This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.3030465, IEEE Transactions on Visualization and Computer Graphics

Visual Causality Analysis of Event Sequence Data

Zhuochen Jin, Shunan Guo, Nan Chen, Daniel Weiskopf, David Gotz, Nan Cao

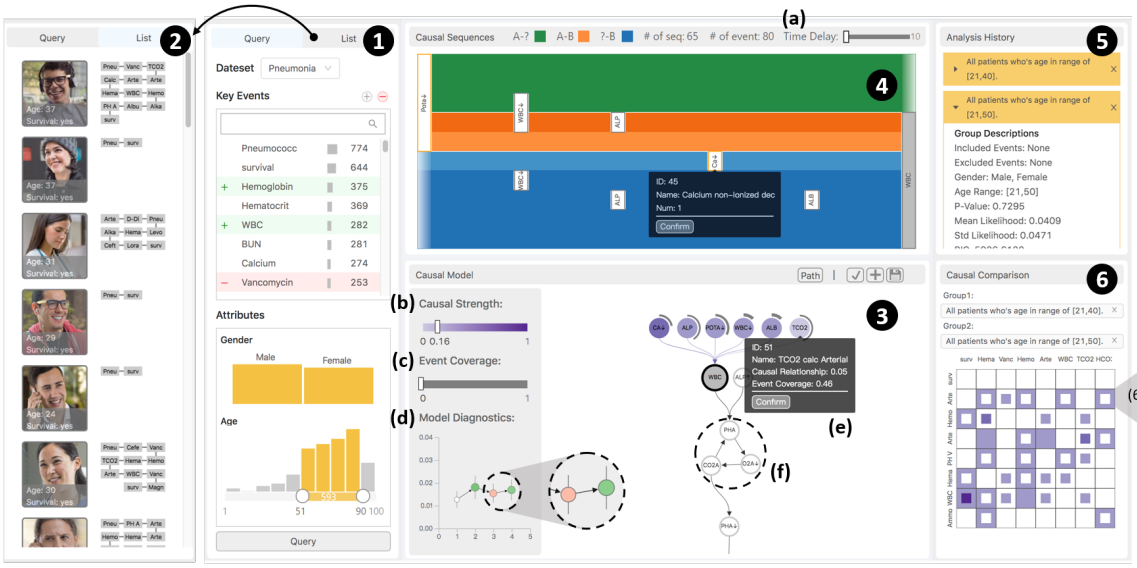


Fig. 1. An overview of the SeqCausal interface. The *query view* (1) provides a set of filters for the user to select sequences for analysis. The *sequence list view* (2) displays individual records retrieved from the query. The *causal model view* (3) displays the causal relations of events calculated from the back-end causality analysis model. Users can modify the graph, for example, confirm or delete a causal link, by examining causal relations from the *causal sequence view* (4), which summarizes causal patterns in raw event sequences. The *analysis history view* (5) stores causalities of different queried subsets, from which users can select any two items to compare their causal relations in the *causal comparison view* (6).

Abstract— Causality is crucial to understanding the mechanisms behind complex systems and making decisions that lead to intended outcomes. Event sequence data is widely collected from many real-world processes, such as electronic health records, web clickstreams, and financial transactions, which transmit a great deal of information reflecting the causal relations among event types. Unfortunately, recovering causalities from observational event sequences is challenging, as the heterogeneous and high-dimensional event variables are often connected to rather complex underlying event excitation mechanisms that are hard to infer from limited observations. Many existing automated causal analysis techniques suffer from poor explainability and fail to include an adequate amount of human knowledge. In this paper, we introduce a visual analytics method for recovering causalities in event sequence data. We extend the Granger causality analysis algorithm on Hawkes processes to incorporate user feedback into causal model refinement. The visualization system includes an interactive causal analysis framework that supports bottom-up causal exploration, iterative causal verification and refinement, and causal comparison through a set of novel visualizations and interactions. We report two forms of evaluation: a quantitative evaluation of the model improvements resulting from the user-feedback mechanism, and a qualitative evaluation through case studies in different application domains to demonstrate the usefulness of the system.

Index Terms—Event sequence data, causality analysis, visual analytics

1 Introduction

The recovery of underlying causality in observational data is one of the fundamental problems in science. Event sequences are widely collected in the form of a series of time-stamped events across a broad range of applications, such as electronic health records, financial transactions, and web clickstreams. The progression of individual sequences carries rich information on how events are mutually effected. Analyzing collections of temporal event sequences can help analysts extract cause–effect relationships between events, which may be beneficial to various analytical tasks, such as event forecasting and intervention planning. For example, in the medical domain, uncovering the causal relationships residing in sequences of medical records can help doctors understand critical symptoms that indicate a certain disease and promising treatment plans. In digital marketing, exploiting causal factors behind the increase and decrease in sales can provide insights into marketing strategies.

Although randomized controlled trials [6] are the gold standard for discovering causality, conducting such experiments is extremely difficult and costly. Therefore, causal analysis approaches have been developed for inferring causalities from modeling cause–effect relationships in observational data [34, 38, 41]. In temporal data, the discovery of causal relationships is commonly based on the theory of Granger causality [19], which is defined regarding the predictability and the temporal ordering of events. There is an extensive amount of research that

- *Zhuochen Jin, Shunan Guo, Nan Chen and Nan Cao are with iDV^x Lab at Tongji University. E-mail: {zcjin.idvx, g.shunan, christy05.chen, nan.cao}@gmail.com. Nan Cao is the corresponding author.*
- *Daniel Weiskopf is with University of Stuttgart. E-mail: weiskopf@visus.uni-stuttgart.de.*
- *David Gotz is with University of North Carolina at Chapel Hill. E-mail: gotz@unc.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

focuses on learning Granger causality in event sequence data, including those based on graphical modeling [31, 39], Hawkes processes [2, 51], and deep neural networks [37]. While these techniques have demonstrated their capabilities in identifying some reliable causal relations, many of them rely on rather general presumptions of the data distributions, which may fail to encode a sufficient amount of domain-specific knowledge [7, 30]. In addition, the high complexity of causal models can lead to a lack of sufficient interpretability and explainability to support decision-making.

Several recent studies have proposed analyzing causality through visual analytics, attempting to compensate for the deficiency of automatic causality analysis methods by bringing in human supervision [46, 47]. They utilize a set of visualizations and interactive tools to help human experts investigate and examine causal analysis results obtained from the model. However, these methods are mainly designed for non-temporal multivariate data with a limited number of variables. They are generally not applicable to temporal event sequences, as unique characteristics of event sequence data pose several special challenges. First, event sequence datasets often contain a large variety of event types [18, 21]. This high dimensionality of event sequence data can significantly increase the complexity of the causal analysis result. Second, sequences of various event types occurring in different orders lead to a high heterogeneity between individuals [22]. This hinders the extraction and summarization of common causal patterns in raw event sequences, resulting in difficulties in the interpretation and verification of the causality analysis results.

We introduce SeqCausal to address the aforementioned challenges: (1) the incorporation of human knowledge, (2) the lack of interpretability and explainability of automatic causality analysis, and (3) the temporal complexity specified in causality analysis of event sequence data. SeqCausal is an integrated visual analytics prototype designed for analyzing causalities in event sequence data. Concretely, we recover the Granger causality of events within a collection of event sequences based on Hawkes process modeling. To address the first challenge, we further enhance the causality analysis algorithm with a user-feedback mechanism that is able to leverage sparse corrections from human experts to update the entire causal model. Moreover, we introduce a set of visualizations and interactions for exploring, interpreting, and verifying complex causalities in high-dimensional and heterogeneous event sequences to address the second and third challenges. We quantitatively evaluate the ability of the user-feedback mechanism to improve the performance of automatic causality analysis, and present case studies to demonstrate the utility of our visual analytics prototype. The major contributions of this paper are as follow:

- **System.** We introduce an interactive visual analysis prototype that supports a workflow of exploration, verification, and comparison of causalities in event sequence datasets. To address the exploration difficulty introduced in the third challenge, the system integrates interactions for bottom-up exploration of complex causal graphs to enhance the efficiency in causal exploration. The system also enables users to interpret and examine the validity of causal relations from raw event sequences so as to meet the second challenge.
- **Algorithm.** We design a user-feedback mechanism to enhance the causality analysis algorithm in order to address the first challenge. It is able to transfer user corrections on the automatically generated causal relations to the causal model so that the model can be iteratively refined to better accord with users' domain knowledge.
- **Visualization.** We design a set of novel visualizations to display event causalities and summarize causal patterns in raw event sequences. This targets at resolving the difficulty in exploring and interpreting causalities in event sequences as introduced in the third challenge. In particular, we employ a causal graph to display causal relations and design a layout algorithm to better reveal causal structures (i.e., causality chains, circles). We also employ a flow-based visualization with an optimized layout for aggregating raw event sequences and showing how sequences progress among key events in the causal graph.

2 Related Work

Causal modeling is an active research area with extensive literature. Depending on the types of analyzed data, existing techniques can be broadly categorized into methods for independent and identically distributed (i.i.d.) data and non-i.i.d. data [20]. Temporal event sequences as a special type of non-i.i.d. data require distinct causal modeling algorithms that comply with the "temporal precedence" assumption [24]. In this section, we summarize prior research that is most relevant to our work, including causal modeling techniques specifically designed for event sequence data, and visual analysis techniques developed to facilitate causal analysis.

