jigsaw paper published in VAST’07 [50].

process. The second backbone topic was triggered by the 120 paper cluster on the data stream and user interface. In addition to the side branches of the visual text analytics (also a directly influenced cluster) and 210 miscellaneous papers, the main stream propagated through the study of user interfaces (two clusters with 66 and 124 papers) and finally to human-computer interaction (HCI) research (gestures, citizen science, field studies, etc.). By examining the influence graph structure, we also identified two outstanding paper clusters. A cluster of three papers (labels: “view”, “process”) including the analytical reasoning paper by Shrinivasan and Wijk appear to be the most influential. This small cluster receives little incoming influence but generates a large influence flow. Another noticeable cluster is that of four papers (labels:“gesture”, “creation”), as indicated by the mouse hovering in Fig. 2a. This cluster serves as a gateway between visualization research (left) and HCI research (right), with large flows passing through the cluster.

We further analyzed the dynamics of the Jigsaw paper’s influence by Eiffel evolutionary visualization. In a flip-book mode, we displayed in animations the process of how influence propagates, and captured the overall dynamic picture, although it is still difficult to detect and memorize detailed dynamic influence patterns. In another movie mode, by incorporating the temporal summarization result, the evolution of Jigsaw’s influence is divided into three time periods and displayed in more succinctly: *i*) 2007-2010, when some initial papers on visual text analytics and user navigation process cited the Jigsaw paper (Fig. 7a); *ii*) 2011-2012, when more indirect influences occurred, but the focus continued to be on text analytics and summarization, as well as the user analysis process and performance (bottom-left and top-right large paper groups in Fig. 7b); *iii*) recently, 2013-2015, the influenced topic became more diversified (Fig. 7c). One emerging topic is “display”. When we selected the major paper group on that topic (the node in the center of Fig. 7c on “study”) and examined their details, we found that most papers had reported studies of an HCI topic called “public display”.

The influence of the Jigsaw paper can also be analyzed with respect to the actual topics, their importance, and the associated key authors. In Fig. 8a, we filtered the influence graph to show only papers with high similarity (>0.7) to the topic of the original Jigsaw paper. The similarity score between any two papers is derived from the word mover distance [54] on the vector representation of their title +abstract. Each vector adopts a distributed representation of words using Word2Vec [55]. By examining the result in Fig. 8a, we found that two initial branches on the graph have a larger ratio displayed in the foreground (i.e., 3/5 and 68/106), which indicates that the follow-up research on document entity analysis and the visual text analytics are related more to the original Jigsaw paper. Meanwhile, research on reasoning/navigation (0/3) and their follow-up papers are less relevant (7/27, 18/49, and 20/39). We further filtered the influence graph to show the papers with at

