

Supporting Guided Exploratory Visual Analysis on Time Series Data with Reinforcement Learning

Yang Shi , Bingchang Chen, Ying Chen, Zhuochen Jin, Ke Xu, Xiaohan Jiao, Tian Gao, and Nan Cao 

Abstract— The exploratory visual analysis (EVA) of time series data uses visualization as the main output medium and input interface for exploring new data. However, for users who lack visual analysis expertise, interpreting and manipulating EVA can be challenging. Thus, providing guidance on EVA is necessary and two relevant questions need to be answered. First, how to recommend interesting insights to provide a first glance at data and help develop an exploration goal. Second, how to provide step-by-step EVA suggestions to help identify which parts of the data to explore. In this work, we present a reinforcement learning (RL)-based system, Visail, which generates EVA sequences to guide the exploration of time series data. As a user uploads a time series dataset, Visail can generate step-by-step EVA suggestions, while each step is visualized as an annotated chart combined with textual descriptions. The RL-based algorithm uses exploratory data analysis knowledge to construct the state and action spaces for the agent to imitate human analysis behaviors in data exploration tasks. In this way, the agent learns the strategy of generating coherent EVA sequences through a well-designed network. To evaluate the effectiveness of our system, we conducted an ablation study, a user study, and two case studies. The results of our evaluation suggested that Visail can provide effective guidance on supporting EVA on time series data.

Index Terms—Time Series Data, Exploratory Visual Analysis, Reinforcement Learning

1 INTRODUCTION

A time series is a consecutive sequence of data points that are measured over time and is prevalent in various application domains such as health and finance. To understand new time series data, exploratory visual analysis (EVA) stands out as a preferred method that involves identifying questions of interest, investigating visualized data, and formulating hypotheses [6]. However, conducting EVA on time series data is challenging for its target users, who are typically experts in their domain but novices with regard to visual analytics. EVA primarily relies on visualization as the mode of output and input interface for exploration, which can be demanding or even overwhelming for users to interpret and manipulate [10]. “Which variable to look at?” or “How to set visualization configurations and parameters?” Such questions are not easily answered and thus appropriate guidance is needed to narrow the gap that hinders effective continuation of EVA on time series data [10].

Prior research in EVA for time series data has focused on computational analysis [58, 74] or visual representation [9, 40, 73]. However, few attempts have been made to provide guidance on walking users through the process of EVA. Such guided sequencing of actions and discourse with data can facilitate users’ analytical reasoning tasks [51]. To automate guided EVA for time series data, two challenges need to be addressed: (i) *goal unclear*. Typically, an analyst starts EVA without a clear exploration goal until he or she learns more about the data. Thus, interesting data insights should be automatically extracted and recommended to users, providing a first glance at data and helping develop a clear goal; and (ii) *path unknown*. The path leading to the desired goal may also be unknown to the analyst, especially when dealing with time series data that is usually characterized by multiple variables or long sequences. Thus, it is important to provide the analyst with step-by-step EVA suggestions that help determine which parts of the data to explore.

To address the above challenges, we developed Visail, a reinforcement learning (RL)-based system to support guided EVA on time series data. Specifically, given a time series dataset uploaded by a user, Visail can generate an EVA sequence consisting of multiple steps, with each step illustrating a time subsequence of interest and its corresponding data insight. Here, insight refers to a unified formulation of interesting data patterns [19]. To do this, we first collected a set of exploratory data analysis (EDA) notebooks to understand common insights regarding time series data and their co-occurrence relationships. Then, we designed an RL-based algorithm to learn how to explore insights and form an EVA sequence, steering a coherent path to analyze data, and helping develop an exploration goal. Next, Visail presents the EVA sequence through visualization charts combined with textual descriptions. Visail also integrates an online feedback mechanism, empowering users to modify the sequence based on their preferences and then update the generation accordingly. We evaluated Visail on its ability to support EVA via an ablation study, a user study, and two case studies. The results of our evaluation suggested that Visail can effectively and efficiently support guided EVA on time series data.