2.1 Causal Modeling for Event Sequences

The progression of successive time-stamped events can carry a great deal of information about the underlying causal mechanism. In this context, many approaches have been developed to recover the mutual causation of events, which mainly includes graphical modeling methods, Hawkes-based techniques, and deep learning approaches.

Graphical causal models, such as Peter & Clark (PC) and Functional Causal Model (FCM) [38, 43], are well-recognized causal discovery methods originally developed for non-temporal multivariate data. A number of papers attempt to extend typical graphical models to handle temporal data by incorporating an additional restriction on the temporal ordering of cause and effect. For example, TiMINo [39] and VAR-LiNGAM [25] enrich the causal equation in FCM with time lags of causal relationships. Similarly, PCMCI [40] and tsFCI [15] adapt typical conditional independence testing in the time-lagged correlation analysis. Graphical modeling techniques mostly require prior assumptions on the causal relationships, based on which the algorithm searches and verifies true causalities. In many domains, however, the lack of such assumptions and a large number of event types become serious impediments to the application of these methods.

Another direction of research is based on the theory of Hawkes processes [23], which corresponds to an autoregressive event sequence modeling technique that captures the self-excitation and mutual-excitation of events. This underlying principle of Hawkes processes has elicited a group of studies that attempt to recover causal relationships of events over a period of time from their intensity of influence inferred from the model. Eichler et al. [11] apply the concept of Granger causality to Hawkes processes using a least squares estimation of the impact function. Xu et al. [51] advance this technique with a set of regularizers to improve the robustness and computational complexity of the model. Unlike graphical-modeling-based methods, which merely estimate the causal relationships between events, Hawkes-based techniques are able to calculate the change of causal strength within each pair of events over time. This information may integrate more causal semantics into the analysis context and result in more interpretable discoveries.

With deep learning techniques gaining popularity, some recent causal discovery methods attempt to leverage the capabilities of deep neural networks in capturing complex event dependencies. For example, Zhang et al. [52] utilize the neural point processes [35] based on recurrent neural networks in place of Hawkes processes in causal discovery. Nauta et al. [37] discover causal relationships and causal delays in temporal data with an attention-based convolutional neural network. Although deep learning approaches generally achieve better accuracy and scalability than graphical modeling algorithms and Hawkes-based causal discovery algorithms, the lack of interpretability poses a great problem for understanding and justifying causal relationships.

To balance between the informativeness and interpretability of the analysis result, in this paper, we base our work on the state-of-art Granger causality analysis algorithm based on Hawkes processes [51]. In particular, we extend this earlier work to accommodate interactive visual analysis through a user-feedback mechanism that takes users' modifications of the initial causal relationships and updates the model accordingly. Together with the interactive visual interface, our method leads to more accurate and comprehensive causal findings.

2.2 Visual Causality Analysis

A wide variety of methods have been developed to visualize causality in data analysis. Traditional visualizations, such as directed acyclic

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.3030465, IEEE Transactions on Visualization and Computer Graphics

graph (DAG) layouts and Hasse diagrams, can be employed to illustrate causality to a certain extent. However, they become inefficient as an increasing number of variables may introduce more edge crossings. Elmqvist et al. propose two visual methods, Growing Squares [14] and Growing Polygons [13], which enhance node representations in DAGs with color-coded squares and polygons to provide an overview of influences on each event. They also leverage animation to present the temporal ordering of causality. Although both methods are effective in uncovering the causal structure of events, they fail to integrate causal semantics into the graph, which is important for a deeper understanding of the causal relationships. To incorporate additional causal semantics, Kabada et al. [27] introduce a set of animations following Michotte's rules of causal perceptions [36] to intuitively illustrate causal strength, amplification, dampening, and multiplicity. Recent studies put more effort into integrating automatic causal analysis algorithms and causality visualizations into a visual analytics system to facilitate interactive causal analysis and reasoning. Chen et al.'s [7] visual causal analysis system aims to provide hypothesis generation and evaluation and support decision-making, which leads to a number of visual analytics systems designed to support interactive analysis of data correlation and causation. For example, Zhang et al. [53] utilize a force-directed graph layout to present the correlation between numerical and categorical variables in multivariate data. ReactionFlow [8] aims to support a better understanding of causal relationships between proteins and biochemical reactions in biological pathways. It organizes the causal pathways into a flow-based structure to emphasize the downstream and upstream of the causal relationships. To include domain knowledge, Wang and Mueller [46] present an interactive visual interface that allows analysts to edit and verify causal links according to their domain expertise. They further extend this work with a path diagram visualization to better expose causal sequences of the variables [47].

Despite the extensive visual analytics approaches for analyzing causalities, most of the existing techniques focus on non-temporal multivariate data and methods for analyzing causal relationships in temporal event sequences still remain deficient. Most relevant to our work is the visual analytics framework introduced by Lu et al. [32], which annotates critical changes in media topic volume with causalities of media events. Our work focuses on extracting accurate causal relationships between events from a general event sequence dataset and assisting analysts in making interpretable causal discoveries.

3 Requirement Analysis and Approach Overview

Prior to the development of SeqCausal, we had a thorough discussion with experts in the medical domain on the specific analytical tasks and challenges of analyzing causalities in electronic health records. By including design principles from previous visual causality analysis techniques, we identify a set of design requirements:

- **R1. Extract key events and subgroups for analysis.** Real-world event sequence datasets usually contain a large number of divergent progression patterns and irrelevant event types, which may introduce a lot of noise in causal analysis. The system should, therefore, allow users to query sequence subsets that follow a similar progression context and select key event types to ensure the performance of the causal model.
- **R2. Ensure efficiency in causal exploration.** The high dimensionality of event sequence data can result in a large and complex causal graph that is hard to investigate as a whole. To cope with this issue, the system should incorporate interactive approaches to improve efficiency in exploring causalities.
- **R3. Enhance the interpretability of causal relationships.** The lack of interpretability is an inherent issue of machine learning models, which also exists in the context of causal analysis [44]. The system should, therefore, provide explanations from underlying data by demonstrating corresponding sequences in the dataset that follow particular causal patterns.
- **R4. Support identification of spurious causalities.** The theory of Granger causality exploits the association of event variables under the restriction of temporal precedence [19]. However, temporal precedence alone is sometimes not sufficient for establishing true cause–effect relationships [10]. Hence, the system should support identifying spurious causalities from the causality analysis result.

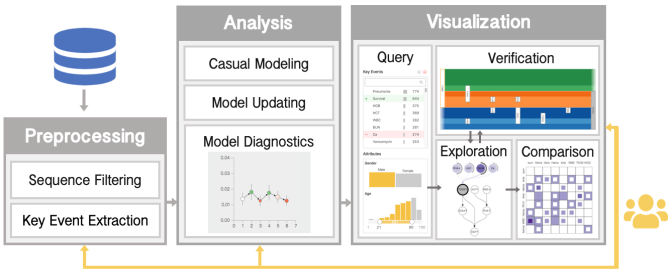


Fig. 2. The framework of the SeqCausal system, including a data preprocessing module, a causal analysis module, and a visualization module.

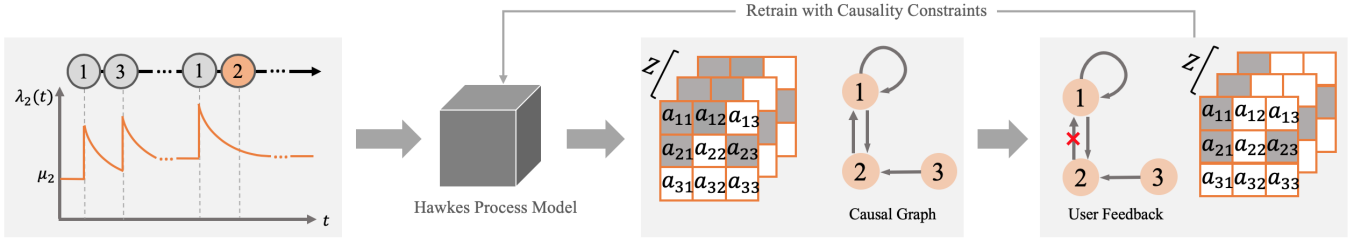
- **R5. Incorporate human knowledge in the causal model refinement.** Automatic causal analysis algorithms are generally not capable of including an adequate amount of human knowledge [7]. For example, doctors are required to follow medical guidelines that contain verified causalities and restrictions of medical treatments that are not included in the model assumption. Consequently, the system should allow users to modify the model output and incorporate user feedback into model refinement.
- **R6. Provide diagnostic measures on model quality.** Bringing human supervision into causal model refinement may introduce user biases in the analysis result [45]. To guard against the potential negative effect of the causal model from biases, it is necessary to support objective model diagnostics mechanisms to guide user refinements on the model output.
- **R7. Allow comparison of causalities for different subgroups.** Causal relations inferred from different groups of sequences can vary dramatically. For example, in the medical scenario, patients may have different applicable medicines due to different symptoms. Comparing causalities of different cohorts can help doctors make personalized treatment plans. To this end, the system should allow comparing causalities in different subgroups of sequences.