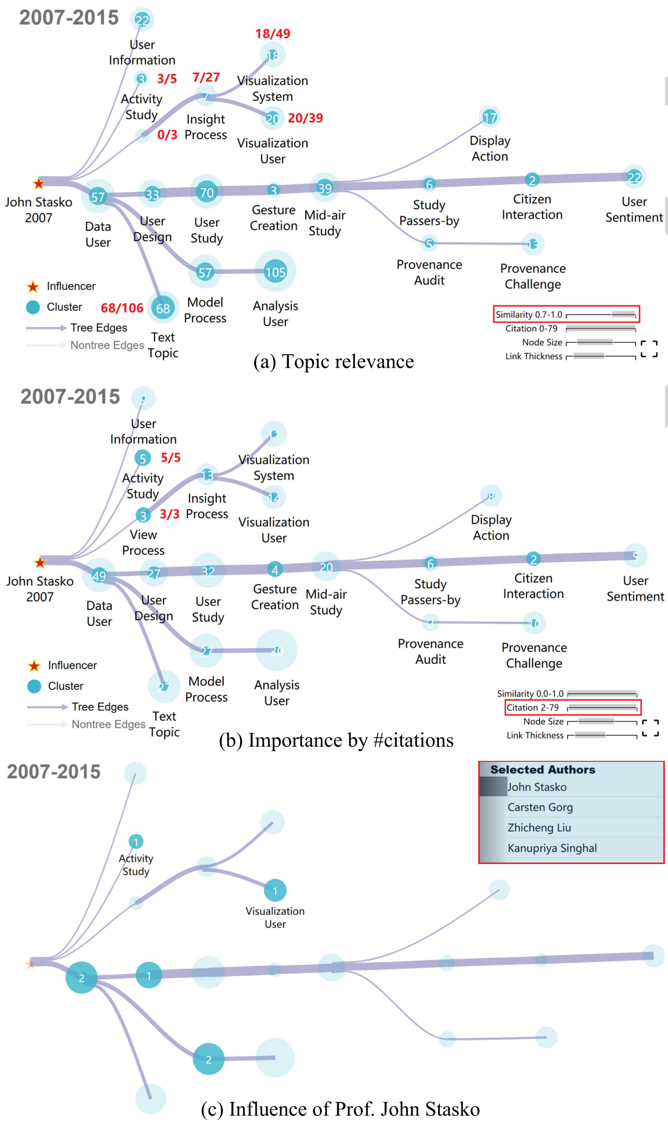


Fig. 8. Multifaceted analysis of the influence of Jigsaw paper.

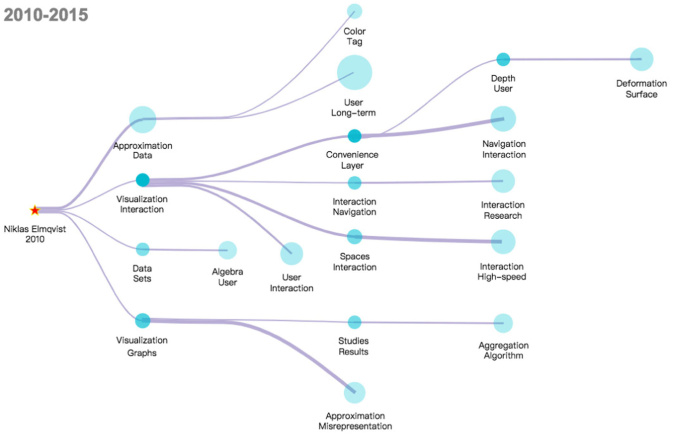


Fig. 9. Influence graph of the hierarchical aggregation survey paper in TVCG [56].

least two citations in the database. As shown in Fig. 8b, the papers on document entity analysis (5/5) and reasoning/navigation (3/3) are more influential than the other topics in the same graph. We also studied the influence of Prof. John Stasko, the leading author of the Jigsaw paper, on this research topic by displaying the papers he has co-authored on the graph. As shown in Fig. 8c, after authoring the Jigsaw paper, he published seven more papers with citation linkages to the original paper, which cover most of the branches in the influence graph. On citation influence analysis, we also include expert feedback in Appendix F, available in the online supplemental material to evaluate the usefulness of Eiffel outside the visualization community.

In another trial, we studied the influence of a survey paper on hierarchical aggregation for information visualization [56], as shown in Fig. 9. By configuring the node transparency to reflect the average number of citations/influence, we identified two classes out of four directly influenced paper clusters. The first class is the cluster with 19 papers in the top, which is large in size but has little average influence. Drilling down to the content of this cluster reveals a diversified summary ranging from network data to approximation algorithms. The follow up large cluster with 40 papers is similar in its mixed content and low level of influence. This research thread may not be the major core field influenced by the source paper.

In the lower area of the figure, there are three small but highly influential paper clusters directly connected to the source. The top cluster, i.e., a single paper studying tangible views for visualization, appears to have the largest influence. Its follow-up four branches continue to address different types of tangible interactions, including bending interaction, tabletop interaction, mobile interaction, and augmented-reality interactions, etc. The small cluster in the middle area is a paper on real-time visual queries of big data, and its follow-up works are mostly related to visual queries. The last smaller cluster at the bottom of the figure contains two papers, with surprisingly similar titles on TreeMatrix visualization. We double-checked the data set and found these papers to be duplicate entries (we have made significant efforts to reduce duplication, but may not have eliminated all of them). This provides side evidence of the correctness of the summarization result: papers with the same citation relationship are put into the same cluster. The TreeMatrix paper has a few direct influences, but only one about aggregation algorithms has further influenced other papers.

