Influence Maximization With Visual Analytics

Alessio Arleo, *Member, IEEE*, Walter Didimo, *Member, IEEE*, Giuseppe Liotta, *Senior Member, IEEE*, Silvia Miksch, *Member, IEEE*, and Fabrizio Montecchiani

Abstract—In social networks, individuals' decisions are strongly influenced by recommendations from their friends, acquaintances, and favorite renowned personalities. The popularity of online social networking platforms makes them the prime venues to advertise products and promote opinions. The *Influence Maximization* (IM) problem entails selecting a *seed set* of users that maximizes the influence spread, i.e., the expected number of users positively influenced by a stochastic diffusion process triggered by the seeds. Engineering and analyzing IM algorithms remains a difficult and demanding task due to the NP-hardness of the problem and the stochastic nature of the diffusion processes. Despite several heuristics being introduced, they often fail in providing enough information on how the network topology affects the diffusion process, precious insights that could help researchers improve their seed set selection. In this paper, we present VAIM, a visual analytics system that supports users in analyzing, evaluating, and comparing information diffusion processes determined by different IM algorithms. Furthermore, VAIM provides useful insights that the analyst can use to modify the seed set of an IM algorithm, so to improve its influence spread. We assess our system by: (i) a qualitative evaluation based on a guided experiment with two domain experts on two different data sets; (ii) a quantitative estimation of the value of the proposed visualization through the ICE-T methodology by Wall *et al.* (IEEE TVCG - 2018). The twofold assessment indicates that VAIM effectively supports our target users in the visual analysis of the performance of IM algorithms.

Index Terms—Information visualization, visualization systems and software, influence maximization, visual analytics, information diffusion

1 Introduction

People in social networks influence each other in both direct and indirect ways, through a mechanism often known as the *word-of-mouth effect* (see, e.g., [1], [2]). For instance, individuals' decisions to purchase a product or adopt an opinion are strongly influenced by recommendations from their friends and acquaintances. For this reason online social networking platforms are becoming the favorite venue where companies advertise their products/services and where politicians run their campaigns. In this context, research on so-called *influence maximization* focuses on understanding and leveraging such influence to obtain a much larger spread of the product or opinion than traditional marketing campaigns targeted to single individuals. Formally, the *influence maximization problem* (IM) asks to select a *seed set* of users that maximizes the influence spread, i.e., the expected number of users positively influenced by an information diffusion process, triggered by the seeds and evolving according to some stochastic diffusion model. The set of seeds should be relatively small, as users in this set have to be targeted individually. For instance, a classical application of IM is viral marketing [3], where companies want to maximize the adoption of a new product starting from some carefully selected early adopters who represent the seed set triggering the diffusion process.

Under the most common stochastic diffusion models, finding the optimal seed set in a network is known to be an NP-hard problem [1]. On the positive side, a greedy algorithm guarantees that the optimal influence spread can be approximated within a factor of $(1 - \frac{1}{e})$, where e is the base of the natural logarithm [1]. Besides the problem hardness, being the information diffusion process stochastic, even the evaluation of influence spread of any seed set is computationally intensive [4]. This makes the design of scalable and effective IM algorithms a great challenge that motivated a large and still increasing body of literature [5]. To evaluate an IM algorithm, some commonly adopted key performance metrics are quality of spread, running time efficiency, and main memory footprint. In addition, it is desirable for the algorithm to be robust across diffusion models, networks (with distinct structural properties), and parameters. Overall, analyzing and engineering such algorithms remain difficult and demanding tasks; as reported by Arora *et al.* [6], after a thorough experimental analysis, there is no single state-of-the-art technique for IM.

Contribution. The objective of the research described in this paper is to exploit the power of visual analytics (VA) to support domain experts in analyzing, evaluating, and comparing IM algorithms. We present VAIM (Visual Analytics for Influence Maximization), a system that, besides providing facilities to simulate an information diffusion process

- *Alessio Arleo and Silvia Miksch are with the Centre for Visual Analytics Science and Technology, TU Wien, 1040 Vienna, Austria. E-mail: {alessio.arleo, silvia.miksch}@tuwien.ac.at.*
- *Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani are with the Engineering Department, University of Perugia, 06123 Perugia, Italy. E-mail: {walter.didimo, giuseppe.liotta, fabrizio.montecchiani}@unipg.it.*

Manuscript received 4 April 2022; revised 29 June 2022; accepted 7 July 2022. Date of publication 13 July 2022; date of current version 2 September 2022. This work was supported in part by the Smart CT Research Cluster at Vienna University of Technology, in part by MIUR under Grant 20174LF3T8 "AHeAD: Efficient Algorithms for HArnessing networked Data" in part by the University of Perugia under Grants RICBA19FM and RICBA20ED and in part by TU Wien Bibliothek through its Open Access Funding Programme. (Corresponding author: Alessio Arleo.) Recommended for acceptance by J-D. Fekete. This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2022.3190623, provided by the authors. Digital Object Identifier no. 10.1109/TVCG.2022.3190623

over a given network, offers problem-oriented VA tools to explore the related data. An early version of this research has been presented as a short paper [7]. The main features of VAIM are as follows.

- VAIM implements some of the most popular IM algorithms (e.g., HIGHDEG [1] and SDISC [8]) and information diffusion models, such as the IC (Independent Cascade) and LT (Linear Threshold) models. The modular architecture of VAIM makes it possible to easily integrate additional implementations, which can also be executed on distributed cloud-based infrastructures.
- VAIM offers multiple interactive coordinated views that make it possible to visually compare and analyze the performance of a diffusion model over complex networks with thousands of vertices, for different choices of the seed sets (i.e., for different IM algorithms or for different choices of the parameters of the same algorithm).
- Besides the analysis and evaluation of IM algorithms, VAIM allows users to interactively modify the seed set and iterate the process until a satisfying spread is achieved. Such feature can be used to either fine-tune the output of an algorithm or to improve the design of an algorithm via reverse engineering.

We performed a twofold assessment of the effectiveness of VAIM: (i) A qualitative guided experiment with two domain experts on two different data sets; (ii) an evaluation based on the ICE-T methodology [9], aimed to quantitatively measure the *value* of the examined visualization and infer the quality of the underlying design choices. The assessment indicates that VAIM effectively supports users in the visual analysis of the performance of IM algorithms.

Paper Organization. The remainder of this paper is structured as follows. Section 2 contains a brief review of the literature related to our research. Section 3 provides basic background and notation. Section 4 illustrates our approach and presents the main features of VAIM. Section 5 describes our twofold assessment process. Section 6 contains a summarizing discussion, outlining the limitations of our approach and opportunities for further research in this context. Section 7 concludes the paper.

2 Related Work

This section reviews the main literature on the visual analysis of diffusion processes that may arise in various contexts and highlights the main differences with our approach. We first review the visualization of information diffusion on social networks and then its applications in other domains. We refer the reader to the works by Guille *et al.* [10] and by Li *et al.* [5] for surveys about influence maximization and information diffusion in social networks.

2.1 Information Diffusion in Social Networks

There are several visualization systems designed to analyze information diffusion processes in social networks. TwitInfo [11], [12] aggregates tweets in the spatial, temporal, and event dimensions supporting the exploration of event propagation processes. Whisper [13] exploits a flower-like visualization for real-time monitoring of the diffusion of a given topic, highlighting the spatio-temporal information of the process over the world. OpinionFlow [14] uses Sankey diagrams and density maps to visually summarize opinion diffusion processes. FluxFlow [15] adopts a timeline visualization to analyze anomalous information diffusion spreading. D-Map [16] collects data from Sina Weibo (a chinese microblogging website) and offers a map-based ego-centric visualization to reveal dynamic patterns of how people are involved and influenced in a diffusion process. SocialWave [17] uses abstract visualizations to explore and analyze spatio-temporal diffusion of information, taking into account further factors in the diffusion process such as cultural proximity and linguistic similarity. Visual-VM [18] provides a visualization system for viral marketing in social networks. More approaches are elaborated in Chen *et al.* [19].

All the aforementioned systems and approaches are designed to reveal different facets of the diffusion processes, with the resulting visualization merging geographical and other user-related information. On the other hand, unlike VAIM, they neither support the user in analyzing the impact of the seeds (which in fact may be unknown) and of the network structure in terms of influence spread, nor offer simulation tools to experiment different diffusion models and/or IM algorithms. Also, the networks analyzed with VAIM may come from diverse scenarios and may not contain geographical information about users.

Vallet *et al.* [20] present a visualization framework to compare different diffusion models based on a common set of graph rewriting rules. Different from VAIM, Vallet *et al.* do not focus on comparing different IM algorithms. Also their visualization approach is suitable for networks with up to few hundreds of vertices and edges.