Guided by the above design requirements, we developed SeqCausal, a web-based visual analytics system for recovering causalities in general event sequence datasets. SeqCausal uses the open-sourced JavaScript framework *React*. The front-end functionality is achieved by *D3.js*. The back-end causality analysis algorithm is implemented with Python. The framework of the system is illustrated in Fig. 2. The data preprocessing module is equipped with an efficient query mechanism that allows users to filter a subset of sequences fitting certain criteria and key events of interest to build a causal model (**R1**). The causal analysis module is primarily responsible for extracting causal relationships between events from the preprocessed dataset and further delivering to the visualization module for visual causality analysis. In addition, the causal analysis module provides a user-feedback mechanism that integrates modifications from users to update the causal model (**R5**) with the underlying model diagnostics guaranteeing model quality upon each iteration (**R6**). The visualization module supports the following functionality: 1) causal exploration, which supports user-driven investigations and edits of the model output (**R2**), 2) causal verification, which summarizes causal patterns in original sequences to help with causal interpretation (**R3**) and guide model refinement (**R4, R5**), and 3) causal comparison, which allows users to compare causalities of different queries (**R7**).

4 Causal Discovery from Event Sequences

This section introduces the causality analysis algorithm for extracting Granger causality from event sequence datasets. Figure 3 gives an overview of the causal analysis pipeline, which is composed of three key steps (Fig. 3(a–c)). First, we employ Hawkes processes to model Granger causalities in event sequences using Xu et al's technique [51]. Then, we train the model to fit the data by maximizing the likelihood. The parameters of the trained model are utilized to infer Granger causalities of event types. Lastly, we enhance the interactivity

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.3030465, IEEE Transactions on Visualization and Computer Graphics



(a) Event Sequence Modeling (b) Inferring Granger Causality (c) Refining Model with User Feedback

Fig. 3. The pipeline of causality analysis algorithm. The algorithm consists of three key steps: (a) training a Hawkes process model to fit the event sequence dataset, (b) inferring impact factors $a_{vv'}$ between two events from the trained model to generate initial causal relations, and (c) incorporating users' modifications to retrain the model with a constrained objective.

of the model with a user-feedback mechanism (**R5**) that incorporates human knowledge for model refinement.

4.1 Background of Hawkes Processes

Granger causality is capable of characterizing causality in temporal data according to incremental predictability: if the occurrence of an event B enhances the predictability of an event A, event B Granger causes event A. Hawkes processes [23] build a statistical model that describes the triggering patterns between events. The assumption behind Hawkes processes is similar to the theory of Granger causality in the context of event sequence data, which presumes that the occurrence of an event may increase the probability of occurring another event in the near future [35]. This consilience makes Hawkes processes particularly suitable for learning Granger causality in event sequences. Therefore, in the first step, we leverage the Granger causality analysis method of Hawkes processes proposed by Xu et al. [51] to establish our analysis model. Given a collection of event sequences with V types of events, the occurrence probability of event $v \in \{1,\dots,V\}$ at time t can be inferred from its conditional intensity (i.e., number of event occurrences per time unit), $\lambda_v(t)$, which is formally defined as:

$$\lambda_v(t) = \mu_v + \sum_{v'=1}^{V} \int_0^t \phi_{vv'}(r)\, \mathrm{d}N_{v'}(t-r) \tag{1}$$

where the first term μ_v is a constant baseline intensity, and the second term indicates the increase of intensity brought by the excitation of all historical events on event v. Specifically, $N_{v'}(t-r)$ is the number of events v' before time $t-r$, and $\phi_{vv'}(r)$ is a time-varying impact function that captures the influence of historical event v' on event v:

$$\phi_{vv'}(t) = \sum_{z=1}^{Z} a_{vv'}^{z}\, \kappa_z(t) \tag{2}$$

In particular, the impact function incorporates a linear combination of a set of Z Gaussian sampling functions $\{\kappa_z(t)\}_{z=1,\dots,Z}$ to simulate the decaying influence of event v' on event v. Z is set as the minimum value according to Silverman's rule of thumb [42], which limits the maximum brandwidth for Gaussian sampling that is calculated based on the time duration between two events and the number of times that two events co-occur in one sequence. $\boldsymbol{a}_{vv'} = [a_{vv'}^1,\dots,a_{vv'}^Z]^\top$ are the impact coefficients, which indicate the level of stimulation effect caused by event v' on event v.

4.2 Learning Granger Causality

In the second step, we search over the parameter space to fit the Hawkes processes to the sequence collection. The parameters include the base intensity of all events $\boldsymbol{\mu} = [\mu_v]_{v=1,\dots,V} \in \mathbb{R}^V$ and the impact coefficients $\boldsymbol{a} = [a_{vv'}^z]_{v,v'=1,\dots,V}^{z=1,\dots,Z} \in \mathbb{R}^{V\times V\times Z}$ for each pair of events (v,v'). We formulate the training objective as follows:

$$\operatorname{argmin}_{\boldsymbol{\mu},\boldsymbol{a}} \quad -L + \alpha \sum_{v,v'} \|\boldsymbol{a}_{vv'}\|_2 \tag{3}$$

where the first term is the negative log-likelihood of the Hawkes process on the sequence dataset [23]. Given a collection of event sequences $S = \{s_i\}_{i=1,\dots,I}$, where each sequence $s_i = \{(v_m^i, t_m^i)\}_{m=1,\dots,M_i}$ is a series of M_i event–time pairs with $v_m^i \in \{1,\dots,V\}$ and t_m^i representing the event type and timestamp of the m-th event respectively, the log-likelihood L can be expressed as follows:

$$L = \sum_{i=1}^{I} \left\{ \sum_{m=1}^{M_i} \log \lambda_{v_m^i}\left(t_m^i\right) - \sum_{v=1}^{V} \int_0^{T_i} \lambda_v(r)\, dr \right\} \tag{4}$$

The second term of the training objective is a group-lasso regularizer which ensures that the inferred impact coefficients are interpretable by the theorem of Eichler et al. [12]. According to the theorem, the Granger causality between event types can be directly inferred from the impact coefficients $\boldsymbol{a}_{vv'} = [a_{vv'}^1,\dots,a_{vv'}^Z]^\top$, and events v and v' have no causal relationship only if $a_{vv'}^z = 0$ for all $z \in \{1,\dots,Z\}$. The hyperparameter α controls the influence of the regularization term.

The objective function is optimized by applying an EM-based algorithm [29], and the learning result $\boldsymbol{a} = [a_{vv'}^z]_{v,v'=1,\dots,V}^{z=1,\dots,Z}$ captures the causal relationships and causal strengths between events. We define the causal strength of the causality $v' \to v$ as follow:

$$\text{Strength}_{vv'} = \frac{1}{Z} \sum_{z=1}^{Z} a_{vv'}^z \tag{5}$$

Thus, we can obtain a directed causal graph $G(\mathscr{V},\mathscr{E})$ whose edges $\mathscr{E} = \{v' \to v\}$ are weighted by the causal strength $\text{Strength}_{vv'}$.

4.3 Updating Causality with User Feedback

To incorporate human knowledge in causality analysis **(R5)**, we further designed a user-feedback mechanism that is able to make refinements on the model according to user inputs. In particular, the user can modify the causal graph from the visual interface by preserving authentic causal relations and deleting spurious ones. Based on the user's modifications, a new causal graph $\hat{G}(\hat{\mathscr{V}},\hat{\mathscr{E}})$ is generated, and the model can be updated automatically by optimizing a new objective function:

$$\begin{aligned} &\operatorname{argmin}_{\boldsymbol{\mu},\boldsymbol{a}} \quad -L + \alpha_u \sum_{v,v'} \left\| \boldsymbol{a}_{vv'(\hat{G})} \right\|_2 \\ &\text{s.t.} \quad \boldsymbol{a}_{vv'} = \boldsymbol{0} \quad \text{for} \quad (v' \to v) \notin \hat{G} \end{aligned} \tag{6}$$

where L is the log-likelihood of Hawkes process, α_u is the control hyperparameter, and $\sum_{v,v'} \left\| \boldsymbol{a}_{vv'(\hat{G})} \right\|_2$ is the user-specified regularizer:

$$\boldsymbol{a}_{vv'(\hat{G})} = \begin{cases} \boldsymbol{0}; & \text{if } (v' \to v) \text{ is confirmed} \\ \boldsymbol{a}_{vv'}; & \text{otherwise} \end{cases} \tag{7}$$

Specifically, if a causal relation is removed by the user, the constraint of Equation (6) ensures that the model parameters are optimized toward setting the corresponding impact factor as 0, so that the updated causal model aligns with user feedback. If a causal relation is confirmed by the user, the updates of the corresponding impact factor can be liberated from the group-lasso regularizer according to Equation (7). This aims to prevent the impact factor from being set as 0 by the regularizer. After refining the model, the causal graph will be redrawn based on the updated parameters $\boldsymbol{a} = [a_{vv'}^z]_{v,v'=1,\dots,V}^{z=1,\dots,Z} \in \mathbb{R}^{V\times V\times Z}$. The user can investigate the updated causalities and iteratively make modifications until the analysis result is satisfactory.