6.3.2 Social Influence Graph

In another case study, we applied Eiffel to a large-scale social influence graph on Twitter, which describes the spread of rumors and announcements regarding the discovery of the Higgs boson [57], [58].

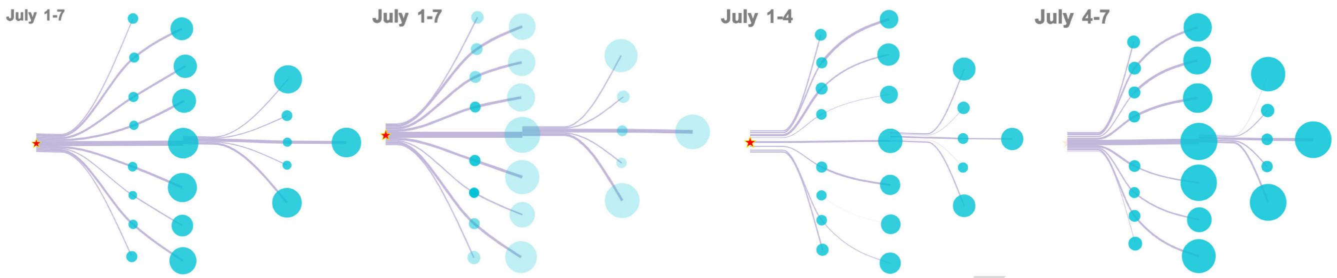
We constructed the original influence graph by aggregating posts by the same users into nodes and folding the links among posts into influence links among users. An artificial node is inserted into the graph as the influencer. Influence links are added from the influencer to each source user who posted related original tweets during this time. The influence graph was summarized using the Eiffel summarization framework, whose flow rate maximization approach fits well the objective of detecting salient influence diffusion patterns. Meanwhile, the backbone tree extracted by the edge summarization accounts for more than 85.5 percent of the overall flow rate after the node summarization. To reveal the user's characteristics, the node color transparency is used to represent the average #followers of users in the same cluster. Twitter's policy forbids the display of further details regarding the identity of users.

Fig. 10a shows the overall structure of the social influence graph ($k = 20$), which is composed of two subgraphs: $i)$ The left area features a two-stage propagation pattern in which the posts of a small portion of users (opinion leaders) were retweeted by a large number of other users (ordinary people). This pattern is validated by Fig. 10b, which shows the average #followers of users by the transparency of their node color. Opinion leaders generally have a higher average #followers, whereas ordinary people have fewer followers. $ii)$ The right area shows the interactions between large groups of people, i.e., the discussions held in small circles of ordinary people.

When we switch to analyze the influence graph in the movie mode, we can compare the influence propagation patterns in two time periods: $i)$ from July 1st to July 4th before/upon the announcement of the new particle, during which a rumor was spread on Twitter (Fig. 10c); $ii)$ from July 4th to July 7th upon/after the announcement when more discussions were posted by Twitter users (Fig. 10d). If we compare these two graphs, we see little difference in their propagation paths, i.e., rumors and news spread on Twitter via similar information channels from opinion leaders to the masses, and later on among the masses themselves. One interesting finding is that whereas the graph size in the second stage is almost three times larger than that of the first stage, the number of opinion leaders remains stable (<50 percent growth) as more discussions arise regarding the confirmed news.

7 Discussion

In evolutionary IGS, the number of clusters (k) is fixed. This is mainly because as k increases, the maximal IGS objective achieved (i.e., the overall flow rate) also increases [5]. There may not be an optimal k under the IGS summarization framework. In Eiffel, we compute IGS summarizations with multiple ks (e.g., 10, 20, 40) and allow users to switch between visualizations of different granularities based on their analysis goals (e.g., overview or details). The limitations



(a) Overview (b) Avg. of #Follower (c) July 1st-4th (d) July 4th-7th

Fig. 10. Eiffel visualization on Higgs social influence graph.

of the current design are two-fold: for an overview, the labels selected for each cluster can be too general to interpret, and there is no way to drill down to the detailed summarization of each cluster for further analysis. For a detailed view, the number of clusters can be so large that the visualization becomes very cluttered. In future work, to overcome these limitations, we plan to develop hierarchical summarizations of influence graphs, in which the visualizations can be fully customized to display both an overview and the details of any particular cluster.