SpreadViz [21] is a Web-based system that allows users to visualize a diffusion process by starting from one or multiple seeds. The user can choose the number of seeds and the criterion applied by the system to select them, as the most influential nodes according to one of some predefined centrality indexes. Also, the user can choose one among three possible stochastic diffusion processes. The spread is shown by progressively highlighting vertices in a node-link diagram computed by the force-directed algorithm in the popular D3.js library [22]. To set up the simulation, the user can upload a network and select among several diffusion models. The main view of the user interface presents a node-link visualization of the network, showing all the nodes and their links. Each node is assigned a color to represent its status in a specific time instant of the simulation selected by the user. The paper presents a demo with a graph with 3k nodes and 8k edges. While it is an interesting tool to perform simulations, unlike VAIM it does not provide a direct support to comparing different simulations in the same view and they do not allow users to modify the seed set during the visual analysis.

NDLib [23] is a general-purpose simulation framework written in Python to easily run information diffusion simulations. As a library, it does not provide a visualization component that depicts the evolution of the process over the network graphically, however we include it in this discussion for the sake of completeness.

We finally mention an interesting work by Saito *et al.* [24], which describes a new force-directed layout algorithm driven

by a diffusion process under the LT or the IC model. This algorithm incorporates conditional probability of information diffusion between two nodes with the aim of computing visualizations that guarantee path continuity (i.e., any information diffusion path is continuous) and path separability (i.e., each different information diffusion path is clearly separated from each other). This algorithm was evaluated on graphs with up to 10k nodes and 245k edges. VAIM offers a simultaneous visual comparison of different diffusion processes on the same graph layout, which makes it difficult to exploit the layout algorithm by Saito *et al.*

2.2 Applications of Information Diffusion

Information diffusion processes include, among others, disease spreading in social environments (see, e.g., [25]), malware diffusion among computers (see, e.g., [26]), and mobile phone viruses propagation (see, e.g., [27]). The spreading of these epidemic phenomena are often described as information diffusion processes over suitable networks: The visualization of the "information spread" in the network helps in understanding the dynamics of the phenomenon.

Afzal *et al.* [28] propose an interactive VA tool to monitor the simulation of a disease outbreak. The goal of the system is to provide a decision support environment in which users can explore epidemic models and their long and short term impact. A library of mitigation measures is included, so that the users can evaluate their effect in containing the epidemic. The tool is targeted to epidemiologists, local public health officials and other healthcare officials. This system only shows the macroscopic effects of the spread, while VAIM also aims at uncovering the local effects of diffusion processes.

Bryan *et al.* [29] propose Epidemic Simulation System (EpiSimS), a modeling tool for analyzing disease spread within the US. Visual analysis is used to provide guidance for users with limited knowledge in statistical modeling to understand the epidemic phenomenon. The system also provides predictive models which help forecasting the evolution of the disease. The system's goal is to guide the user in all the stages of the decision process: from the choice of the model parameters, to the analysis of the results of the simulation. Maciejeweski *et al.* [30] present a system to identify hotspots and trends in spatio-temporal data. The system is used with healthcare surveillance data, and the linked view approach is used to predict and forecast the potential health threats to a community. The goal of the tool is to find potential hotspots vulnerable to a disease outbreak. Differently from VAIM, the systems in [29], [30] allow users to build models that predict the spread considering their knowledge of the application domain, thus their focus is not on the visual analysis of the diffusion process but only on the outcome of the simulation.

Guo [31] presents a system to visualize interaction patterns for pandemic relief decision support. The system encodes the movements of people between places ("activity graph") and whether someone spread the contagion from one place to the other ("spread graph"). The visualization combines matrices and flow graphs: differently from VAIM, the data is not time-dependent, as all the links are contained in a single instance of the activity and spread graph.

Van den Broeck *et al.* [32] present GLEaMviz, a publicly available software that simulates the spread of human-to-human infectious diseases on world scale. The simulations combine real-world data on populations and human mobility with stochastic models of disease transmission. In terms of visualization, the system provides a dynamic geographic map and charts describing the geo-temporal evolution of the disease. Also, many recent VA systems focus on the infection spread related to COVID-19 pandemic (see, e.g., [33], [34]). As already mentioned, all these systems differ from VAIM as they mainly exploit visualizations on geographic maps instead of network visualizations.

3 Background and Notation

We model a social network as a directed edge-weighted graph $G = (V, E)$, where each node in the set V represents a user and each edge in E represents a direct relationship between two users. Namely, a direct edge $(u, v) \in E$ means that u can influence v with a probability that is related to the edge weight. Note that, considering directed graphs is not restrictive, as any undirected relationship can be modeled with two edges oriented in opposite directions.

A *diffusion model* M captures the stochastic information diffusion process among the nodes of G. During the process, a node $v \in V$ can be either *active* (i.e., already influenced) or *inactive* (i.e., not yet influenced). We only consider *progressive* models, where an inactive node may become active but not vice versa (this is the scenario adopted by most of the IM algorithms [5]). Let $S \subseteq V$ be the initial set of active vertices of G, called *seed set*. The *influence spread* of S, denoted by $\sigma_{G,M}(S)$, is the expected number of active vertices once the diffusion process - over the graph G and under the model M - terminates. More formally, the IM problem asks for a set $S^* \subseteq V$ of at most $0 < k \leq |V|$ seeds that maximizes the influence spread, that is:

$$S^* = arg\,max\{\sigma_{G,M}(S) | S \subseteq V \wedge |S| \leq k\}.$$

The most commonly used diffusion models are the *Independent Cascade* (IC) and the *Linear Threshold* (LT). Other models make use of additional parameters but do not differ significantly in terms of the underlying iterative framework. In the IC model, a diffusion instance unfolds through an iterative process: in step 0, only the seed vertices are active; in step $j > 0$, each vertex u activated at step $j - 1$ will activate each of its inactive neighbors v with probability $0 \leq p(u, v) \leq 1$. If u activates v we say that the edge (u, v) *triggers* the activation of v. The process halts when no more vertices can be activated. In the LT model, each edge $(u, v) \in E$ is associated with a weight $b(u, v) \leq 1$, and each vertex v has a threshold $0 \leq \Theta(v) \leq 1$. A diffusion instance follows again an iterative process. In step 0 the only active vertices are the seeds, while in step $j > 0$ any inactive vertex v becomes active as soon as the total weight of the edges incident to its active neighbors (which must be smaller than or equal to 1) is at least $\Theta(v)$. If v is activated, we say that all the edges that connect v to its active neighbors *trigger* the activation of v. The process halts when no more vertices can be activated.

Unfortunately, the IM problem is NP-hard under both the IC and LT models, as well as under other models [1]. A simple greedy approach to approximate the optimal solution exists [35], but it does not scale to large networks,

which motivated the design of faster IM heuristics. We refer the reader to seminal papers [1], [3] and to recent surveys [5], [10] for a broad dissertation.

4 The VAIM System

The design of VAIM relies on the "Data-Users-Tasks" model proposed by Miksch and Aigner [36]. Following this model, we describe the data (Section 4.1), as well as the users and the tasks (Section 4.2) driving our design process. Next we describe we describe the design of VAIM's visual interface (Section 4.3), which adopts an overview+detail approach. Finally, we discuss the main architectural choices of our implementation (Section 4.4).

4.1 Data

For each step of a diffusion process, we are interested in estimating the following three quantities: (i) the influence spread achieved at that step, (ii) the probability that a node is active at that step, and (iii) the probability that an edge triggered an activation at that step. Due to their stochastic nature, we model these quantities as random variables, and we employ a Monte Carlo method for their estimation. Namely, given a graph $G = (V, E)$, a diffusion model M and a seed set S, we repeatedly simulate the corresponding diffusion process, until the influence spread converges. The convergence is obtained when the difference in terms of influence spread between two consecutive simulations (over the entire sequence of simulations already computed) goes below a predefined threshold (which we set at 2%). Let K be the number of performed simulations and let T be the maximum number of steps over all simulations. For each step $0 \leq j \leq T$ and for each node $v \in V$, denote by $s(v, j)$ the number of simulations in which v has been activated at any step $i \leq j$. The spread at step j indicates how many vertices on average one can expect to be active at step j; we define it as $\sigma^*(j) = \frac{\sum_{v \in V} s(v,j)}{K}$. Also, we define the probability that v is active at step j as $\alpha(v, j) = \frac{s(v,j)}{K}$. Similarly, for each edge $e \in E$, denote by $s'(e, j)$ the number of simulations in which e triggers an activation at any step $i \leq j$. We define the probability that e triggers an activation either at step j or at some previous step as $\beta(e, j) = \frac{s'(e,j)}{K}$. Clearly, for any given vertex v and for any given edge e, the functions $\alpha(v, j)$ and $\beta(e, j)$ do not decrease as j increases. According to the Monte Carlo method, we assume that the arithmetic mean is a good estimator for the three quantities we are interested in. Besides the arithmetic mean, we also store the corresponding standard deviation.

This procedure yields a triple $\langle G, \alpha, \beta \rangle$ that can be viewed as an uncertain and dynamic graph G, where each node v and edge e exist at any step $0 \leq j \leq T$ with probability $\alpha(v, j)$ and $\beta(e, j)$, respectively.