In summary, the main contributions of this work are as follows:

- **System.** We developed Visail¹ to support users in conducting EVA on time series data by generating step-by-step suggestions. Visail shows the suggestions in the form of captioned charts to facilitate communication and provides various interaction techniques for authoring.
- **Algorithm.** We proposed an RL-based algorithm to generate EVA sequences. Through the state space, action space, and reward function specifically designed for time series data, the algorithm learns from the environment to perform data exploration tasks.

2 RELATED WORK

Our work builds on prior research on guided data exploration, visual analysis of time series data, and automated data insights.

2.1 Guided Data Exploration

Exploratory Data Analysis (EDA) was first raised by John Tukey [20], which focuses on exploring and understanding a new dataset. Since then, researchers have explored how to guide EDA where the goal is to discover insights and present them as a sequence of interesting data “views” [22]. For example, ATENA [5] and Dora The Explorer [44] both formulate EDA as a control problem and employ reinforcement learning (RL) to generate a sequence of operations (e.g., filter, group-by) and data queries that optimize the exploration process.

• Yang Shi, Bingchang Chen, Ying Chen, Tian Gao, Xiaohan Jiao, and Nan Cao are with Intelligent Big Data Visualization Lab at Tongji University while Zhuochen Jin and Ke Xu are with Huawei Cloud Computing Technologies Co., Ltd. Nan Cao is the corresponding author. E-mail: {yangshi.idvx, 2131933, 2131926, gaotian, xh_xiaohan, nan.cao}@tongji.edu.cn. {chjzcjames, lukexuke}@gmail.com.

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.xxx/TVCG.201x.xxxxxxx

¹<https://idvxlab.com/visail/>

As a subset of EDA, Exploratory Visual Analysis (EVA) relies on visualization as the primary output to derive insights and channels of interaction [6]. To facilitate guided EVA, a suite of tools has been developed [17, 52, 53, 67, 69, 76]. For example, ChartSeer [76] recommends the next exploration step to users by calculating the similarity between charts based on the Vega-Lite representation of charts [49] while other work recommends next interesting visualizations to users by embodying analytical actions that correspond to different operations on a dataset [11, 38]. To show the results of EVA, different forms of representation were used such as dashboards [17, 30, 69] and fact sheets [53, 65]. For example, Dashbot [17] employs RL to explore the large space of chart combinations by taking recommended actions (e.g., add, change) and generating dashboards while Calliope [53] explores data using a Monte Carlo tree search and presents the top-rated path as a data story consisting of multiple data facts, while each data fact is visualized as a captioned chart.

The aforementioned work was geared toward multi-dimensional tabular data. Our work focuses specifically on time series data, which raises new and unique challenges [57] and proposed to leverage RL to provide EVA guidance on time series data for the first time. Also, our work used a mixed-initiative approach [28] that not only guides the user exploring data but also the user can guide Visail providing feedback on the insights generated.

2.2 Visual Analysis of Time Series Data

Visual analysis of time series data can be placed along a spectrum of goal specifications, from targeted question answering to open-ended exploration [6]. In terms of targeted question answering, existing studies focused on developing visual analysis tools for various questions, which are mainly categorized into five visual analysis tasks, including summarization [23, 45], anomaly detection [41, 60], comparison [24, 68], prediction [71], and causality analysis [18]. For example, MulTiDR [23] uses dimensionality reduction to obtain a visual summary of extensive, high-dimensional time series data while TreeRoses [60] overlays the visualization of periodic multivariate time series data for a more straightforward interpretation of patterns and outliers.

On the other end of the spectrum, open-ended exploration describes a common scenario where an analyst receives an unfamiliar dataset and lacks a clear goal to start the analysis process. Several research efforts have been devoted to open-ended exploration through interactive browsing and querying [58, 74, 75, 77]. For example, KronoMiner [74] visualizes time series data based on a radial display, providing users with rich interaction to explore different patterns or drill into details. Sips et al. [58] identifies potential interesting patterns from environmental time series data using a matrix visualization and offers interactions for users to explore the detected patterns.