Regarding the application of citation influence analysis in Eiffel, we currently obtain the maximal influence graph of one source paper by an exhaustive search along its reversed citation links. This primitive approach can lead to a very large initial graph in which many nodes (papers) are unrelated to the topic of the source paper. Although venue-based filtering can restrict the graph to pertinent research communities, it cannot generate topic-based influence graphs. As the next step in the Eiffel system, we plan to study the semantics of citation links between papers and the computation of fine-grained topic-based influence graphs.

We showcased this work with the citation influence analysis as the main application. The same technique can be also used in a wide range of other scenarios, including the social influence analysis mentioned in Section 6.3.2, the functional influence analysis of a suspicious line of code in the execution of a program, etc. In these applications, users should first determine the level of basic elements as the node of the influence graph. We choose the scientific papers in the citation case and the posting authors in the social case because they are considered the fundamental unit that generates the influence. When multiple sources of influence exist, special treatments should be placed before using our technique. In the social case, we introduce an artificial influence node that triggers all the sources of influence. Finally, the selected granularity of time could also be important for the success of evolutionary influence graph visualization. For influence process that develops at a moderate pace, e.g., the citation influence, we adopt the granularity of a year or a month. For other processes that evolve rapidly, e.g., social influence on Twitter, we can choose a finer granularity of a day or even an hour.

8 Conclusion

In this paper, we presented Eiffel, a system that draws dynamic influence graphs with evolutionary flow map visualizations. Eiffel addresses multiple challenges when summarizing structurally complex and time-varying influence graphs, which are formulated as evolutionary influence graph summarization problems. To solve these problems, we proposed scalable matrix decomposition, flow selection, and temporal segmentation algorithms to summarize the influence graph in nodal, relational, and temporal dimensions. The flow map of an influence graph summarization is designed to highlight the dominant flow patterns with minimal visual clutter while maximizing the information efficiency of the influence flows. The results of two case studies, which address academic citation influence graphs and Twitter social influence graphs, demonstrate the usefulness of the Eiffel system. We conducted a controlled user experiment to compare Eiffel visualization design with baseline static graph visualization. The results confirm the effectiveness of the use of flow map in evolutionary influence graph analysis tasks.

Acknowledgments

This work was supported by NSFC Grants 61772504, 61672061, U.S. NSF Grant IIS-1456763, IIS-1455886, IIS-1651203, IIS-1715385, and DUE-1833129.

References

[1] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*. London, U. K.: Pearson, 1998.
[2] J. Abello, F. van Ham, and N. Krishnan, "ASK-GraphView: A large scale graph visualization system," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 669–676, Sep.-Oct. 2006.
[3] A. Quigley and P. Eades, "FADE: Graph drawing, clustering and visual abstraction," in *Proc. Int. Symp. Graph Drawing*, 2000, pp. 197–210.
[4] C. Dunne and B. Shneiderman, "Motif simplification: Improving network visualization readability with fan and parallel glyphs," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2013, pp. 3247–3256.
[5] L. Shi, H. Tong, J. Tang, and C. Lin, "VEGAS: Visual influEnce GrAph summarization on citation networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3417–3431, Dec. 2015.
[6] M. Ghoniem, J.-D. Fekete, and P. Castagliola, "On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis," *Inf. Vis.*, vol. 4, no. 2, pp. 114–135, 2005.
[7] R. Keller, C. M. Eckert, and P. J. Clarkson, "Matrices or node-link diagrams: Which visual representation is better for visualising connectivity models?" *Inf. Vis.*, vol. 5, no. 1, pp. 62–76, 2006.
[8] D. Phan, L. Xiao, R. Yeh, P. Hanrahan, and T. Winograd, "Flow map layout," in *Proc. IEEE Symp. Inf. Vis.*, 2005, pp. 219–224.
[9] N. J. van Eck and L. Waltman, "Visualizing bibliometric networks," in *Measuring Scholarly Impact*. Berlin, Germany: Springer, 2014, pp. 285–320.
[10] M. J. Cobo, A. G. López-Herrera, E. Herrera-Viedma, and F. Herrera, "Science mapping software tools: Review, analysis, and cooperative study among tools," *J. Assoc. Inf. Sci. Technol.*, vol. 62, no. 7, pp. 1382–1402, 2011.