4.2 Users and Tasks

VAIM primarily targets researchers in the influence maximization and information diffusion domains, investigating how diffusion models perform under different graph topologies. VAIM can also support users that require investigating diffusion processes in their own application domain, such as epidemiologists. Usually such users are not aiming to explore the whole network, but rather focus on smaller sections for more detailed investigations. Consequently, an overview visualization should support the selection process and a more sophisticated visual exploration environment should ease the investigation of the spread and how the structure of the network influences the diffusion process. To this end, VAIM is designed to support the following tasks:

T1Simulate. Estimate the outcome of a diffusion process on a given network under a given diffusion model, with the seed set computed by an IM algorithm.

T2Evaluate. Visually analyze both the quality of spread of a seed set and the impact of the network structure on the diffusion process, such as areas with a higher rate of active nodes, isolated areas, etc. The user can fast forward, rewind, and pause the process animation.

T3Compare. Visually compare the performance of different seed sets computed by different IM algorithms.

T4Feedback. Modify the seed set and iterate the simulate-evaluate-compare process.

4.3 Visualization Design

The visualization design adopts a focus+context approach. The interface is organized as a dashboard with multiple coordinated views (see Fig. 1). The chosen colour schemes and palettes are colorblind friendly [37].

– Diffusion process control (Fig. 1 A). This area is opened on demand and hidden during the analysis process. Here the user can load a graph and the corresponding simulations onto the visualization. Moreover, from here it is possible to launch a new simulation, by setting parameters such as the stochastic model and seed selection technique (Task T1). The seed set can be computed either randomly or by an IM algorithm by providing the set target size (or "budget"). The initial seed set can be also selected by modifying a previously used one available in the system according to the feedback offered by VAIM (Task T4), as explained later.

–Density matrix view (Fig. 1 B). This component provides an overview of the network structure. It consists of a schematic matrix visualization, which is obtained by firstly computing a node-link layout of the whole network with some fast algorithms, such as centralized or distributed force-directed techniques (e.g., [38], [39], [40]), and then by slicing the plane into cells. The color intensity of each cell reflects the number of nodes that fall in that cell. This view gives a high-level idea of how the nodes are distributed in the drawing area. Since force-directed algorithms tend to keep close together groups of nodes that are strongly related to each other, one can assume that cells with high node concentration correspond to dense portions of the network. Structural details of portions of the network that fall in one or more cells can be obtained through an exploration with the node-link view (see below), in a focus+context fashion. The number of cells in the matrix view can be increased/decreased through a slider. Hovering with the mouse on a cell (if no simulations are loaded), opens a tooltip with the number of nodes in that cell. This approach requires very little data transfer between the server and the client (see Section 4.4), it approximates the network structure highlighting higher density areas, and remains readable as the graph size increases. The layout for the generation of the matrix can be computed automatically when the graph is loaded in

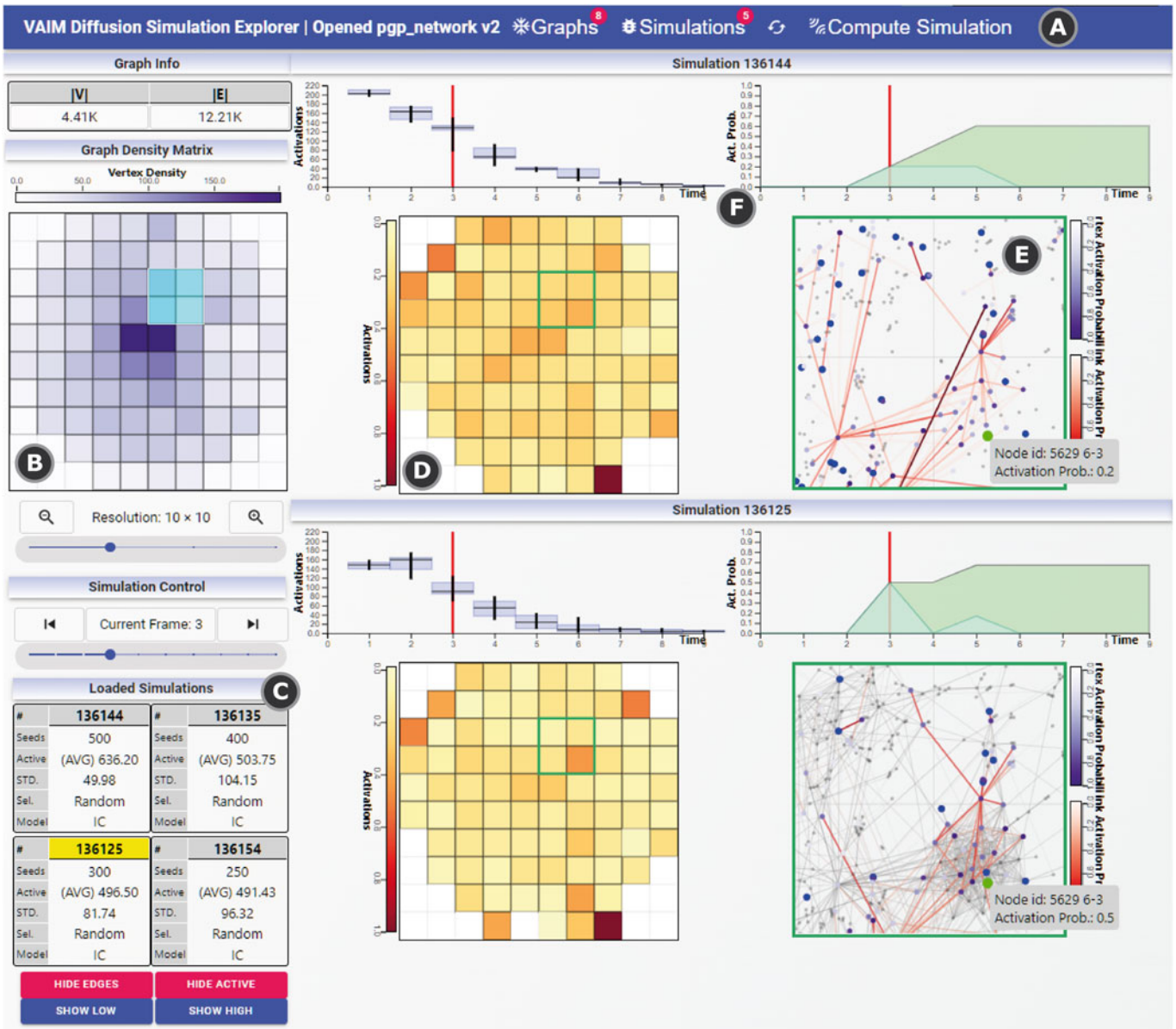


Fig. 1. An overview of VAIM's visual interface, which shows two distinct simulations on the same network at a specific time step (the interface supports the simultaneous comparison of up to four simulations, arranged in a 2×2 grid). The network consists of 4,410 nodes and 12,210 edges. The components of the interface are the following: (A) Diffusion Process Control; (B) Density Matrix View; (C) Simulation Control; (D) Diffusion Matrix View; (E) Node-link View; (F) Process Trend View.

the system the first time or can be pre-computed by the user (for example to obtain a layout that highlights different aspects of the topology of the graph), and then loaded in VAIM. The layout is saved in the database and not recomputed to ensure efficiency. In our prototype, we use *SFDP* by Hu [41] as the algorithm for the first layout, as its code is easy to integrate [42] and it is able to scale up to graphs with millions of nodes and edges (see, e.g., [39]).

– Simulation Control (Fig. 1 C). The controls in this view allow the user to manipulate the visualizations of the currently loaded simulations (Task T2), and compare them on some aggregated metrics and high-level information (Task T3). Using a time slider, the user can navigate through time (i.e., the different frames of each simulation), causing all the loaded simulations to simultaneously animate. Below, the basic information and statistics about the loaded simulations are aggregated into four distinct subareas arranged into a grid: these information include the number of seeds, the algorithm used for seed selection, the diffusion model, the average number of expected activate nodes (i.e., the spread) and its standard deviation. The user can select any of the loaded simulations by clicking on their respective box to execute additional operations on their individual views. The part of interface to the right of the Simulation Control is split into up to four distinct subareas arranged into a grid: Each subarea corresponds to one of the simulations under comparison and consists of different interactive views, which are described below.

– Diffusion matrix view (Fig. 1 D). For a specific simulation, this view shows the spread distribution over the considered network (Task T2). The diffusion process is conveyed using a schematic matrix visualization defined on the same set of cells as the density matrix. This choice facilitates not only the association between the diffusion matrix and the

density matrix views, and therefore correlate the spread distribution with the structure of the network, but also comparisons among diffusion matrix views of distinct diffusion processes (Task T3). The color of each cell varies in a *YlOrRd* scale (yellow to orange to red), and reflects the "amount" of nodes in the cell that are likely active at the considered step of the diffusion process. More precisely, for a cell c and for a given step j of the diffusion process, we compute the mean of the probabilities $\alpha(v, j)$ over all nodes v in c, and we map this value in the *YlOrRd* scale to establish the color of c. When this value is 0, the cell is filled with white. By using the "Show Low" and "Show High" buttons from the Simulation Control view, the user can highlight the cells in this view with low and high *efficiency* respectively, measured as the average activation probability over all the nodes in the cell at the last step of the simulation. When either of the buttons is pressed, in the corresponding cells a number appears, representing the efficiency multiplied by a factor 100. By hovering with the mouse on any cell, a tooltip reveals the cell value, and the same cell is highlighted in both the density and other diffusion matrices (along with their corresponding tooltips) to ease the comparison between different simulations. Similarly to the density matrix view (see above) we chose a matrix to display the spread progress as it scales to larger graphs with little to no impact on the system's performance. Moreover, as the node distribution is the same as the density matrix view, it is possible to directly correlate the temporal evolution of the spread with the vertex density. We investigated the use of pixel-based representations: however, they would have likely required further onboarding to the user and we would have lost the correlation between the spread dynamics and node density.