The above visual analysis tools were primarily designed for those who are equipped with visual analytics skills and focus on single-step guidance. In contrast, our work considers users who lack visual analytics expertise and falls under the open-ended exploration dichotomy of the spectrum, helping users gain a brief understanding of time series data by steering the process of EVA.

2.3 Automated Data Insights

Data insights describe a set of interesting data patterns that “*represent the nuanced understanding of the data shaped by a user’s mental model as well as the social and organizational context where findings are discovered*” [37]. Data insights can be characterized by their properties of being actionable, collaboratively refined, unexpected, confirmatory, spontaneous, trustworthy, and interconnecting [37]. Given their importance, data insights have received increasing interest in the visualization community and more tools have been developed for their automatic generation. Common types of automated data insights include outliers, value/derived value, correlation, difference, trend, distribution, extreme, visual motifs, cluster, metadata, rank, and compound fact [14, 36].

In terms of tasks supported by automated data insights, many research efforts have centered on exploratory analysis, communication, and focused analysis [36]. Tools focused on the task of exploratory analysis aim to provide a first glance at data before modeling and

analysis is performed [7, 15, 16, 19, 62]. For example, SeeDB [62] explores the space of visualizations and recommends visualizations for trends to facilitate visual analytics. To support the task of communication, tools have been developed to generate charts combined with textual descriptions [9, 53, 59, 65]. For example, Dashot [65] generates infographic-style fact sheets while Calliope [53] generates a sequence of data facts connected by coherent relations to support data-driven storytelling. Also, directed analysis helps answer high-level questions by recommending possible data facts [34, 35]. Inspired by the aforementioned work, Visail also applies the auto-insights techniques to extract data insights but takes a step further by generating step-by-step EVA suggestions.

3 PRELIMINARY STUDY

To motivate the design of Visail, we first conducted a series of semi-structured interviews with data analysts to understand the design requirements of Visail. Based on their feedback, we then collected a set of EDA notebooks to identify those insight types that are commonly uncovered in the analysis of time series data.

3.1 Formative Interviews with Data Analysts

To understand how data analysts conduct open-ended EVA of time series data in practice, we collaborated with a database team of five members from an information and communications technology company. The members of this team, though well-versed in database systems, were not visualization experts and had limited experience in using complex visual analytic tools. During the ten-month collaboration, we held regular meetings with these experts to discuss the design and development of Visail. Example interview questions include “*what will interest you when given a new dataset to analyze*” and “*what is important to you in tools for analyzing time series data?*”. Later, three iterations of Visail were developed and demonstrated to the five members for the purpose of collecting feedback. By analyzing the findings and challenges derived from the interview, we identified the following design requirements.

DR1 Extracting data insights. Visail should automatically extract data insights from a given time series dataset. These insights will serve as building blocks of EVA and provide a basic understanding of the dataset.

DR2 Generating EVA sequences. Visail should generate EVA sequences composed of the extracted insights to provide step-by-step suggestions. A sequence should form a coherent EVA path and covers important observations of the dataset, directing users to the data of interest.

DR3 Supporting visual representation and user interaction. Visail should present the EVA sequences using a combination of visualization charts and textual descriptions. Also, Visail should support user interaction: users can explore time series data, edit the extracted insights, and modify the EVA sequences.

DR4 Enabling online recommendation based on user modification. Visail should support users in modifying specific steps of the EVA sequences and automatically update the subsequent steps based on their adjustment. Facilitating online recommendations can involve users’ analysis preferences in the generation.

3.2 Data Collection and Analysis

To understand insight types that frequently emerge from an exploratory analysis of time series data (DR1), we sought to collect relevant examples or cases. However, existing studies primarily focus on high-level analysis tasks such as lookup and comparison [1] or more general data types [3]. As an alternative, we turned our attention to exploratory data analysis (EDA) notebooks (e.g., Jupyter Notebook), which are the single most popular and effective means for data scientists to perform EDA [48]. We collected EDA notebooks from Kaggle, which is one of the largest data competition platforms with a collection of high-quality EDA notebooks. First, we filtered competitions tagged with ‘time series analysis’. Then, we searched for top-voted notebooks (with at least 10 votes) tagged with ‘exploratory data analysis’. The query returned competitions, including a total of eight multivariate time series, with

an average length of 220,536 and an average number of variables of 31,707. Ten EDA notebooks were collected from each of the eight competitions, resulting in 80 EDA notebooks in total.