[11] C. Chen, "CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature," *J. American Society Inf. Sc. Technol.*, vol. 57, no. 3, pp. 359–377, 2006.

[12] C. Chen, "Searching for intellectual turning points: Progressive knowledge domain visualization," *Proc. National Academy Sci. United States America*, vol. 101, no. suppl 1, pp. 5303–5310, 2004.

[13] H. Small, "Co-citation in the scientific literature: A new measure of the relationship between two documents," *J. American Society Inf. Sc.*, vol. 24, no. 4, pp. 265–269, 1973.

[14] I. V. Marshakova, "System of document connections based on references," *Nauchno-Tekhnicheskaya Informatsiya Seriya 2-Informatsionnye Protsessy I Sistemy*, no. 6, pp. 3–8, 1973.

[15] S. Noel, C.-H. H. Chu, and V. Raghavan, "Co-citation count versus correlation for influence network visualization," *Inf. Vis.*, vol. 2, no. 3, pp. 160–170, 2003.

[16] E. Garfield, S. Paris, and W. G. Stock, "Histcite: A software tool for informetric analysis of citation linkage," *Inf. Wissenschaft und Praxis*, vol. 57, no. 8, 2006, Art. no. 391.

[17] E. Maguire, J. M. Montull, and G. Louppe, "Visualization of publication impact," in *Proc. Eurographics IEEE Co-Supported Int. Vis. Symp.*, 2016, pp. 103–107.

[18] N. J. Van Eck and L. Waltman, "Software survey: Vosviewer, a computer program for bibliometric mapping," *Scientometrics*, vol. 84, no. 2, pp. 523–538, 2010.

[19] P. Bergström and E. J. Whitehead Jr, "Circleview: Scalable visualization and navigation of citation networks," in *Proc. Symp. Interactive Visual Inf. Collections Activity*, 2006.

[20] N. J. van Eck and L. Waltman, "Citnetexplorer: A new software tool for analyzing and visualizing citation networks," *J. Informetrics*, vol. 8, no. 4, pp. 802–823, 2014.

[21] J. Matejka, T. Grossman, and G. Fitzmaurice, "Citeology: Visualizing paper genealogy," in *Proc. Extended Abstracts Human Factors Comput. Syst.*, 2012, pp. 181–190.

[22] W. De Nooy, A. Mrvar, and V. Batagelj, *Exploratory Social Network Analysis with Pajek*, vol. 27, Cambridge, U.K.: Cambridge Univ. Press, 2011.

[23] M. Bastian, S. Heymann, M. Jacomy, et al., "Gephi: An open source software for exploring and manipulating networks," in *Proc. Int. AAAI Conf. Weblogs Social Media*, 2009, vol. 8, pp. 361–362.

[24] D. Auber, "Tulipa huge graph visualization framework," in *Graph Drawing Software*. Berlin, Germany: Springer, 2004, pp. 105–126.

[25] D. Hansen, B. Shneiderman, and M. A. Smith, *Analyzing Social Media Networks with NodeXL: Insights from a Connected World*. San Mateo, CA, USA: Morgan Kaufmann, 2010.

[26] N. Elmqvist and P. Tsigas, "Citewiz: A tool for the visualization of scientific citation networks," *Inf. Vis.*, vol. 6, no. 3, pp. 215–232, 2007.

[27] F. Heimerl, Q. Han, S. Koch, and T. Ertl, "Citerivers: Visual analytics of citation patterns," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 190–199, Jan. 2016.

[28] N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu, "Whisper: Tracing the spatiotemporal process of information diffusion in real time," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2649–2658, Dec. 2012.