Computational complexity. The computational complexity of the causality analysis algorithm is $O(ZV^2n^3)$ per training iteration, which is the same for both the initial computation and the update of the causality on user-feedback. It depends on three data attributes: the number of the sampling functions Z, the number of event types V, and the number of

occurrences of all events in the dataset n. We implemented the causality analysis model with Python using the NumPy package, which is able to compute the causality between events in parallel. Running times under different data sizes are reported in Section 7.

5 Visual Causality Analysis

In this section, we first introduce the main components in the SeqCausal interface. Then, we describe the details for each system functionality.

5.1 User Interface

The user interface of SeqCausal is composed of six key views. The left panel includes the *query view* (Fig. 1(1)), allowing the user to select a dataset and filter sequences from the database for analysis (**R1**), and a *sequence list view* (Fig. 1(2)), showing the profile of each individual sequence determined by the query result.

Views in the middle are designed to support causal exploration and verification. The *causal model view* (Fig. 1(3)) suggests potential causal relationships using a node-link causal graph, allowing users to investigate the causalities and make updates on the causal model after verifying the causal relations (**R2, R4, R5**). The *causal sequence view* (Fig. 1(4)) facilitates causal verification by showing the causal patterns in raw event sequences using a flow-based visualization. We separate these two functionality of our system into two visualizations so that the user can turn to raw event sequences only when needed. The user can verify causal relations according to their domain expertise without investigating the *causal sequence view*, or leverage aids from raw event sequences to assist causal reasoning. In addition, two different hierarchical layouts for the *causal model view* and the *causal sequence view*, emphasizing causal structures (i.e., causality chains and circles) and temporal ordering of events, respectively. To guide iterative updates of the causal graph, the *model diagnostics panel* (Fig. 1(d)) shows incremental changes of the model quality (**R6**).

Views on the right include the *analysis history view* (Fig. 1(5)), which stores user queries and the causal analysis result of the corresponding sequence subdivision from which users can select two subgroups for comparison and the *causal comparison view* (Fig. 1(6)), showing the differences between causal graphs inferred from two subgroups of sequences through a matrix-based visualization (**R7**).

5.2 Causal Exploration

Real-world event sequence datasets are generally large and heterogeneous, containing many event types. This characteristic can lead to great challenges in visualizing and exploring complex event causalities. Therefore, we designed our system to enable flexible data selection, display causalities with intuitive visualizations, and provide efficient interactions to allow exploring causalities incrementally.

Select sequences for analysis (R1). The *query view* for filtering homogeneous subsets from a large collection of event sequences ensures a high quality of causality analysis. The user can choose a dataset from the drop-down list and filter sequences based on the occurrence of key events and the attribute of records. The *key events panel* displays the list of all event types in the dataset, which allows users to determine event-based query criteria. For example, a doctor may need to filter patients diagnosed with certain diseases or taking specific medicines for analysis. The event types are ranked by the coverage rate (i.e., the proportion of sequences containing each event), which is visually encoded with the length of a horizontal bar, and the exact number of sequences being covered is labeled at the right. An event search is also provided to help the user navigate the event list. By switching between ⊕ and ⊖ in the *query view*, the user can select events by inclusion criteria or exclusion criteria. The selected events are highlighted with green and red background correspondingly. When executing the query, the system only retrieves sequences that contain all events in the inclusion criteria and no events in the exclusion criteria. The *attributes panel* shows the distribution of records on metadata (e.g., gender and age), allowing the user to filter sequences based on non-temporal attributes. To reduce the negative influence of heterogeneous sequence progression on the performance of the causality analysis, the system further clusters sequences by their progression similarities using the measure proposed by Wongsuphaswat et al. [49], and retrieve a major cluster of sequences for analysis.

Visualize and explore causal relationships (R2). After querying the dataset, the causal analysis module generates the causal relationships of all event types in the dataset. The causal relationship is demonstrated in a node-link causal graph, with nodes representing event types and links representing causal relationships pointing from the cause to the effect. To support investigating causal relations in the graph, we design a layout algorithm to reduce the visual complexity of the graph when facing a large number of event types and complex causal structures. As suggested in a previous study [4], laying out the causal graph in sequential order can facilitate the searching process of root cause and derived effects. Following this finding, we choose to visualize the causal graph using a top-bottom sequential layout. However, causal structures in event sequence data can be more complex than sequential chains of cause–effect relations. It may also include causal circles (e.g., (Fig. 1(f))) or self-exciting causalities that cannot be satisfied by simply applying a sequential layout. Therefore, we devise a layout algorithm that calculates the position $\mathbf{p}_i = (x_i, y_i)$ for each node $i \in \{1, \ldots, V\}$ to better illustrate the local structures (i.e., causal chains and circles) in the causal graph. Algorithm 1 gives an overview of the key steps in the layout algorithm, which is detailed as follows.

Algorithm 1 Causal Graph Layout Algorithm

Input: The directed causal graph $G(\mathscr{V}, \mathscr{E})$
Output: A position $\mathbf{p}_i = (x_i, y_i)$ for each vertex i of $\mathscr{V}$

1: Detect circles $C = \{C_n(\mathscr{V}_{c_n}, \mathscr{E}_{c_n})\}_{n=1,\ldots,N_c}$ by depth-first search
2: Traverse G by breadth-first search and calculate node depth d_i for each vertex i
3: **for** $i \in \mathscr{V}$ **do**
4: Calculate $y_i = (CanvasHeight/MaxDepth) \times d_i$
5: Update $\mathbf{p}_i$ for all vertices by minimizing Equation (8)
6: Update x_i for all vertices to remove node overlap

Given the causal graph $G(\mathscr{V}, \mathscr{E})$, we first use depth-first search to detect causality circles $C = \{C_n(\mathscr{V}_n^c, \mathscr{E}_n^c)\}_{n=1}^{N_c}$ in the graph. To prevent endless loops when iterating over the graph, we take nodes in the same causality circle as one unit. Then, we transform the causal graph into a minimum spanning tree by traversing the graph using breadth-first search. Each node or causality circle is assigned with a depth d_i indicating their level of hierarchy. Nodes within a circle unit are assigned with node depths same as the entrance node of the circle. Based on the depths, we determine the y-position for each node and causality circle by fitting the hierarchy into the canvas in a top-to-bottom manner.

To minimize edge crossings between adjacent levels of the hierarchy, we further arrange the position of nodes by minimizing the following objective function:

$$J(x,y) = S(x) + \sum_{n=1}^{N_c} \sum_{(v \to v') \in \mathscr{E}_n^c} w_{vv'}^c \left\| \mathbf{M}_n (\mathbf{p}_v - \mathbf{p}_{v'}) - (\mathbf{q}_{nv} - \mathbf{q}_{nv'}) \right\|^2 \quad (8)$$

The first term is the stress function of Stress Majorization [16] we employ to minimize edge crossings as follows:

$$S(x) = \sum_{i<j} w_{ij} \left(\left\| x_i - x_j \right\| - d_{ij} \right)^2 \quad (9)$$

where x_i is the x-position of i-th node, d_{ij} represents the graph-theoretical distance between nodes i and j, and $w_{ij} = d_{ij}^{-2}$ is the normalization constant that prioritizes nodes with small distance. The second term in the training objective is a circular constraint for regularizing the shape of causality circles. N_c is the number of circular sub-graphs in the causal graph. The positions of nodes in the n-th circle, $\{\mathbf{p}_v\}_{v \in \mathscr{V}_n^c}$, are calculated through an affine transformation $\mathbf{M}_n$ that matches the nodes in the n-th circle to a reference shape with equal number of vertices positioned at $\{\mathbf{q}_{nv}\}_{v \in \mathscr{V}_n^c}$. The optimization goal of the second term is to minimize the difference of distances for any two vertices in the causality circle (i.e., $\mathbf{M}_n(\mathbf{p}_v - \mathbf{p}_{v'})$) and the corresponding vertices in the reference shape (i.e., $\mathbf{q}_{nv} - \mathbf{q}_{nv'}$). Similar to the Stress Majorization, the second term also employs a normaliza-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.3030465, IEEE Transactions on Visualization and Computer Graphics

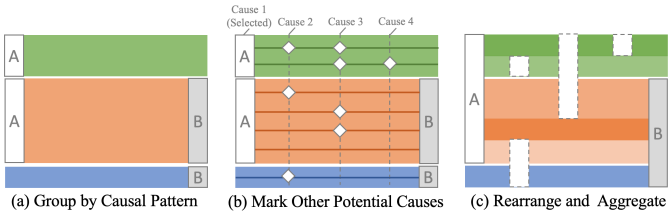


Fig. 4. The *causal sequence view* displays causal patterns in raw event sequences related to a selected cause–effect relation (e.g., A→B). (a) Event sequences are divided into three groups based on the presence of the causes and effects. (b) All other potential causes are vertically aligned and arranged from left to right in temporal order, and then (c) aggregated to help users examine the validity of the selected cause–effect relation and suggest other possible causalities.

tion constant $w^c_{vv'} = \|\mathbf{M}_n(\mathbf{p}_v - \mathbf{p}_{v'})\|^{-2}$ to prioritize closer nodes in the causality circle. The transformation matrix for each causality circle $\mathbf{M}_n$ is calculated by adapting the circular constraint in Wang et al. [48] defined as follows:

$$\text{argmin}_{\mathbf{M}_n} \sum_{i \in \mathcal{V}^c_n} \omega_i \|\mathbf{M}_n \mathbf{p}_i - \mathbf{q}_{ni}\|^2 \tag{10}$$

where ω_i is set to the degree of vertex i for illustrating the circular structure more clearly. In the final step, we tweak node positions on the x-axis to remove overlaps [9]. The layout algorithm runs at the back-end of our system. The objective function is implemented in Python and optimized using NetworkX and SciPy.