– Node-link view (Fig. 1 E). Besides each diffusion matrix, a detailed node-link diagram of a portion of the network can be visualized. The user can freely choose this portion through a brushing selection of any group of $k \times h$ cells in the density matrix. The combination of this view with the density and diffusion matrix enables a scalable focus+context exploration and assessment of the diffusion spread (Task T2). In the diagram, the nodes' individual appearance changes according to their current status in the diffusion process. Inactive nodes are smaller and colored in dark grey; active nodes are larger and colored in a scale ranging from white to purple; the color reflects the activation probability up to the considered time step (white corresponding to probability 0 and purple corresponding to probability 1). To differentiate the seeds from the other nodes, they are colored blue since the beginning of the simulation. Similarly, the edges are colored in a scale ranging from white to red; the color reflects the probability that the edge is used in the diffusion process up to the considered time step. Differently from nodes, inactive edges are normally hidden; by interacting with the Simulation Control, the user can either hide/show the underlying edge structure (Fig. 1 C) and/or all edges whose activation probability is not zero. Finally, the node-link view can be used to modify the current seed set (Task T4), by receiving suggestions from the system. Namely, the user can select an area of the view, and the system retrieves data from the corresponding cell of the diffusion matrix view. If the cell has high (low) efficiency, the system suggests nodes to be removed (added) from the seed set, ranked by increasing (decreasing) out-degree (i.e., the number of nodes they can influence); the user can choose how many of these nodes must be selected. This operation can also be done regardless of the system suggestions.

– Process trend view (Fig. 1 F). As for the diffusion matrix view, we have a process trend view for each diffusion process under comparison. Each process trend view consists of two charts, one located above the corresponding diffusion matrix view and the other above the node-link view. The first chart conveys the spread $\sigma^*(j)$ at each step j through a series of box-and-whisker plots. The second chart is coordinated with the node-link view; namely, when the user hovers on a node v, the chart shows the activation probability $\alpha(v, j)$ of v within the time instant j and also the probability that v is activated exactly at j (Task T2). Doing so highlights the same nodes on the other loaded simulations and triggers their respective trend views (see Fig. 1) to enable the comparison with other processes (Task T3).

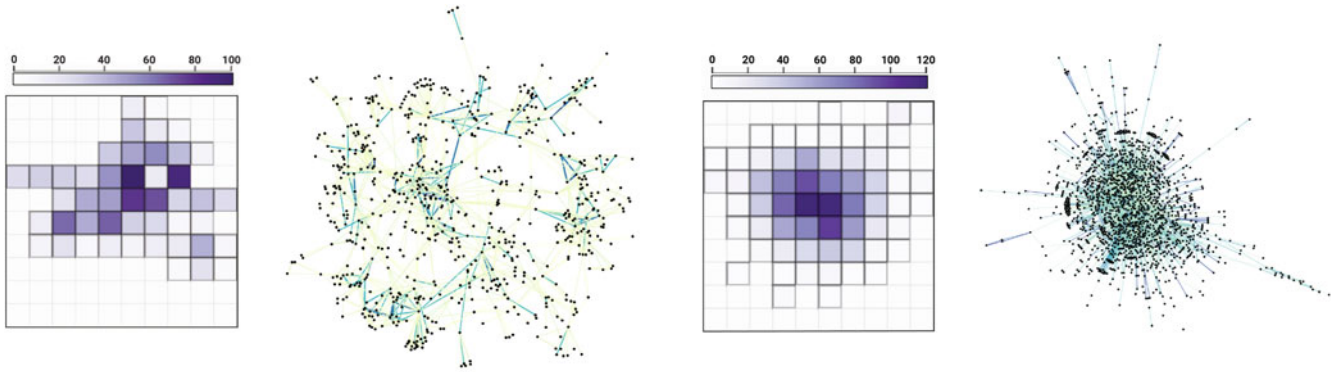
4.4 Architecture

To support task T1, VAIM must simulate multiple diffusion models on networks of various size. On the other hand, tasks T2–T4 should be supported by an advanced user interface that makes use of suitably designed visual abstractions (described in Section 4.3). Based on these requirements, VAIM has a client-server architecture structured as follows.

The server side hosts algorithms to simulate common diffusion models, as well as ad-hoc algorithms to visualize the output data; further diffusion models can be easily integrated by adding the corresponding implementations thanks to a modular design. The server also hosts IM algorithms for automatic seed selection. The current implementation of VAIM is written in Java 14 and includes custom implementations of the IC and LT diffusion models, and two seed selection algorithms: a strategy based on degree centrality, introduced by Kempe *at al.* [1] (HIGHDEG in the following), and the *SingleDiscount* (SDISC) heuristic proposed by Chen *at al.* [4] based on degree discount. The output data are stored in a Neo4j graph database [43]. The use of a graph database in this context has several advantages. In particular, retrieving specific subgraphs that match queries on node and edge attributes is simple and fast with Cypher, the built-in graph query language of Neo4J.

The client-side is a Web application developed using React [44] and it depends heavily on the D3 library [22] for data visualization. Server-side and client-side communicate via a REST API. The client is designed to accommodate up to four simulations at once. This bound on the number of simulations that can be simultaneously handled preserves interactivity and performance as the graph size increases, and it guarantees that all simulations can be displayed in a single view, thus favoring a juxtaposition-based comparison approach. We designed our system to fit a 16:9 display, and it was tested primarily on FullHD displays (with a resolution of 1920x1080).

The VAIM source code, for both the server and client, a sample database, and the server API description (for extending the capabilities of the system) are available online [45].



(a) Case Study 1: Co-authorship Network. (b) Case Study 2: Bitcoin Trust Network.

Fig. 2. The two networks used for the case studies with experts — (a) co-authorship network and (b) Bitcoin trust network. For each network, the figure depicts the density matrix visualization as shown in VAIM and the corresponding node-link representation with the edge weights encoded in color from yellow (low weight) to blue (high weight). The legend above the matrix represents the number of nodes per matrix cell.

5 Evaluation

In this section we describe the twofold assessment of VAIM, namely: Section 5.1 reports the results of two case studies with domain experts; Section 5.2 describes an evaluation based on the ICE-T method [9].

5.1 Case Studies With Experts

This assessment aims to establish whether VAIM can effectively support domain experts in the execution of Tasks T1–T4 (described in Section 4.2). Moreover, we look for insights on the experts' workflow and interaction patterns when using the system.

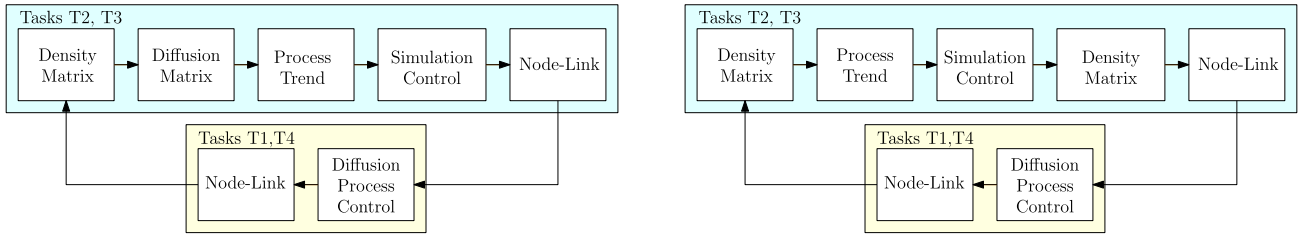
Each individual case study is conducted as follows. Before the session, we select a real network that is of interest for the expert. The networks used for the two case studies are shown in Fig. 2. Three simulations are run on the selected network: one with a random sampling of the seed set (RANDOM in the following) and the other two with two well-known heuristics for IM seed set selection included in VAIM, namely HIGHDEG and SDISC. In all cases, the size of the seed set is fixed to 10% of the number of nodes in the network. All the data is loaded onto an instance of VAIM running on a remote web server, accessible to the expert. The experiment session is held remotely and recorded (with the expert's consensus). The session is opened by a 15-minute presentation in which the system and its design are presented. The participant is then guided during a first hands-on session with VAIM. When ready, the participant is asked to: (i) obtain insights from the readily available simulations (T2, T3); (ii) modify the seed set of one of the existing simulations and run a new one (T1, T4); (iii) assess the effects of the new seed set and its impact on the resulting diffusion process (T2, T3). The interaction is supervised but unguided by the examiners. The expert could continue interacting with the system also after she completed the tasks. The case study is held with a think-aloud protocol; a critical feedback is also asked to the expert to summarize her experience.