[29] F. Viégas, M. Wattenberg, J. Hebert, G. Borggaard, A. Cichowlas, J. Feinberg, J. Orwant, and C. Wren, "Google+ ripples: A native visualization of information flow," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1389–1398.

[30] S. Chen, S. Chen, Z. Wang, J. Liang, X. Yuan, N. Cao, and Y. Wu, "D-map: Visual analysis of ego-centric information diffusion patterns in social media," in *Proc. IEEE Conf. Visual Analytics Sci. Technol.*, 2016, pp. 41–50.

[31] X. Huang, C. Quan, S. Liu, and Y. Man, "Visualization and pattern discovery of social interactions and repost propagation in sina weibo," in *Proc. Int. Joint Conf. Neural Netw.*, 2014, pp. 1401–1408.

[32] C.-T. Ho, C.-T. Li, and S.-D. Lin, "Modeling and visualizing information propagation in a micro-blogging platform," in *Proc. Int. Conf. Advances Social Netw. Anal. Mining*, 2011, pp. 328–335.

[33] D. Ren, X. Zhang, Z. Wang, J. Li, and X. Yuan, "Weiboevents: A crowd sourcing weibo visual analytic system," in *Proc. IEEE Pacific Vis. Symp.*, 2014, pp. 330–334.

[34] P. Ye, C. Wang, Y. Liu, Q. Zhu, and K. Zhang, "Visual analysis of micro-blog retweeting using an information diffusion function," *J. Vis.*, vol. 19, no. 4, pp. 823–838, 2016.

[35] Q. Li, H. Qu, L. Chen, R. Wang, J. Yong, and D. Si, "Visual analysis of retweeting propagation network in a microblogging platform," in *Proc. 6th Int. Symp. Visual Inf. Commun. Interaction*, 2013, pp. 44–53.

[36] B. D. Dent, *Cartography - Thematic Map Design*. New York, NY, USA: WCB McGraw-Hill, 1999.

[37] A. M. MacEachren, *How Maps Work: Representation, Visualization, and Design*. New York, NY, USA: Guilford Press, 1995.

[38] K.-T. Chang, *Introduction to Geographic Information Systems*. New York, NY, USA: McGraw-Hill, 2006.

[39] D. Guo, "Flow mapping and multivariate visualization of large spatial interaction data," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1041–1048, Nov.-Dec. 2009.

[40] "Overview of flow mapping," [Online]. Available: https://www.gislounge.com/overview-flow-mapping/

[41] F. Beck, M. Burch, S. Diehl, and D. Weiskopf, "The state of the art in visualizing dynamic graphs," in *Proc. Eur. Conf. Vis.*, 2014, p. 11.

[42] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE TPAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[43] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. SIAM Int. Conf. Data Mining*, 2005, pp. 606–610.

[44] "Stevens' power law," [Online]. Available: https://en.wikipedia.org/wiki/Stevens%27powerlaw

[45] E. R. Gansner and S. North, "An open graph visualization system and its applications to software engineering," *Softw. - Practice Experience*, vol. 30, pp. 1203–1233, 2000.

[46] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for visual understanding of hierarchical system structures," *IEEE Trans. Syst. Man Cybernetics*, vol. 11, no. 2, pp. 109–125, Feb. 1981.

[47] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko, "Effectiveness of animation in trend visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1325–1332, Nov./Dec. 2008.

[48] "Semantic scholar," [Online]. Available: https://www.semanticscholar.org/

[49] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 113–120.

[50] J. Stasko, C. Gorg, Z. Liu, and K. Singhal, "Jigsaw: Supporting investigative analysis through interactive visualization," in *Proc. IEEE Symp. Visual Anal. Sci. Technol.*, 2007, pp. 131–138.

[51] "Aminer citation network data set," [Online]. Available: https://cn.aminer.org/citation

[52] "Citeseerx citation data set," [Online]. Available: http://csxstatic.ist.psu.edu/about/data

[53] D. H. Chau, A. Kittur, J. I. Hong, and C. Faloutsos, "Apolo: Making sense of large network data by combining rich user interaction and machine learning," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2011, pp. 167–176.