The complete causal graph of an event sequence dataset can be large and complex given a large number of event types. To relieve the user from investigating and verifying many cause–effect relations at a time (**R2**), SeqCausal incorporates a user-driven causal exploration procedure that allows the user to focus on investigating causal relations related to an outcome event incrementally from the bottom to the top. The intention is to uncover only causal pathways that lead to an outcome event of interest to reduce the number of events and causal relations evolved, and eliminate invalid chains of causal relations before branching out. This procedure starts by adding an outcome event as an initial effect using ⊕ at the top of the *causal model view*. By double-clicking on the effect, the graph expands one layer at the top to show the direct causes suggested by the causal model. As shown in Fig. 1(3), the effect under inspection is colored in gray and the causes are colored by their causal strength. In addition, an outer ring is displayed on each node, and the length represents event coverage (i.e., the proportion of raw sequences that have the cause and effect events appear successively). The user can filter causal relations by causal strength (Fig. 1(b)) and event coverage (Fig. 1(c)). To explore causal chains of the outcome event, the user can continue to expand the graph by iteratively uncovering causes of the topmost events, and stop when the causal chain for the outcome event at the bottom is completed (i.e., no new event is added to the graph). In each iteration, we primarily highlight the newly involved events and causal relations to guide the user's attention toward inspecting them. The user can switch to an overview of the entire graph at any time by clicking on the background of the *causal model view*.

5.3 Causal Verification

To build a causal model that conforms to objective rules and the user's domain knowledge, SeqCausal allows the user to verify the causal relations suggested by the algorithm and update the causal model. In particular, SeqCausal displays causal patterns in raw event sequences to support the user in interpreting the causality (**R3**) and identifying invalid causal relations (**R4**). After verifying the causality, the user can modify the causal graph, for example, delete mistaken or add omitted causal relationships, to update the causal model (**R5**). Real-time model diagnostics are provided concurrently with user modification to ensure high model quality (**R6**). In the following, we introduce the system functionality designed to support causal verification and modification.

Visualize causal patterns in raw sequences (R3, R4). To help users interpret and examine the validity of causality analysis results, we associate the calculated causal relations with original data by uncovering the causal patterns in raw event sequences. The user can select a causal relation (e.g., event **A** causes event **B**) from the *causal model view* and observe how the sequences progress through the causes and effects from the *causal sequence view*. In particular, there are three categories of relative causal patterns: sequences that go through the cause but never come across the effect afterward (**A→?**), sequences that contain both the cause and effect in successive order (**A→B**), and sequences that have the effect but not the cause before (**?→B**). To distinguish these situations, we categorize raw sequences into three groups using a flow-based visualization (Fig. 4(a)). The leftmost and rightmost nodes indicate the cause and effect event colored in white and gray, respectively. The edges between nodes indicate groups of subsequences, colored by the sequence categories. The height of nodes and edges is proportional to the number of sequences in the group. We allow users to control causal delays in the subsequences by setting the length of subsequences with a *time delay slider* (Fig. 1(a)).

The system also displays other potential causes suggested by the causal graph on the subsequences to help the user explore other possible causal relations (**R4**) and justify the validity of the selected causal relation. For example, in Fig. 1(4), the raw event sequences are categorized into three groups according to a selected causal relation: *POTA →WBC*. To facilitate comparing the selected cause with other potential causes, the system also marked the occurrence of other direct causes besides *POTA* on the subsequences. Specifically, we first mark each potential cause in each individual subsequence. As shown in Fig. 4(b), each line represents an individual subsequence, and the potential causes are anchored in the subsequences. The causes are vertically aligned and horizontally ordered from left to right by their average time of occurrence. Note that we only display one time of occurrence for each cause, as the frequency of occurrence does not affect the validity of causality. In order to simplify the visualization and make it easier to observe the commonness in the occurrence of potential causes, we further reorder subsequences within each category and aggregate common potential causes in adjacent subsequences (as shown in Fig. 4(c)). Causes in adjacent subsequences of different categories are also aggregated to further reduce the number of intermediate nodes. In particular, we leverage the reordering algorithm as follows.

We first calculate the pair-wise similarity of two subsequences S_i and S_j as follows:

$$d(S_i, S_j) = \left\|\mathbf{w} \odot (\mathbf{v}_i - \mathbf{v}_j)\right\| \quad \mathbf{w}, \mathbf{v}_i, \mathbf{v}_j \in \mathbb{R}^n$$

where n is the number of potential causes. $\mathbf{v}_i$ and $\mathbf{v}_j$ are the one-hot vectors representing the occurrence of potential causes in subsequences S_i and S_j. $v_{i,k} = 1$ if the k-th potential cause appears in S_i, otherwise $v_{i,k} = 0$. $\mathbf{w}$ is a constant vector that represents the coverage rate of each potential cause, which prioritizes the aggregation of events that occur in most subsequences. Then, we abstract the sequence ordering problem into a Traveling Salesman Problem (TSP), in which the concept of cities and distances represent subsequences and the pair-wise similarity, respectively. We utilize simulated annealing [17] to search for an accessing order with approximately minimal cost. In this way, subsequences in close proximity can have more potential causes in common that can be aggregated.

Our system supports the user to switch among potential causes by either clicking on the nodes in the *causal model view* or the *causal sequence view*. On switching to another potential cause, the corresponding node in the *causal sequence view* will move to the left end with a smooth transition, and the rest of the view will be updated accordingly. The user can also select a causal path (i.e., chains of cause–effect events) by clicking Path in the *causal model view*, and successively select a series of cause–effect events to emphasize the causal path on the graph. The *causal sequence view* will be updated with the events in the causal path arranged from left to right, and edges showing the progression pattern of sequences flow through the causal path.

Verify and modify causalities (R4, R5). The user can determine whether a causal relationship holds true according to the observations in the *causal sequence view* or based on their domain expertise. For example, if the sequences contain large numbers of "A→?" and "?→B"

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.3030465, IEEE Transactions on Visualization and Computer Graphics

patterns or mostly go through another potential cause, the direct causal relation is not likely to be true. Moreover, we measure the probability that the selected causal relation is valid on a particular subgroup of sequences by calculating the regression likelihood. This probability is encoded by the color saturation of edges, and edges with deeper colors indicate the selected causal relationships generally fit better to the group of sequences. After investigating this statistical information and incorporating the domain knowledge, the user can determine whether the causal relationship holds true and eliminate spurious causalities. By clicking Confirm in the tooltip (Fig. 1(e)), a causal relation is confirmed and the corresponding cause event will be colored in gray. After the users finish confirming the causal relationships, they may update the causal analysis model by clicking ✓. In response, the causality analysis model will be retrained with the user's feedback of the confirmed causal relations and update the causal graph with the regenerated causality analysis result. The layout of the causal graph is recomputed following Algorithm 1. To make it easier to track nodes in the causal graph before and after the update, we add a stabilization constraint $\sum_i \left\|x_i - x_i'\right\|^2$ to the original training objective (Equation (8)) when performing graph updates. The stabilization constraint iterates over common nodes of the causal graph before and after the update and minimizes their change in x-position. After updating the causal graph, the user can either continue to explore the causes for topmost confirmed nodes by double-clicking them or save the final causal analysis result for the queried sequences to the *analysis history view* by clicking in the *causal model view*.

Diagnose the causal model (R6). Every time the user updates the causal analysis model, the *model diagnostics panel* (Fig. 1(d)) records the change of the overall model quality. This aims to help the user determine the number of iterations to update the graph, which may vary between datasets according to their causal complexity (e.g., lengths of the causal chains and the number of event types). In general, the user can choose to stop adding more iterations when the model shows no significant improvement. The performance of the model is evaluated by three metrics: the regression likelihood of all causal relationships on the queried data, Bayesian Information Criterion (BIC) [5], and p-value. The regression likelihood indicates the model goodness of fit, and BIC estimates the complexity of the causal model to ensure better generalization capabilities. The p-value evaluates the significance of improvement between two model updates. The circles are positioned in a two-dimensional space defined by the number of model updates on the x-axis and the mean regression likelihood on the y-axis. The error bar represents the standard deviation of the regression likelihood. The color of the circles encodes the change of the BIC score in comparison to the previous model. Green circles represent the better generalization capability and red circles represent the worse. The detailed values of these metrics are displayed in a tooltip activated when the user hovers the mouse over a circle. When the performance of the model declines, the user can revert the causal model and causal graph to a previous savepoint by clicking on the circles.