We remark that the graphs we selected for the following experiments are real networks whose size is comparable to the one found in other experiments on graph visualization [46] and in other related papers about information diffusion (see, e.g., [24], [47]). In the following we also report the running times for our simulations on the datasets used in our case studies, obtained in a local installation (i.e., server, database, and client on the same machine) on a laptop with an Intel i7 8750H CPU with 16GB of RAM.

5.1.1 Case Study 1: Co-Authorship Network

The participant is a graph drawing and network visualization expert with notable experience in social network analysis. The network used for the case study is derived from a co-authorship network about papers published in the IEEE InfoVis conference from 1995 to 2015 [48]. In this network, the nodes represent authors and the edges represent co-authorship relationships. We processed the network in such a way that each edge (u, v) receives a weight proportional to the number of papers co-authored by u and v. We then normalize the edge weights such that they range in the interval [0,1]. We can interpret the edge weights as a measure of how much an author can influence another author (average 0.33, median 0.30). Indeed, it is reasonable to assume that a strict collaboration between two researchers results in a higher probability that one of the two researchers influences the other in spreading or exploiting scientific ideas/results on topics of common interest. Each edge is then oriented in one of the two possible directions, to reflect a scenario in which one of the two co-authors (e.g., the more expert) is more likely to influence the other. An alternative choice is to create, for every co-authorship relation, two edges of the same weight with opposite direction; we did not make this choice to avoid doubling the original density of the network. The resulting network has 698 nodes and 1,806 edges, and an average degree of 5. VAIM computes a full simulation, comprised on average of 4 different rounds, with the IC model in 6 seconds starting from a randomly sampled seed set of 10% the size of the vertex set (70). It takes 12 seconds with a seed set of 20% of the number of vertices. Most of the time is spent in database I/O.

A schematic illustration of the interaction workflow followed by the expert in this case study is shown in Fig. 3a. The participant began by inspecting the density matrix view. Since she was aware that the graph was a social network, she



(a) First case study. (b) Second case study.

Fig. 3. The different interaction workflows followed by the expert users in the two case studies.

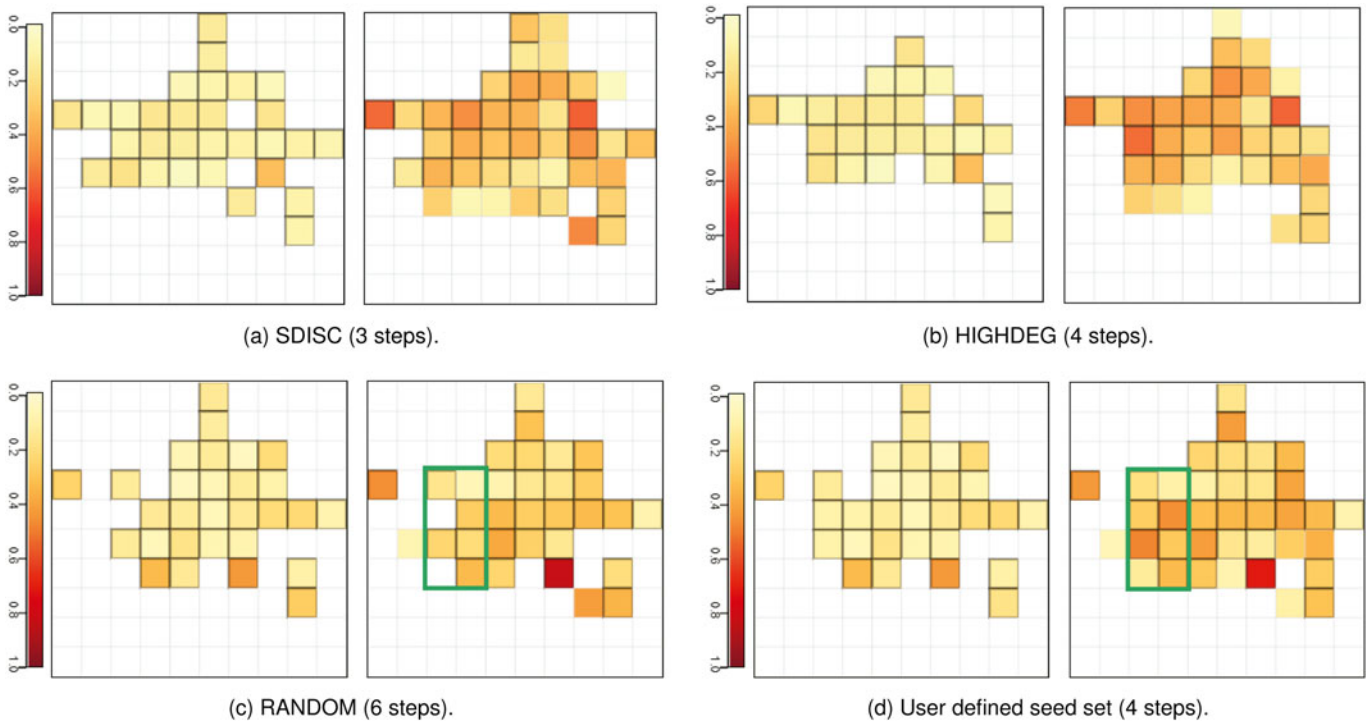


Fig. 4. Coverage matrices from Case Study 1. In each subfigure the left matrix represent the average activation probability per cell at step 0 (only seeds are active) and the right one shows the diffusion process at the last time step. Close-ups of the areas circled in green are shown in Fig. 5.

could hypothesize a scale-free structure, in which the hub nodes belong to cells with higher density. This hypothesis could be easily confirmed by looking at the diffusion matrix view, namely at the distribution of the seeds chosen by the heuristic algorithms (and HIGHDEG in particular), with the majority of the seeds (70 for each IM algorithm) placed in areas with high density (see Fig. 4b). Afterwards, the participant used the process trend view and focused on the number of time steps taken by each simulation to reach convergence. SDISC converged in 3 steps, HIGHDEG in 4, and RANDOM in 6. To explain this behavior, the user focused on the densest portions of the network using the node-link view. As expected, the heuristics (SDIS and HIGHDEG) had picked the majority of the hubs, while RANDOM mostly missed them. Hubs are unlikely to be close to each other, with the majority of hubs' neighbors having low degree and therefore a low chance to influence other nodes. Once hubs are active, the diffusion process subsides fast, but with a high activation rate.

Once satisfied with the acquired knowledge about the network and the simulation trends, the participant decided to use the VAIM tools to improve the spread of the RANDOM seed selection. She focused on cells with low coverage, searching for them using the "Show Low" button in the simulation control area. The participant sampled a few of them, and for each one she applied the following procedure. First, she compared the seed selection of RANDOM with SDISC and HIGHDEG using the node-link view. If RANDOM seeds were insignificant (i.e., with out-degree 0, therefore completely unable to influence other nodes) or with low out-degree, the participant replaced them with either the same or a smaller number of nodes suggested by VAIM. The result of this process was a selection of 67 seeds that yielded an average coverage of 120.20 nodes, with a 52% improvement over RANDOM, that had an average spread of 78.83 nodes. SDISC and HIGHDEG achieved an average spread of 148.20 and 145.60, respectively (with the original 70 seeds). Fig. 5 shows a portion of the network in the Node-Link view, in which it appears evident the increase of the spread using the modified seed set with respect to the RANDOM one.

5.1.2 *Case Study 2: Bitcoin Trust Network*

In this case study, we recruited an information visualization expert knowledgeable in cyber-security and blockchain

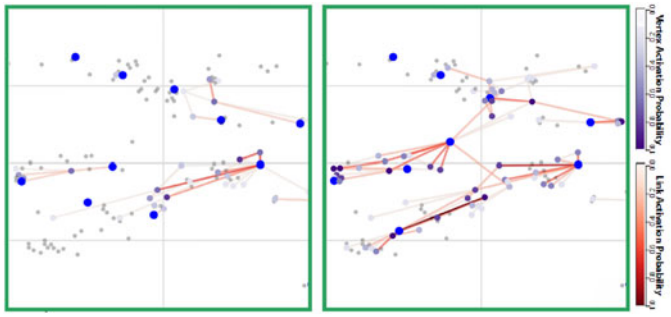


Fig. 5. Close-up of the diffusion matrix in Figs. 4c (left) and 4d (right) using the Node-Link view of VAIM. The modifications and improvements over the RANDOM seed set distribution are evident; the different choices of the seeds made a difference in this area of the network.

technology. We chose, as base network, the Bitcoin OTC weighted trust network [49], [50], a dynamic graph in which nodes represent people and edges represent the relation of trust/distrust between them, on a scale from -10 to +10, at a specific moment in time. We simplified the network by removing the negative-valued edges. Afterwards, we replaced multiple edges with a single edge whose weight is the sum of the individual trust weights. We applied a common logarithm to all the resulting weights, which we later normalized to bring them in the [0,1] interval. Finally, degree-1 nodes and edges whose weight is smaller than 0.3 or equal to 1 were removed, and only the largest connected component is considered. The result of this process is a static weighted network with 1,549 nodes and 4,656 edges, with an average degree of 3. Weights represent a value of *trust*, or influence, between two elements of the network. The average weight is 0.55, with a median value of 0.48. With this dataset, VAIM computes a full simulation, comprised on average of 4 different rounds, with the IC model in 31 seconds starting from a randomly sampled seed set of 10% the size of the vertex set (155). It takes 34 seconds with a seed set of 20% of the number of vertices. Compared to the running times in case study 1 (see Section 5.1.1), we observe that the running times grow slower with the number of seeds, mostly due to the lower average degree of this graph, ultimately resulting in less vertices being pinged for activation.