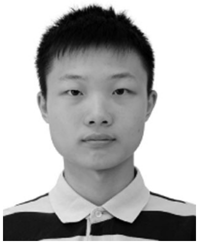
[54] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 957–966.

[55] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[56] N. Elmqvist and J.-D. Fekete, "Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 3, pp. 439–454, May-Jun. 2010.

[57] M. De Domenico, A. Lima, P. Mougel, and M. Musolesi, "The anatomy of a scientific rumor," *Sci. Reports*, vol. 3, 2013, Art. no. 2980.

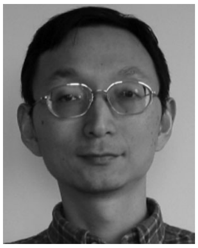
[58] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," 2014. [Online]. Available: http://snap.stanford.edu/data



Yucheng Huang is working toward the senior graduate degree at Peking University and a visiting student at SKLCS, Institute of Software, Chinese Academy of Sciences. His research interests include data visualization and big data algorithms.

Lei Shi received the BS, MS, and PhD degrees from the Department of Computer Science and Technology, Tsinghua University, in 2003, 2006 and 2008. He is a professor in ACT & BDBC, School of Computer Science and Engineering, Beihang University. Previously, he was a professor in SKLCS, Institute of Software, Chinese Academy of Sciences. His research interests span Information Visualization, Visual Analytics, and Data Mining. He is the recipient of the IBM Research Accomplishment Award on "Visual Analytics" and the VAST Challenge Award twice in 2010 and 2012.

Yue Su received the BS degree from the College of Electronics and Information Engineering, Tongji University, and the MS degree from SKLCS, Institute of Software, Chinese Academy of Sciences. His research interests include information visualization and data mining. He has published several papers in data mining and visualization conferences.



Yifan Hu received the BS and MS degrees in applied mathematics from Shanghai Jiao-Tong University, and the PhD degree in optimization from Loughborough University, United Kingdom. He is currently a Principal research scientist at Yahoo Labs, having previously worked at AT&T Labs. He is a contributor to the Graphviz graph drawing system. His research interests include data mining, information visualization, and algorithms.

Hanghang Tong received the MSc and PhD degrees from Carnegie Mellon University, in 2008 and 2009. He is currently an associate professor at School of Computing, Informatics, and Decision Systems Engineering, Arizona State University. Before that, he was an assistant professor at Computer Science Department, City College, City University of New York, a research staff member at IBM T.J. Watson Research Center. His research interests include large scale data mining. He has received the ICDM 10-Year highest impact paper award, four best paper awards and four "best of conference".

Chaoli Wang received the PhD degree from the Ohio State University, in 2006. He is an associate professor of computer science and engineering at University of Notre Dame. His main research interest is scientific visualization, in particular on the topics of time-varying multivariate data visualization, flow visualization, as well as information-theoretic algorithms, graph-based techniques, and deep learning solutions for big data analytics. He is a recipient of the NSF CAREER Award in 2014, an honorable mention at IEEE PacificVis 2013, and best paper awards at I&ST/SPIE VDA in 2013, 2015, and 2019. He is a senior member of the IEEE.

Tong Yang received the PhD degree in computer science from Tsinghua University, in 2013. Now he is a research assistant professor in Computer Science Department, Peking University. His research interests include network measurements, sketches, IP lookups, and KV stores. He published papers in SIGCOMM, SIGKDD, SIGMOD, VLDB, ATC, ToN, ICDE, INFOCOM.

Deyun Wang received the BS degree from the School of Information, Central University of Finance and Economics. He is currently working toward the graduate degree in SKLCS, Institute of Software, Chinese Academy of Sciences. His research interest is network visualization.

Shuo Liang received the bachelor's degree from the LuXun Academy of Fine Arts, and the master's degree from the Department of Visual Communication, Academy of Arts & Design, Tsinghua University. She is engaged in brand image design in the Institute of Arts & Design, Graduate School at Shenzhen, Tsinghua University. Her master graduation project has received the Eyas Award in 2016 Sun Cup Asia Label Awards.