5.4 Causal Comparison

SeqCausal also supports comparing causalities of different sequence subgroups (**R7**) in the *causal comparison view* (Fig. 1(6)). The user can leverage the comparison result to characterize different groups of sequence entities and make customized decisions accordingly. For example, in medical cases, treatment may cause the cure of a disease for one group of patients but not the other. This can be reflected in the difference between the corresponding causal relations. As mentioned in Section 5.3, the user can save the final causal analysis result of the queried dataset to the *analysis history view*. In this view, each item shows a general description of the analyzed dataset according to its querying condition. The detailed descriptions, including the user's editing history, statistics on model performance, and the causal graph can be retrieved by expanding the item.

The user can drag any two items from the *analysis history view* into the *causal comparison view* to compare the causal relations in different subgroups. We utilize a superimposed adjacency matrix to visualize the occurrence of all causal relations in two groups. The rows of the matrix represent causes and the columns represent effects. The encoding of each cell shows the existence of a causal relation in two subgroups. As illustrated in Fig. 1(6), each cell is divided into an outer region and an inner region, with the background color saturation representing the causal strength of the corresponding causal relation in the first group and the second group, respectively. The encoding of the cell can distinguish a total of five situations of the comparison result (shown in Fig. 1(6a–6e), respectively): (a) the causal relation only exists in the first group, (b) the causal relation only exists in the second group, (c) the causal relation exists in both groups but with different causal strength, (d) the causal relation exists in both groups and has the same causal strength, (e) the causal relation does not exist in both groups. In this case, the user can quickly detect the causal relations that have a significant difference in two groups of sequences.

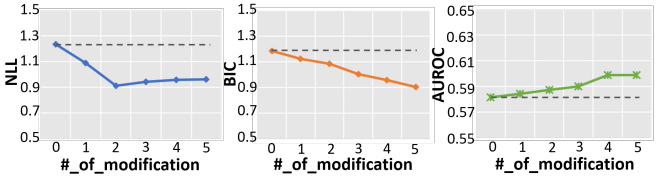


Fig. 5. The performance of the user-feedback mechanism under three metrics: negative log-likelihood (blue), Bayesian Information Criterion (orange), and Area Under ROC (green). The dashed lines show the original model performance on the respective metric without user-feedback.

6 Evaluation

We assess the usefulness of SeqCausal through two forms of evaluations: a quantitative study showing the effectiveness of the user-feedback mechanism incorporated in the analysis algorithm, and qualitative case studies demonstrating the usefulness of SeqCausal system. In the quantitative study, we used a public news media dataset, MemeTracker [28], which has the ground-truth causality for us to measure the accuracy of the causality analysis result. For the qualitative study, we applied two datasets for distinct applications: a public-access intensive care dataset, MIMIC [26], and a media dataset [1] that captures users' commenting trajectory on Reddit. These datasets, however, do not contain ground-truth causality. Therefore, we leverage human knowledge to justify the causality analysis results. In this section, we report our study findings and discuss feedback from study participants.

6.1 Performance of User-Feedback Mechanism

We employ MemeTracker dataset to evaluate the effectiveness of the user-feedback mechanism, which contains time-stamped phrases and hyperlinks for news articles and blog posts from mainstream media sites. Each sequence records the trace of a meme (i.e., a representative quote or a phrase) across various websites. Each event represents an occurrence of a meme on a website, and the website represents the event type. We filter 20 event types of the most active websites and sequence records from August 2008 to September 2008 to train the causality analysis model, and generate causal relations among websites to imply the spreading patterns of memes. The ground-truth causality was provided by whether one website contains the hyperlink linking to another site [2, 50]. In each iteration, we stimulated the user feedback by confirming one causal relation in the ground-truth set to update the model. According to the *model diagnostics panel*, the performance of the model starts to stabilize after five iterations. We report the performance changes in the first five iterations from two aspects: the goodness-of-fit and model accuracy.

Goodness-of-fit. We utilize the negative log-likelihood (NLL) and the BIC score to examine the effect of the user-feedback mechanism on the goodness-of-fit of the causality analysis model. In particular, a smaller NLL value reflects a better fit of the given dataset, and a smaller BIC score indicates lower model complexity and better robustness. As shown in Fig. 5, the NLL value significantly decreased in the first two iterations and slightly bounced back afterward. The BIC score, however, keeps declining across all five iterations. This result indicates that the user-feedback mechanism generally improves the goodness-of-fit of the causality analysis model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.3030465, IEEE Transactions on Visualization and Computer Graphics

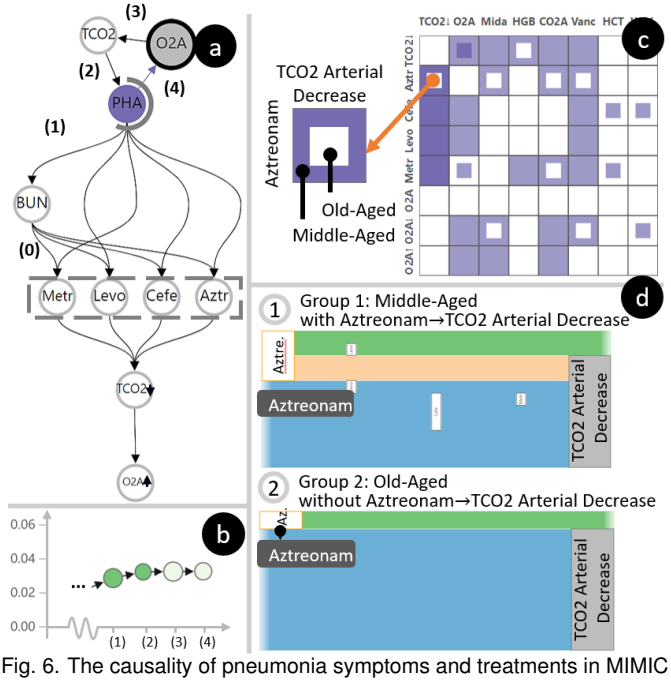


Fig. 6. The causality of pneumonia symptoms and treatments in MIMIC dataset. (a-b) Generating a causality graph for middle-aged pneumonia patients. (c-d) Comparing the causalities of middle-aged and old-aged pneumonia patients with evidence from raw data.

Accuracy. We utilize the Area Under ROC (AUROC) to evaluate the effect of the user-feedback mechanism on the improvement of accuracy. Note that the ground-truth causality provided by human modification is excluded when calculating AUROC in each iteration so that the value is only influenced by the change of causalities from the model. Higher AUROC indicates a better accuracy of the causality analysis result against the ground-truth causality. As shown in Fig. 5, the AUROC value gradually increases as the user provides valid corrections to the causality analysis result. This observation indicates that the performance of the model in terms of inferring accurate causality can be improved by the user-feedback mechanism if the feedback provided by the user is correct.

6.2 Case Studies

We demonstrate the usability of SeqCausal through two case studies in different application scenarios using electronic health records and social media interactions, respectively.

6.2.1 Causality in Electronic Health Records

This case study employs the MIMIC dataset, which contains electronic health records of over 46,000 patients with various diseases. We invited two pulmonologists (**E1, E2**) with more than 8 years of clinical experience to participate in our case study. In particular, the medical experts were also involved in determining the design requirements of SeqCausal discussed in Section 3. Prior to the study, we asked the pulmonologists to identify a list of key event types that might be causally related for analysis, which includes 120 events under the category of laboratory tests and medical treatments. Since all variables in our causality analysis model are discrete events, we preprocessed continuous laboratory tests by filtering out the normal records and discretizing the abnormal value by whether it increases or decreases compared to the previous record. Note that the increase and decrease of values only represent the occurrence of discrete events instead of directions of the causal relation. We encode three situations of the laboratory tests by varying their labels: the abnormal values with no previous record are labeled by the name of the lab test event, and the value increase and decrease are labeled with an ascending or descending arrow in the end.

The study started with a 20-minute introduction of the system and visualization design. Then, the doctors took an hour exploring our system and demonstrated their findings in a think-aloud manner. Finally, we conducted a 30-minute post-study interview collecting the doctors' subjective comments on the system's usability. In the following, we report two representative insights and discuss feedback from the experts.

Causality of pneumonia symptoms. The doctors queried a group of 127 middle-aged patients aging from 50 to 60 who were diagnosed with pneumonia. The retrieved dataset contains 93 types of events. They started by adding *O2 arterial increase* as an outcome event to explore its causes, which is an important sign of recovery for pneumonia patients. After several iterations of confirming causalities and model updates, the doctors noticed that *abnormal BUN* value was identified as the cause of taking four treatments for improving renal functionality (Fig. 6(a-0)). This is in line with their domain knowledge as the abnormality in *BUN* indicates renal damage. In addition, the system suggested that *abnormal arterial pH* was a potential cause for *BUN* anomaly (Fig. 6(a-1)). After inspecting the *causal sequence view*, where half of the patients with *abnormal arterial pH* are also abnormal in the test of *BUN*, the doctors confirmed this causal relation, and the regression likelihood of the model was improved (Fig. 6(b)). The doctors further examined the cause of *abnormal arterial pH* and found a causality circle among three laboratory indices after three iterations of update: *O2*, *TCO2*, and *pH* values in the artery. The doctors confirmed the causality circle and explained: "For patients with pneumonia, the value of oxygen, the value of carbon dioxide, and the value of *pH* in blood always affect each other. Because of this cyclical causality, the conditions of patients will keep getting worse [if not intervened]." At this point, the causal chain for the outcome event *O2 arterial increase* is complete, and the doctors saved the final causality to the *analysis history view*.