A schematic illustration of the interaction workflow followed by the expert in this case study is shown in Fig. 3b. The participant began by inspecting the density matrix view (shown in Fig. 2b); she observed the typical scale-free structure of the network, with a very dense core surrounded

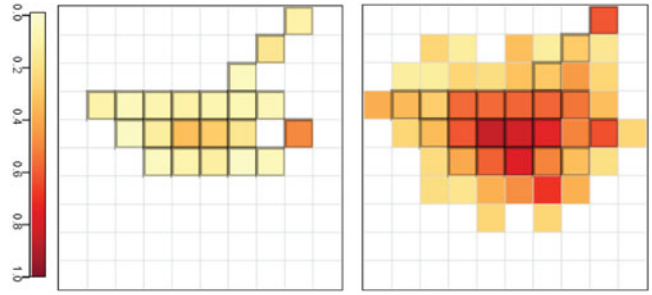
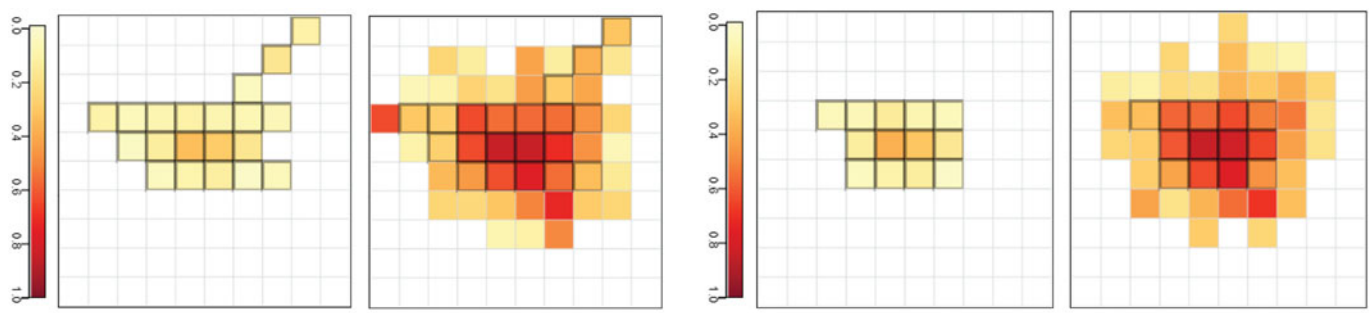


Fig. 7. Diffusion matrix for the user defined seed set in Case Study 2. The participant modified the SDISC seed distribution visible in Fig. 6a.

by a large fraction of peripheral nodes. The participant then focused on the process trend view, from which it was rather clear that the diffusion process quickly unfolded in few steps. She then used the simulation control to move forward the simulation over time. While interacting with the simulation control (see Fig. 6), the participant watched the diffusion matrix view and observed a rather expected pattern: most of the seeds where taken from the core of the network, where the diffusion process remained mostly confined. However, the participant also observed that SDISC also picked a significant amount of seeds from the less central cells of the matrix. She then used the node-link view to further analyze the effect of these peripheral seeds on the diffusion process and she observed that they were able to activate several nodes that were not reached by the seeds in the core of the network. Instead, HIGHDEG did not pick many such seeds and reported slightly worse performance in terms of spread (54% versus 52% respectively). See Figs. 6a and 6b.

Once the correlation between network structure and seeds was clear to the participant, she started modifying the seed set of SDISC, by using the diffusion process control and the node-link view, in a similar fashion as in the previous case study. The participant picked a cell with low coverage placed far from the core of the network and added two seeds from that cell (see Fig. 7). She then run a new simulation with 157 seeds (rather than 155) and waited few seconds to get the new results. Unfortunately, the overall spread decreased from 686 to 677, although it remained higher than the spread reached with HIGHDEG (see Figs. 6b and 7). While this behaviour may be counter-intuitive (a larger set of seeds led to a worse performance), it is most probably due to some statistical fluctuation together



(a) SDISC (4 steps). (b) HIGHDEG (5 steps).

Fig. 6. Coverage matrices from Case Study 2. In each subfigure the left matrix represents the average activation probability per cell at step 0 (only seeds are active) and the right one shows the diffusion process at the last time step.

with a poor choice for the new seeds. The participant concluded that the very dense core of the network plays a fundamental role in the diffusion process and that a further optimization of the performance of SDISC using few more seeds would require several attempts.

5.2 ICE-T Evaluation

Typical usability studies focus on the ability of a system to provide answers to questions about data. Stasko [51] introduced a framework to identify the *value* of a visualization, defined as its ability to convey a true *understanding* of data. This value is a linear combination of four components, each pertaining a specific ability of the visualization:

- *Time*: minimize the time to answer a wide variety of questions about the data;
- *Insights*: discover insights or insightful questions about the data;
- *Essence*: convey an overall essence of the data;
- *Confidence*: generate confidence, knowledge, and trust about the data, its domain and context.

Wall *et al.* [9] developed a methodology, referred to as *ICE-T*, that enables a quantitative assessment of the value of a visualization according to Stasko's equation. This methodology introduces a *hierarchical* extension of the original value framework: each of the four components of Stasko's equation comprises one to three *guidelines*, which capture the core-concepts of the corresponding components; each guideline contains one to three *heuristics*. A heuristic is an actionable and rateable statement that reflects how the visualization achieves the corresponding guideline. These heuristics will be individually rated by visualization (not necessarily *domain*) experts on a 7-point rating scale from 1 (*strongly disagree*) to 7 (*strongly agree*). This rating is collected using a survey[1] with a total of 21 heuristics. In the context of evaluating VAIM, we aim to obtain an average score equal or greater than 5 on all components, for it to be considered a *valuable* visualization [9].

According to the *ICE-T* methodology, we set the experiment as follows. We recruited 5 visualization experts; all of them had at least a basic familiarity with the concept of the IM problem and with the data set(s) used during the experiment. Each individual session started with a preliminary presentation and cold demo of VAIM, in order to make sure the participant was up to speed about its purpose and features. Afterwards, the participant began interacting with the system, making herself familiar with the controls. When ready, the participant was given a set of tasks to solve using the system. This set of tasks was meant to bootstrap her interaction with VAIM, but it was clarified that it was not mandatory to complete them to conclude the experiment. The system was made available as a Web application, and each participant could take as much or as little time to make up her mind about the quality and value of the visualization, and could continue interacting with the system and complete the evaluation offline. The evaluation process was considered completed when the ICE-T survey was filled and sent to the evaluators; we also recorded some post-evaluation feedback to complement their ratings.

1. Available on http://visvalue.org

TABLE 1
Results of ICE-T Evaluation on a 1 (Lowest) to 7 (Highest) Scale

Parameter	Average	Std. Dev.
Insight	6.4	0.71
Time	6	0.87
Essence	6.1	0.97
Confidence	5.9	1.15

A condensed view of the results is presented in Table 1; the complete score sheet is available as supplemental material. We aggregate the scores at each level of the hierarchy by averaging them. VAIM places itself on the higher end of the scoreboard, with the average rating on the Insight aspect close to the maximum (with a 6.4 out 7), followed by Essence (6.1), Time (6), and Confidence (5.9). A high score on the Insight and Essence components suggests that VAIM embodies the following guidelines: "provides a new or better understanding of the data, opportunities for serendipitous discoveries, and facilitates answering questions about data" and that "provides a big picture perspective of the data and an understanding of the data beyond individual data cases" [9]. Confidence, on the other hand, got the lowest score, but still well above our target of 5. In particular, the Confidence heuristic about data quality ("The visualization helps understanding data quality") got the lowest score, averaging 4.8. Differently from the other heuristics, this average was calculated on 4 scores only since one participant provided a "N/A" value to that question. This suggests that a limitation of our system resides in its capability of conveying information about the quality of the underlying data. However, data quality verification was not included in our design requirements.

Overall, on all components VAIM received an average score greater than 5, with a global average of 6.1, thus reaching our goal for this evaluation.

6 Discussion: Lessons Learned

In the following, we elaborate about the lessons learned and observed limitations, outlining directions for future work.