An EDA notebook typically consists of two parts: the first part is open-ended data exploration where analysts aim to get a preliminary understanding of the dataset while the second part is devoted to more focused analysis tasks such as prediction. Our analysis of the 80 notebooks focused on their first parts and specifically the concluding sections called “observation” or “insights”, where analysts describe their process of EDA and list insights observed from the data. To analyze the EDA notebooks, we first coded the descriptions in these sections based on the common types of data insights proposed by Law et al. [36]. We also found that these insight types cannot cover all the insight types introduced in these notebooks and thus added three new insight types specifically related to time series data, including *autocorrelation*, *seasonality*, and *multivariate outlier*. As a result, our analysis arrived at eight insight types, which are described as follows.

Distribution (358, 28.8%) describes the overall pattern and deviations (e.g., mean, variance) of a time series.

Extreme (196, 15.7%) is the time point with the leading value of a time series.

Trend (182, 14.6%) shows the increasing or decreasing movement of a time series over a period of time.

Correlation (176, 14.1%) measures the relationship between two or more time series.

Similarity (125, 10.0%) measures the synchrony of time series data by comparing multiple subsequences of one or multiple time series.

Outlier (94, 7.6%) refers to the time point or subsequence that significantly deviates from the patterns of other values. Outliers can be detected in both univariate and multivariate time series data.

Seasonality (92, 7.3%) describes the presence of variations that appear at specific regular intervals in a time series.

Autocorrelation (22, 1.7%) measures the correlation of a subsequence with a delayed copy of itself. The correlation can be either positive or negative.

After identifying the eight insight types from the EDA notebooks, we attempted to analyze the co-occurrence relationships between them. For example, after identifying an ‘outlier’, analysts might investigate for ‘distribution’ to determine if such an outlier is random or systematic. To do this, we leveraged the N-gram language modeling [26, 31], which is a commonly used Natural Language Processing technique. The N-gram refers to a series of consecutive items with length n obtained by splitting from the text. Inspired by the N-gram, we first coded each EDA notebook as a sequence of insights and considered the sequence as a ‘sentence’ (denoted as $sentence_i$), where each insight is regarded as a ‘word’ (denoted as $word_i$) in the $sentence_i$. Then, we used a sliding window with $n = 2$ as its size to scan each $sentence_i$ with a length of m , resulting in 1,165 $word_i$ pairs. Fig. 1 shows the co-occurrence probability of $word_{i+1}$ (column) given $word_i$ (row) by analyzing the 1,165 $word_i$ pairs. The higher the probability $P(word_{i+1}|word_i)$, the more likely it is that the two corresponding insight types will co-exist in a sequence.

4 PROBLEM FORMULATION

In this section, we introduce how we shaped the generation of EVA sequences (DR2) into a reinforcement learning (RL) problem. First,

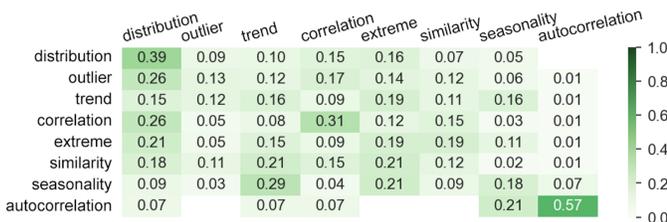


Fig. 1: The probability of co-occurrence relationships between the eight insight types.

we described the Markov Decision Process (MDP) model, which forms the foundation of RL. Next, we defined the state space, action space, and reward function, through which an RL-based algorithm learns how to generate EVA sequences without labeled training data.