Effect of antibiotic medicines in different cohorts. After analyzing the causalities of middle-aged patients, the doctors further queried a group of old-aged patients aging from 80 to 90 for comparison, as they anticipated that the effect of antibiotic therapy might differ between age groups. The queried dataset contains 174 sequences and 79 event types. As shown in Fig. 6(c), both groups have causal relations that link toward the decrease of *arterial TCO2*, which is an important indicator for the improvement of the patient's condition. However, the doctors found that the use of *Aztreonam* was effective in the middle-aged cohort, whereas the old-aged cohort did not have such causal relation. This can also be observed from the *causal sequence view*, where *Aztreonam* seemed effective to half of the middle-aged cohort (Fig. 6(d-1)) but none of the old-aged cohort (Fig. 6(d-2)). In addition, the middle-aged group seemed to have multiple choices of antibiotic treatments. As shown in Fig. 6(d-1), a large group of patients with a decrease in *arterial TCO2* had not taken *Aztreonam*. **E2** found this reasonable, as he said: "Elderly patients are normally weak and suffer from many other complications. It needs to be particularly careful to apply antibiotics medicine to them."

Expert feedback. Both experts felt that the query view is very useful in medical scenarios for filtering a cohort with similar conditions (**R1**). They suggested that more detailed filters could be added, such as ranges of some key laboratory tests. The doctors also felt the design of the *causal graph view* and the *causal sequence view* could help them explore and verify causalities in medical events efficiently (**R2, R4**). As **E1** commented:"This system can help us discover potentially causal related medical events or spurious causal relations, and allow us to verify the relations in the original data efficiently." **E2** found that personally confirming the causal relations enhances his confidence on the causality analysis result, especially with the performance of the model displayed in the *model diagnostics view* (**R5, R6**).

The experts also commented on the visualization of our system. **E1** applauded the layout scheme of the causal graph, as he said: "The causal chains are easy to trace and the cyclical structure is properly emphasized." He also found the design of the *causal sequence view* useful for interpreting and verifying the causal relations (**R3**): "In this view, we can easily infer the causes and effects from original data. It is a good way to interpret and verify the causal relations." In terms of comparing causal relations (**R7**), both experts agreed that it is convenient to retrieve previous analysis results from the *analysis history view*. However, the experts felt that the matrix-based design is a bit overwhelming, and **E1** suggested that: "Using text descriptions to illustrate the differences may be easier to understand."

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2020.3030465, IEEE Transactions on Visualization and Computer Graphics

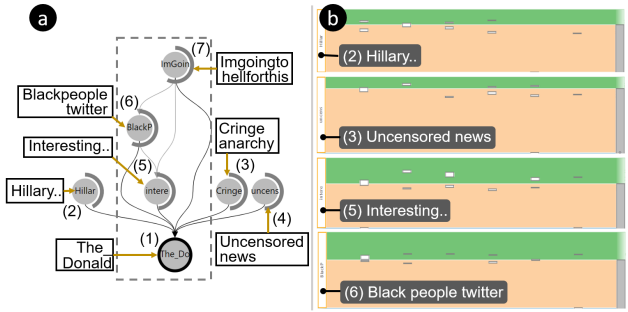


Fig. 7. The causality of subreddit interactions showing how users from other subreddits are attracted to *The_Donald*.

6.2.2 Causality of Subreddit Interactions

We also applied our system in analyzing the sequences of user comments on different subreddits from 2016 to 2017. Each subreddit represents a sub-community of the Reddit community with a particular area of interest. We extracted each user's commenting trajectory on various subreddits as an event sequence, where each event is a subreddit with a timestamp. We invited a Reddit user who is familiar with the characteristics of different sub-communities to analyze how the traffic of different subreddits is mutually affected by their causalities. The study lasted 45 minutes, including a 20-minute system introduction and a 25-minute causality analysis using our system. The participant showed his interest in analyzing how Reddit users were attracted to *The_Donald*, which was created in support of U.S. President Donald Trump and became particularly popular during the presidential campaign in 2016. Therefore, he queried Reddit users who had commented under *The_Donald* by adding it as the key event. The system retrieved 204 sequences with 165 event types for analysis.

He initiated the exploration of causalities by adding *The_Donald* as the outcome event (Fig. 7(a-1)). To eliminate noisy causal relations, he raised the threshold of event coverage to make sure that each causal relation exists in at least 30% of the queried sequences. After three iterations of the model update, new events were no longer added to the causality graph. Fig. 7 shows the final state of the causality graph. The participant found the traffic of *The_Donald* mainly came from three sources: *Hillary...* (Fig. 7(a-2)), a subreddit of negative comments from Hillary Clinton's opponents, who was also participated in the presidential election in 2016; *CringeAnarachy* and *UncensoredNews* (Fig. 7(a-(3,4)), two subreddits of mostly politically related news; and several popular subreddits of anecdotes, including *Interesting...*, *BlackPeopleTwitter*, and *ImGoingHellForThis* (Fig. 7a-(5,6,7)). He then turned to the *causal sequence view* to check the validity of causal relations. Although all subreddits in the graph can directly cause *The_Donald* according to the causality analysis result, the causal patterns in raw event sequences indicate the difference of these causal relations. In particular, the participant found that subscribers of politically related subreddits had a smaller group of sequences with A→? pattern comparing with anecdotes subscribers. Examples are illustrated in Fig. 7(b): subsequences correspond to causes *Hillary...* and *UncensoredNews* have narrower green edges compared to the causes *Interesting...* and *BlackPeopleTwitter*. This indicates that *Hillary...* and politically related news are more likely to be valid causes of the *The_Donald*'s popularity.

7 Discussion

This section includes a discussion on the generalizability of our system, the scalability of our causality analysis algorithm, the limitations of the current study, and promising future directions.

Generalizability. Although the design requirements of SeqCausal were collected from the medical domain, the causality analysis algorithm and visualization were designed for general event sequence analysis and can be easily generalized. For sequences of discrete events in continuous time (e.g., web clickstreams, social media actions, etc.) where events are not observed in fixed time lags, our causality analysis algorithm can be directly utilized, whereas for fixed time-lagged sequences (e.g., text streams, discretized time-series), the sampling function $\kappa(t)$ as mentioned in Section 4.1 needs to be replaced with Poisson sampling to better fit events in discrete time. The visualization design of our system is not tailored to a specific application domain and can be directly applied to any event sequence dataset.

# of event types (V)	# of occurrences for all events (n)		
	5334	10668	16002
31	1.82±0.02	3.75±0.02	5.54±0.02
62	1.88±0.02	3.75±0.02	5.49±0.02
93	1.83±0.02	3.7±0.03	5.56±0.03

Table 1. The running time of the causal modeling algorithm under different numbers of event types and event occurrences.

Scalability. We tested the scalability of our causal modeling algorithm with nine synthetic datasets of different numbers of event types and event occurrences. The synthetic datasets were generated by modifying a MIMIC case study dataset. We recorded the running times on a Linux server with an Intel Xeon CPU (GD6148 2.4 GHz/20-cores) and 192 GB RAM. As illustrated in Table 1, the running time increases with the number of event occurrences. However, it is independent of the number of event types, benefiting from the parallel computation of events described in Section 4.3. Although the periodic delay was not noticed as a problem by our expert users in the case study, the system may become difficult to interact in real-time as the number of event occurrences grows (i.e., the length or the number of sequences becomes larger). This requires further research for more efficient tuning of the causality analysis algorithm.

The scalability issue also exists in visualizing and exploring the causal relations (**R2**). Although we mitigate the problem by introducing the layout algorithm and user-driven causality exploration procedure described in Section 5.2, the growing number of event types can increase the complexity of the causal graph displayed in the *causal model view*, making it difficult for the user to visually explore and interact with. Our current design cannot fully support the analysis of event sequence data with very high dimensionality. A more scalable visualization and efficient interaction mechanism for high-dimensional causal graphs are required in the future research.

Limitations and future work. In addition to the scalability issue, we also recognize several other limitations of our work that point toward promising future directions. First, our system currently supports exploring causalities from effects to search for the causes. However, our medical experts suggested that it is also meaningful to investigate potential effects from causes for prognostic analysis. Allowing bi-directional exploration of the causal relations would increase the exploring space, which requires more efficient interaction methods to be studied in the future. In addition, our causality analysis algorithm is not capable of discovering combined causes. For example, in the case where two treatments together cause the cure of a symptom, our system identifies each treatment as an individual cause. Although there are some causality analysis algorithms that are capable of mining combined causes [3, 33], they mainly focus on analyzing non-temporal data and cannot be directly applied to temporal event sequences. We still need to explore more advanced causality analysis algorithms and design corresponding visualizations to support accurate and efficient analysis of combined causes for event sequence data.