Computational Scalability. It is well-known that finding an optimal seed set in a network for influence maximization is NP-hard [1]. Analyzing and comparing the performance of different IM algorithms on real-world complex networks, by also highlighting the effects of network topology, pose various kinds of scalability issues. We stress-tested our system on a who-trust-whom network from the general consumer review site "Epinions.com" [52], with 75k nodes, 405k edges, and an average degree of 10; we left all the edge weights at 1, meaning that the maximum number of activations and the longest diffusion processes possible would occur in any simulation (i.e., worst case scenario). Moreover, this makes each process deterministic for a specific seed set, removing the need for multiple Monte-Carlo iterations. We used a laptop with an i7 8750H CPU and 16GB of RAM, and ran simulations with IC model and random sampling for seed selection (as in the data we reported in Section 5.1). We tested seeds sets being the 2.5%, 5%, and 10% of the size of the graph: running times averaged 33, 36, and 41 minutes respectively. This confirms our observation, as we already partly discussed in

Section 5.1, that simulation running times increase with the number of activations per round, which, in turn, depends on the average activation probability of the edges, graph connectivity, and the number of seeds. In our implementation, each new activation entails a (slow) database access, thus faster running times might be achieved by moving more of the computation onto system memory, rather than expressing the diffusion models using the Neo4J cypher language. Advanced programming paradigms, such as distributed/parallel computing on large scale graph processing platforms (see, e.g., [38], [39], [53]), might help tackling networks with millions of nodes. Nonetheless, we assume the simulation is done offline to support the interactive nature and responsiveness of the VA solution.

Conversely, also in this scenario, when exploring the diffusion process the system remained responsive also with few cells displayed in the node-link view and with multiple simulations. The choice of matrices as the main medium to display information about the network layout and the diffusion process allows VAIM to handle larger or comparable graphs with respect to existing approaches (see Section 2). Displaying larger portions of such a large graph in the node-link view (i.e., selecting a larger area on the density matrix view) can slow down the user interaction; this limitation can be partially overcome with a local installation, which in terms of time necessary for data transfer has a significant advantage over remote clients.

Visual Scalability. VAIM offers multiple coordinated views (compare, density matrix, diffusion matrix, node-link, and process trend views), and leverages juxtaposition to enable comparison between the different simulations (see Section 4.3). Juxtaposition allows us to compare multiple concurrently loaded simulations at once on all of their dimensions, without requiring new diagrams. However, this places the comparative burden on the user: we mitigate this issue by using coordinated views that provide visual cues that directly point to the user where to look (e.g., when highlighting a cell or node on the diffusion matrix/node-link view of a simulations, the same cell/node is highlighted in all the loaded simulations). We favored this approach as superimposition may suffer from clutter [54], especially when several objects have to be displayed in the same shared space. This technique would be applicable to compare our matrices and node-link visualizations as they share the same spatial context, but we believe juxtaposition provides a more flexible experience, giving the users the possibility of comparing only a subset of the loaded simulations without the need to remove them from view. Explicit encodings would unload the user from the comparative burden [54] (see e.g., [55], [56]), however they tend to suffer from decontextualization, that is the inability of tracing back from the relationship to the elements themselves, which could make it difficult to understand the differences between choosing a seed set over another. Further research is needed to come up with guidelines that would support designing a specialized comparison methodology for this application domain.

Our experiments showed that picking seeds from the node-link view can be challenging with very dense graphs. Hybrid visualization models can help in dealing with locally dense portions (see, e.g., [57], [58], [59], [60]), while sparsifying the network through edge filtering techniques and edge bundling seem to be needed when the network exhibits a global hairball effect (see, e.g., [61], [62]).

Interaction Techniques and Data Semantics. VAIM's design is performed according to specific tasks (Tasks T1–T4 described in Section 4.2), which ask for context-specific and task-oriented interaction techniques, almost completely focused on the network topology alone. Other semantic information relevant to the network, such as text and numeric node/edge attributes that leverage the domain knowledge of the users, may play a beneficial role in the analysis, and would require more advanced and context-specific interaction methods. These could be enriched by *guidance* and knowledge-assisted visual analytics [63] to improve the quality of the insights-gaining process.

Different Interaction Workflows. As illustrated in our two case studies, the experts follow different interaction workflows (see Figs. 3a and 3b). Consequently, it is crucial that VAIM provides the right level of flexibility needed to accomplish the various types of tasks for which it is designed.

Learning Curve to Use VAIM. Within our evaluation, we observed that the use of VAIM's interface requires a quite steep learning curve. However, once this initial price is payed, the interface becomes generally easy to use. We expected such a learning curve because of the intrinsic complexity of the VA problem we address, with many interleaved points of views and perspectives.

Data Quality. The *ICE-T* methodology [9] captures various dimensions. Confidence got the lowest score, because our VA approach does not tackle data quality. Consequently, the results of the *ICE-T* need to be contextualized according to the dimensions addressed.

7 Conclusion

We presented the use of VA concepts to support the analysis, comparison, and fine tuning of IM strategies, which also focus on the effects of network topology. A twofold evaluation of the proposed system (namely, two case studies and the use of the *ICE-T* methodology) gives evidence of the usefulness of our approach.

References

[1] D. Kempe, J. M. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2003, pp. 137–146.

[2] D. Kempe, J. M. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," *Theory Comput.*, vol. 11, pp. 105–147, 2015.

[3] P. M. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2001, pp. 57–66.

[4] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2010, pp. 1029–1038.

[5] Y. Li, J. Fan, Y. Wang, and K. Tan, "Influence maximization on social graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 10, pp. 1852–1872, Oct. 2018.

[6] A. Arora, S. Galhotra, and S. Ranu, "Debunking the myths of influence maximization: An in-depth benchmarking study," in *Proc. SIGMOD Conf. ACM*, 2017, pp. 651–666.

[7] A. Arleo, W. Didimo, G. Liotta, S. Miksch, and F. Montecchiani, "VAIM: Visual analytics for influence maximization," in *Graph Drawing Network Visualization*, D. Auber and P. Valtr, Eds., Cham, Switzerland: Springer, 2020, pp. 115–123.