4.1 MDP Model for EVA

The MDP model [8] describes a stochastic control process. It consists of a set of possible states and a set of possible actions, which can be denoted as follows,

$$MDP = \{S, A, P, R, \gamma\}, \quad (1)$$

where S is the state space, A is the action space, P is probabilities of transition between two states, R is the reward function, and γ is the discount factor. An action $a_n \in A$ is chosen based on the current state $s_n \in S$ and taken to transfer from s_n to s_{n+1} with probability $P(s_n, s_{n+1})$. Then, such a pair of the state and action (s_n, a_n) can get an immediate reward r_n defined by the reward function R . Given its characteristics, the MDP model is often used to model data exploration tasks [5, 17], as analysts usually base their subsequent analysis on existing findings.

An EVA sequence refers to a sequence of small, incremental steps [25, 66] while at each step, analysts take a data-driven action, i.e., deriving an insight from data and presenting it as a visualization chart [2]. Here, the action taken at a certain step is based on all the insights explored in the previous steps. Using the MDP model, an EVA sequence can be modeled as a sequence of state-action pairs $\{(s_1, a_1), (s_2, a_2), \dots, (s_n, a_n)\}$ while $P(s_n, s_{n+1})$ always equals to 1 as it is a deterministic process. We employed RL to learn an agent that maximizes the cumulative reward $\sum_{i=1}^n \gamma^{i-1} r_i$ when taking a series of actions $\{a_1, a_2, \dots, a_n\}$. Specifically, RL was chosen when considering: (i) the process of analysts trying different exploration methods to arrive at exploration goals is naturally similar to RL exploring from the environment and learning the optimal strategy to select actions; (ii) the state space is exponentially large. Thus, it is difficult to use rule-based algorithms to find optimal solutions in an acceptable time; and (iii) a lack of EVA examples might pose a challenge for supervised learning. RL is a self-learning algorithm and is bound to learn from its experience without labeled training data. The details of the state space, action space, and reward function in Equation 1 will be introduced in the following subsections.

4.2 State Space

A state $s_n \in S$ describes all the insights $\{i_1, i_2, \dots, i_n\}$ the agent explored from the first step to the n -th step, which constitutes an EVA sequence of length n . In another word, s_n captures the current insight i_n as well as the previous insights i_1, i_2, \dots, i_{n-1} , providing useful information for the agent when choosing an action a_n to transfer from s_n to s_{n+1} . An insight space I includes all the insights from a given time series dataset while an EVA sequence can be derived from a specific EVA process that explores the insight space I . A state space S enumerates all the possible permutations of the insights from the insight space I .

$$S = \{\{i_1, i_2, \dots, i_j, \dots, i_n\}, i_j \in I \mid n \in [1, N]\} \quad (2)$$

where i_j denotes the j -th insight of an EVA sequence while N is the maximum number of steps defined by a user.

To further formulate the insights included in the state space S , we borrowed the definition of insight regarding tabular data [53, 65] as *metadata* to describe the data patterns characterized by an insight. In addition, we added *value* to the definition to capture the temporal relationship of time series data covered by the insight. The reason is that in time series data, data points are collected sequentially through time [12] while in tabular data, each data point is relatively independent of the other. Thus, our definition of an insight i from time series data is as follows,

$$\text{insight} = \underbrace{\{\text{type, subspace, breakdown, measure}\}}_{\text{metadata}}, \underbrace{\{\text{raw value, derived value}\}}_{\text{value}} \quad (3)$$

where type denotes insight types, including distribution, correlation, trend, extreme, seasonality, outlier, similarity, and autocorrelation, as

identified in our preliminary study. Subspace indicates a specific time range to be observed. Breakdown refers to time granularity by which the data points in the subspace are aggregated. Measure specifies the variable(s) covered by the insight. Raw value refers to the time series data covered by the insight, which contains all the data points within a time subsequence defined by specific subspace, breakdown, and measure. Derived value denotes statistic and inferential values calculated based on the raw value of the insight. It can be used to describe the characteristics of a specific insight type such as the positive or negative slope of a trend. Using the definition, the example, “daily changes in the number of new vaccination cases from April. 30, 2021 to June. 30, 2021”, can be modeled as {“trend”, {[2021-04-30, 2021-06-30]}, {day}, {vaccination cases}, {[2275501, 2959008, ...]}, {“extent”: “+0.7”}}.