8 Conclusion

In this paper, we have presented SeqCausal, an interactive visual analytics system for analyzing causalities in the event sequence dataset. The system employs a Granger causality analysis algorithm based on Hawkes processes with a user-feedback mechanism to leverage human knowledge in revising the causality analysis model. Analysts can utilize the system to discover causal relations of events, investigate complex causalities with efficiency through the causal exploration, verification, and comparison. Our quantitative study has demonstrated that the goodness-of-fit and accuracy of the model can be iteratively improved with our user-feedback mechanism. The case studies have shown the capabilities of our system in helping experts extract interesting insights into potential causal related events and discover useful causal patterns.

Acknowledgments

Nan Cao is the corresponding author. We would like to thank all the study participants, domain experts, and reviewers for their valuable comments. This work was supported in part by the Fundamental Research Funds for the Central Universities (Grant No.22120190216).

REFERENCES

[1] Subreddit interactions for 25,000 users, 2017. Retrieved from `https://www.kaggle.com/colemaclean/subreddit-interactions`.

[2] M. Achab, E. Bacry, S. Gaïffas, I. Mastromatteo, and J.-F. Muzy. Uncovering causality from multivariate hawkes integrated cumulants. *JMLR*, 18(1):6998–7025, 2017.

[3] E. Azizi, E. Airoldi, and J. Galagan. Learning modular structures from network data and node variables. In *ICML*, pp. 1440–1448, 2014.

[4] J. Bae, T. Helldin, and M. Riveiro. Understanding indirect causal relationships in node-link graphs. In *Computer Graphics Forum*, vol. 36, pp. 411–421, 2017.

[5] K. P. Burnham and D. R. Anderson. Multimodel inference: understanding aic and bic in model selection. *Sociological Methods & Research*, 33(2):261–304, 2004.

[6] N. Cartwright. What are randomised controlled trials good for? *Philosophical Studies*, 147(1):59, 2010.

[7] M. Chen, A. Trefethen, R. Banares-Alcantara, M. Jirotka, B. Coecke, T. Ertl, and A. Schmidt. From data analysis and visualization to causality discovery. *IEEE Computer*, 44(10):84–87, 2011.

[8] T. N. Dang, P. Murray, J. Aurisano, and A. G. Forbes. Reactionflow: an interactive visualization tool for causality analysis in biological pathways. In *BMC Proceedings*, vol. 9, p. S6. Springer, 2015.

[9] T. Dwyer, K. Marriott, and P. J. Stuckey. Fast node overlap removal. In *International Symposium on Graph Drawing*, pp. 153–164. Springer, 2005.

[10] M. Eichler. Causal inference with multiple time series: principles and problems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1997):20110613, 2013.

[11] M. Eichler, R. Dahlhaus, and J. Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 38(2):225–242, 2017.

[12] M. Eichler, R. Dahlhaus, and J. Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. vol. 38, pp. 225–242. Wiley Online Library, 2017.

[13] N. Elmqvist and P. Tsigas. Causality visualization using animated growing polygons. In *IEEE InfoVis*, pp. 189–196, 2003.

[14] N. Elmqvist and P. Tsigas. Growing squares: Animated visualization of causal relations. In *Proceedings of ACM Symposium on Software Visualization*, p. 17, 2003.

[15] D. Entner and P. O. Hoyer. On causal discovery from time series data using fci. *Probabilistic Graphical Models*, pp. 121–128, 2010.

[16] E. R. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In *International Symposium on Graph Drawing*, pp. 239–250. Springer, 2004.

[17] X. Geng, Z. Chen, W. Yang, D. Shi, and K. Zhao. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing*, 11(4):3680–3689, 2011.

[18] D. Gotz and H. Stavropoulos. Decisionflow: Visual analytics for high-dimensional temporal event sequence data. *IEEE TVCG*, 20(12):1783–1792, 2014.

[19] C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

[20] R. Guo, L. Cheng, J. Li, P. R. Hahn, and H. Liu. A survey of learning causality with data: Problems and methods. *arXiv preprint*, 2018.

[21] S. Guo, Z. Jin, D. Gotz, F. Du, H. Zha, and N. Cao. Visual progression analysis of event sequence data. *IEEE TVCG*, 25(1):417–426, 2018.

[22] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao. Eventthread: Visual summarization and stage analysis of event sequence data. *IEEE TVCG*, 24(1):56–65, 2017.

[23] A. G. Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443, 1971.

[24] P. W. Holland. Statistics and causal inference. *JASA*, 81(396):945–960, 1986.

[25] A. Hyvärinen, S. Shimizu, and P. O. Hoyer. Causal modelling combining instantaneous and lagged effects: an identifiable model based on non-Gaussianity. In *ICML*, pp. 424–431, 2008.

[26] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.

[27] N. R. Kadaba, P. P. Irani, and J. Leboe. Visualizing causal semantics using animations. *IEEE TVCG*, 13(6):1254–1261, 2007.

[28] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *ACM SIGKDD*, p. 497–506, 2009.

[29] E. Lewis and G. O. Mohler. Research article a nonparametric em algorithm for multiscale hawkes processes. vol. 1, pp. 1–20, 2011.

[30] S. Liu, G. Andrienko, Y. Wu, N. Cao, L. Jiang, C. Shi, Y.-S. Wang, and S. Hong. Steering data quality with visual analytics: The complexity challenge. *Visual Informatics*, 2(4):191–197, 2018.

[31] A. C. Lozano, N. Abe, Y. Liu, and S. Rosset. Grouped graphical granger modeling methods for temporal causal modeling. In *ACM SIGKDD*, pp. 577–586, 2009.

[32] Y. Lu, H. Wang, S. Landis, and R. Maciejewski. A visual analytics framework for identifying topic drivers in media events. *IEEE TVCG*, 24(9):2501–2515, 2017.

[33] S. Ma, J. Li, L. Liu, and T. D. Le. Mining combined causes in large data sets. *Knowledge-Based Systems*, 92:104 – 111, 2016.

[34] D. Malinsky and D. Danks. Causal discovery algorithms: A practical guide. *Philosophy Compass*, 13(1):e12470, 2018.

[35] H. Mei and J. M. Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *NeurIPS*, pp. 6754–6764, 2017.

[36] A. Michotte, G. Thines, A. Costall, and G. Butterworth. La causalité perceptive. *Journal de psychologie normale et pathologique*, 60:9–36, 1963.

[37] M. Nauta, D. Bucur, and C. Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340, 2019.

[38] J. Pearl et al. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 2009.

[39] J. Peters, D. Janzing, and B. Schölkopf. Causal inference on time series using restricted structural equation models. In *NeurIPS*, pp. 154–162, 2013.

[40] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic. Detecting causal associations in large nonlinear time series datasets. *arXiv preprint*, 2017.

[41] S. Shimizu. LiNGAM: Non-Gaussian methods for estimating causal structures. *Behaviormetrika*, 41(1):65–98, 2014.

[42] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.

[43] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman. *Causation, Prediction, and Search*. MIT Press, 2000.

[44] P. A. Stokes and P. L. Purdon. A study of problems encountered in granger causality analysis from a neuroscience perspective. *National Academy of Sciences*, 114(34):E7063–E7072, 2017.

[45] E. Wall, L. M. Blaha, C. L. Paul, K. Cook, and A. Endert. Four perspectives on human bias in visual analytics. In *Cognitive Biases in Visualizations*, pp. 29–42. Springer, 2018.

[46] J. Wang and K. Mueller. The visual causality analyst: An interactive interface for causal reasoning. *IEEE TVCG*, 22(1):230–239, 2015.

[47] J. Wang and K. Mueller. Visual causality analysis made practical. In *IEEE VAST*, pp. 151–161, 2017.

[48] Y. Wang, Y. Wang, Y. Sun, L. Zhu, K. Lu, C.-W. Fu, M. Sedlmair, O. Deussen, and B. Chen. Revisiting stress majorization as a unified framework for interactive constrained graph visualization. *IEEE TVCG*, 24(1):489–499, 2017.

[49] K. Wongsuphasawat, C. Plaisant, M. Taieb-Maimon, and B. Shneiderman. Querying event sequences by exact match or similarity search: Design and empirical evaluation. *Interacting with Computers*, 24(2):55–68, 2012.

[50] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha. Learning time series associated event sequences with recurrent point process networks. *IEEE TNNLS*, 30(10):3124–3136, 2019.

[51] H. Xu, M. Farajtabar, and H. Zha. Learning granger causality for hawkes processes. In *ICML*, pp. 1717–1726, 2016.

[52] W. Zhang, T. K. Panum, S. Jha, P. Chalasani, and D. Page. Cause: Learning granger causality from event sequences using attribution methods. *arXiv preprint*, 2020.

[53] Z. Zhang, K. T. McDonnell, E. Zadok, and K. Mueller. Visual correlation analysis of numerical and categorical data on the correlation map. *IEEE TVCG*, 21(2):289–303, 2014.