[8] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 199–208.
[9] E. Wall *et al.*, "A heuristic approach to value-driven evaluation of visualizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 491–500, Jan. 2018.
[10] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *SIGMOD Rec.*, vol. 42, no. 2, pp. 17–28, 2013.
[11] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "TwitInfo: Aggregating and visualizing microblogs for event exploration," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2011, pp. 227–236.
[12] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Processing and visualizing the data in tweets," *SIGMOD Rec.*, vol. 40, no. 4, pp. 21–27, 2011.
[13] N. Cao, Y. 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. Graphics*, vol. 18, no. 12, pp. 2649–2658, Dec. 2012.
[14] Y. Wu, S. Liu, K. Yan, M. Liu, and F. Wu, "OpinionFlow: Visual analysis of opinion diffusion on social media," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1763–1772, Dec. 2014.
[15] J. Zhao, N. Cao, Z. Wen, Y. Song, Y. Lin, and C. Collins, "#FluxFlow: Visual analysis of anomalous information spreading on social media," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1773–1782, Dec. 2014.
[16] S. Chen *et al.*, "D-Map: Visual analysis of ego-centric information diffusion patterns in social media," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, 2016, pp. 41–50.
[17] G. Sun, T. Tang, T. Peng, R. Liang, and Y. Wu, "SocialWave: Visual analysis of spatio-temporal diffusion of information on social media," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, pp. 15:1–15:23, 2018.
[18] C. Long and R. C. Wong, "Visual-VM: A social network visualization tool for viral marketing," in *Proc. IEEE Int. Conf. Data Mining Workshop*, 2014, pp. 1223–1226.
[19] S. Chen, L. Lin, and X. Yuan, "Social media visual analytics," *Comput. Graph. Forum*, vol. 36, no. 3, pp. 563–587, 2017.
[20] J. Vallet, H. Kirchner, B. Pinaud, and G. Melançon, "A visual analytics approach to compare propagation models in social networks," in *Proc. Graphs Models Workshop*, 2015, pp. 65–79.
[21] K. Skianis, M. G. Rossi, F. D. Malliaros, and M. Vazirgiannis, "SpreadViz: Analytics and visualization of spreading processes in social networks," in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2016, pp. 1324–1327.
[22] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.
[23] G. Rossetti, L. Milli, S. Rinzivillo, A. Sîrbu, D. Pedreschi, and F. Giannotti, "NDlib: A python library to model and analyze diffusion processes over complex networks," *Int. J. Data Sci. Anal.*, vol. 5, no. 1, pp. 61–79, 2018.
[24] K. Saito, M. Kimura, and H. Motoda, "Effective visualization of information diffusion process over complex networks," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2008, pp. 326–341.
[25] S. Eubank *et al.*, "Modeling disease outbreaks in realistic urban social networks," *Nature*, vol. 429, pp. 180–184, 2004.
[26] P. Szor, "Fighting computer virus attacks," in *Proc. 13th USENIX Secur. Symp.*, 2004.
[27] P. Wang, M. C. González, R. Menezes, and A. Barabási, "Understanding the spread of malicious mobile-phone programs and their damage potential," *Int. J. Inf. Sec.*, vol. 12, no. 5, pp. 383–392, 2013.
[28] S. Afzal, R. Maciejewski, and D. S. Ebert, "Visual analytics decision support environment for epidemic modeling and response evaluation," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, 2011, pp. 191–200.
[29] C. Bryan, X. Wu, S. Mniszewski, and K.-L. Ma, "Integrating predictive analytics into a spatiotemporal epidemic simulation," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, 2015, pp. 17–24.
[30] R. Maciejewski *et al.*, "Forecasting hotspots—A predictive analytics approach," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 4, pp. 440–453, Apr. 2011.
[31] D. Guo, "Visual analytics of spatial interaction patterns for pandemic decision support," *Int. J. Geographical Informat. Sci.*, vol. 21, no. 8, pp. 859–877, 2007.
[32] B. W. V. D. C. Gioannini, M. Q. B. Gonçalves, V. Colizza, and A. Vespignani, "The GLEaMviz computational tool, a publicly available software to explore realistic epidemic spreading scenarios at the global scale," *BMC Infect. Dis.*, vol. 11, no. 37, pp. 859–877, 2011.
[33] S. Afzal, S. Ghani, H. C. Jenkins-Smith, D. S. Ebert, M. Hadwiger, and I. Hoteit, "A visual analytics based decision making environment for COVID-19 modeling and visualization," in *Proc. IEEE Visualization Conf.*, 2020, pp. 86–90.
[34] B. Chen *et al.*, "Visual data analysis and simulation prediction for COVID-19," 2020.
[35] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions - I," *Math. Program*, vol. 14, no. 1, pp. 265–294, 1978.
[36] S. Miksch and W. Aigner, "A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data," *Comput. Graph.*, vol. 38, pp. 286–290, 2014.
[37] M. Harrower and C. A. Brewer, "ColorBrewer.org: An online tool for selecting colour schemes for maps," *Cartographic J.*, vol. 40, no. 1, pp. 27–37, 2003.
[38] A. Arleo, W. Didimo, G. Liotta, and F. Montecchiani, "Large graph visualizations using a distributed computing platform," *Inf. Sci.*, vol. 381, pp. 124–141, 2017.
[39] A. Arleo, W. Didimo, G. Liotta, and F. Montecchiani, "A distributed multilevel force-directed algorithm," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 4, pp. 754–765, Apr. 2019.
[40] S. G. Kobourov, "Force-directed drawing algorithms," in *Handbook on Graph Drawing and Visualization*, R. Tamassia, Ed., Boca Raton, FL, USA: Chapman and Hall/CRC, 2013, pp. 383–408.
[41] Y. Hu, "Efficient, high-quality force-directed graph drawing," *Mathematica J.*, vol. 10, no. 1, pp. 37–71, 2005.
[42] E. R. Gansner and S. C. North, "An open graph visualization system and its applications to software engineering," *Softw. Pract. Exp.*, vol. 30, no. 11, pp. 1203–1233, 2000.
[43] Neo4j Graph Platform, Accessed: May 2020. [Online]. Available: https://neo4j.com/
[44] "React: A JavaScript library for building user interfaces," Accessed: May 2020. [Online]. Available: https://reactjs.org/
[45] "Vaim git repository," Accessed: Jun. 2022. [Online]. Available: https://github.com/EngAAlex/VAIM
[46] V. Yoghourdjian *et al.*, "Exploring the limits of complexity: A survey of empirical studies on graph visualisation," *Vis. Inform.*, vol. 2, no. 4, pp. 264–282, 2018.
[47] B. Pinaud, J. Vallet, and G. Melançon, "On visualization techniques comparison for large social networks overview: A user experiment," *Vis. Inform.*, vol. 4, no. 4, pp. 23–34, 2020.
[48] "Citevis citation datafile," 2016. Accessed: May 2020. [Online]. Available: http://www.cc.gatech.edu/gvu/ii/citevis/infovis-citation-data.txt
[49] S. Kumar, F. Spezzano, V. Subrahmanian, and C. Faloutsos, "Edge weight prediction in weighted signed networks," in *Proc. IEEE 16th Int. Conf. Data Mining*, 2016, pp. 221–230.
[50] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. Subrahmanian, "REV2: Fraudulent user prediction in rating platforms," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 333–341.
[51] J. Stasko, "Value-driven evaluation of visualizations," in *Proc. 5th Workshop Beyond Time Errors: Novel Eval. Methods Vis.*, 2014, pp. 46–53.
[52] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the Semantic Web," in *Proc. Int. Semantic Web Conf.*, 2003, pp. 351–368.
[53] J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica, "{GraphX}: Graph processing in a distributed dataflow framework," in *Proc. 11th USENIX Symp. Operating Syst. Des. Implementation*, 2014, pp. 599–613.
[54] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, "Visual comparison for information visualization," *Informat. Visualization*, vol. 10, no. 4, pp. 289–309, 2011.
[55] B. Alper, B. Bach, N. Henry Riche, T. Isenberg, and J.-D. Fekete, "Weighted graph comparison techniques for brain connectivity analysis," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2013, pp. 483–492.
[56] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson, "MatrixWave: Visual comparison of event sequence data," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, 2015, pp. 259–268.

[57] L. Angori, W. Didimo, F. Montecchiani, D. Pagliuca, and A. Tappini, "Hybrid graph visualizations with ChordLink: Algorithms, experiments, and applications," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 2, pp. 1288–1300, Feb. 2020.
[58] V. Batagelj, F. Brandenburg, W. Didimo, G. Liotta, P. Palladino, and M. Patrignani, "Visual analysis of large graphs using (X,Y)-Clustering and hybrid visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 11, pp. 1587–1598, Nov. 2011.
[59] N. Henry, J. Fekete, and M. J. McGuffin, "NodeTrix: A hybrid visualization of social networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 6, pp. 1302–1309, Jun. 2007.
[60] N. Henry, A. Bezerianos, and J. Fekete, "Improving the readability of clustered social networks using node duplication," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1317–1324, Nov./Dec. 2008.
[61] Y. Jia, J. Hoberock, M. Garland, and J. C. Hart, "On the visualization of social and other scale-free networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1285–1292, Nov./Dec. 2008.
[62] X. Huang and C. Huang, "NGD: Filtering graphs for visual analysis," *IEEE Trans. Big Data*, vol. 4, no. 3, pp. 381–395, Mar. 2018.
[63] D. Ceneda *et al.*, "Characterizing guidance in visual analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 111–120, Jan. 2017.

Alessio Arleo (Member, IEEE) received the BSc degree in computer and electronic engineering, the MSc degree in computer and TLC engineering, and the PhD degree from the University of Perugia, Italy, in 2018, defending a thesis titled "Distributed Large Graph Visualization: Algorithms and Experiments". He is a postdoc with CVAST group, TU Wien, Vienna, Austria. His research interests include graph drawing and visualization, distributed algorithms, and software engineering.

Walter Didimo (Member, IEEE) received the PhD degree in computer science from the University of Rome "La Sapienza", in 2000. He is currently an associate professor with the Department of Engineering, University of Perugia. His research interests include graph drawing, information visualization, algorithm engineering, and computational geometry. He collected more than 150 international publications in the above areas and chaired the program committee of the International Symposium on Graph Drawing.

Giuseppe Liotta (Senior Member, IEEE) is a professor with the Department of Engineering, University of Perugia, Italy. His research interests include information visualization, graph drawing, and computational geometry. On these topics, he published more than 250 research papers. He chaired the Steering Committee of the International Symposium of Graph Drawing and Network Visualization and currently serves as the editor in chief of *Computer Science Review* and of the *Journal of Graph Algorithms and Applications*.

Silvia Miksch (Member, IEEE) is a University professor and head of the Research Unit "Visual Analytics" (Centre for Visual Analytics Science and Technology (CVAST)), Institute of Visual Computing and Human-Centered Technology, TU Wien. She served as a paper co-chair of several conferences including IEEE VAST 2010, 2011, 2020, and VIS Overall Papers Chair (IEEE VIS 2021) as well as EuroVis 2012 and on the editorial board of several journals including *IEEE Transactions on Visualization and Computer Graphics* and Computer Graphics Forum. She acted in various strategic committees, such as the VAST steering committee and the VIS Executive Committee. In 2020 she was inducted into the IEEE Visualization Academy. Her main research interests include visualization/visual analytics (particularly Focus+Context and Interaction), and Time.

Fabrizio Montecchiani received the PhD degree in information engineering from the University of Perugia, in 2014, where he currently works as the associate professor. His research interests include graph drawing, computational geometry, visual analytics, and Big Data algorithms. He collected more than 100 scientific publications in the above areas. In 2021, he has been awarded by the ICEATCS as the Best Young Italian Researcher in Theoretical Computer Science. He has been a guest editor of the *Journal of Graph Algorithms and Applications*, and he has been a PC member of international conferences such as EuroCG, GD, MFCS, and WG.