4.3 Action Space

From the state s_n , the agent selects an action a_n to discover a new insight i_{n+1} , which constitutes the latest insight of the next state s_{n+1} . The action space A lists all the possible actions to explore new insights from time series data. To formulate A , we referred to the *metadata* of insight (Equation 3) and ignored its *value*, as the *value* is determined by the data itself and cannot be altered by analysts through manipulation. As *metadata* can be represented using a four-tuple, it is natural to consider four ways to extend to new insights based on a specific insight. The new insights differ from the current insight by one element of the *metadata*, with each corresponding to a specific analysis behavior. The benefits of defining such ‘one element at a time’ actions are two-fold: (i) they serve as composable building blocks that enable full exploration of all the possible insights included in the insight space I ; and (ii) they facilitate the decomposition of complex exploration operations and then present them in a step-by-step manner. Accordingly, we identified four extending types of actions, including EXT(type), EXT(subspace), EXT(breakdown), and EXT(measure), on one insight to constitute homogeneous insights. Note that when performing EXT(measure), EXT(type) is also allowed. The reason is that when multiple variables are reduced to one variable, the original multivariate insight such as ‘correlation’ can no longer be applicable, and vice versa. Thus, EXT(measure) and EXT(type) should be executed simultaneously.

After identifying the four extending types, we attempted to find specific actions corresponding to each type. First, we analyzed all the insight *word_i* in the 80 EDA notebooks and represented each of them as a four-tuple. Then, we compared the difference between the four-tuples of every two consecutive *word_i* in a sequence *sentence_i*. Next, we mapped such a difference into one or a series of actions, corresponding to a change in one or multiple extending types. Table 1 shows the action space A . For example, “CA_2 shows a decrease in 2015 may have resulted in CA_4 experiencing its highest sales of up to 3,000 in the same year” involves two actions, CHANGE_MEASURE and CHANGE_TYPE; it first changes measure from ‘CA_2’ to ‘CA_4’ and then changes type from ‘trend’ to ‘extreme’.

4.4 Reward Function

After executing the action a_n to transfer to a new state and obtain a new EVA sequence, the environment returns a reward calculated by the reward function to inform the agent how well the EVA sequence is generated through a_n . Rewards for data exploration tasks can be designed by balancing *familiarity* and *curiosity*, where familiarity emphasizes the association between two consecutive states while curiosity encourages the discovery of unknown states [44]. Similarly, in the context of EVA, a high-quality EVA sequence can be evaluated if insights from $\{i_1, i_2, \dots, i_n\}$ are interrelated with each other (familiarity) while each one also exhibits uniqueness (curiosity).

Familiarity. Familiarity rewards the agent when the interrelationship between the insights in $\{i_1, i_2, \dots, i_n\}$ at s_n is well-maintained. Inspired by the approach to constructing homogeneous data patterns in EDA [42], we calculated familiarity RF by iterating through the *metadata* of insights $\{i_1, i_2, \dots, i_{n-1}\}$, comparing each of them with that of i_n , and counting the number of elements with the same value.

Type	Action	Description
EXT(type)	CHANGE_TYPE	Change the type of an insight.
	SHIFT_FORWARD	Move the subspace of an insight forward by a range equal to the current subspace.
	SHIFT_BACKWARD	Move the subspace of an insight backward by a range equal to the current subspace.
EXT(subspace)	SHIFT_PERIODICAL	Move the subspace of an insight to the same time period in the previous year/month/day/hour.
	AGGREGATE	Perform aggregation on the breakdown of an insight.
EXT(breakdown)	REMOVE_AGGREGATE	Remove aggregation on the breakdown of an insight.
	CHANGE_MEASURE	Change the measure of an insight to another variable.
EXT(measure)	INCREASE_MEASURE	Increase the measure of an insight from one variable to multiple variables.
	DECREASE_MEASURE	Decrease the measure of an insight from multiple variables to one variable.

Table 1: The action space in our formulation.

$$RF = \frac{1}{4 * (n-1)} \sum_{k=1}^{n-1} |m_k \cap m_n| \quad (4)$$

where \cap computes the subset of identical elements between the *metadata* of the insight i_k (m_k) and that of the insight i_n (m_n).

Curiosity. Curiosity motivates the agent to explore new states and go further in its data exploration. It includes two terms relevant to the raw value and derived value of insight, respectively. First, the raw value term RC_r evaluates the difference between time subsequences of corresponding insights, encouraging discovering new subsets of data [5,44]. To do this, we employed the CasualCNN model [21], which is capable of representing time subsequences of varying lengths as embeddings of equal length. Then, the difference between two time subsequences was computed based on the distance of their embeddings.

$$RC_r = \frac{1}{n-1} \sum_{k=1}^{n-1} \{1 - \text{cosine}(e_n, e_k)\} \quad (5)$$

where $\text{cosine}(\cdot)$ calculates the cosine similarity between the two embeddings e_k and e_n .

Second, the derived value term RC_d rewards the agent for finding a unique insight. Such an insight tends to be more meaningful and interesting [53]. To compare the uniqueness of insight, we computed the statistical significance (e.g., p-value) based on the derived value of each insight [19,65]. The details of our methods can be found in the *Supplementary Material*. The uniqueness of insight is used as its derived value term RC_d ,

$$RC_d = \text{score}(i_k) \quad (6)$$

where $\text{score}(\cdot)$ evaluates the statistical significance of the insight i_k .

Finally, we combined both familiarity (RF) and curiosity rewards (RC_r and RC_d) by weighted sum as the final reward r_n .

$$r_n = w_1 RF + w_2 RC_r + w_3 RC_d \quad (7)$$

where the weights were set as follows, $w_1 = 0.4$, $w_2 = 0.4$, $w_3 = 0.2$.

5 DESIGN OF VISAIL

Based on the problem formulation, we designed Visail that supports guided EVA on time series data. In the following, we first describe the overview of the system pipeline of Visail, followed by the details of each component.

5.1 System Pipeline

Based on the design requirements (DR1-DR4), the system pipeline was designed to include three main modules: Segmentation, Generation, and Representation.

P1 Segmentation. The segmentation module first partitions a given time series into multiple time subsequences based on its characteristics. Such time subsequences serve as the basic units for extracting insights (DR1). The module then selects a specific time subsequence and extracts its corresponding insight to form an initial EVA sequence as the starting point for the generation.

P2 Generation. The generation module generates an EVA sequence based on the RL-based algorithm (DR2). It starts with the initial EVA sequence. Then, at each following step, the module discovers a meaningful insight through the action predicted by the RL agent (DR1) and appends it to the sequence to form a coherent EVA path. Such an insight-by-insight generation method also facilitates online feedback responding to user interaction by suggesting an update on the EVA sequence (DR4).

P3 Representation. The representation module displays the generated EVA sequences through annotated visualizations along with textual descriptions specifically designed for effective communication (DR3). Also, this module provides various interaction techniques to aid in exploring time series data as well as adjusting the generated results (DR3).

5.2 Segmentation

The segmentation module partitions a given time series by referring to the approach of causal analysis regarding time series data [18]. If the time series is periodic, then segmentation is conducted according to its period; otherwise, segmentation is based on its peak. If the time series is non-seasonal and lacks peaks, it is considered as a whole as a segment (i.e., a time subsequence). After segmentation, the process of EVA starts with selecting a specific time subsequence with a meaningful insight as a starting point, which acts as the initial EVA sequence. First, two methods were used for subsequence selection: one is through user specification, and the other one is through time-bounded search [19]. Regarding time-bounded search, in a given time budget, the segmentation module samples a set of subsequences from the time series and extracts all their insights. Then, the insight with the highest statistical significance (as described in Sec. 4.4) is selected to constitute the initial EVA sequence.

5.3 Generation

The generation module leverages RL to generate EVA sequences in an insight-by-insight manner, where the coherence between consecutive insights should be maintained. In this subsection, we first describe the workflow of the RL-based algorithm, followed by the details of its key components, including feature engineering and the RL agent. Then, we demonstrate how the algorithm can be used for online recommendation.

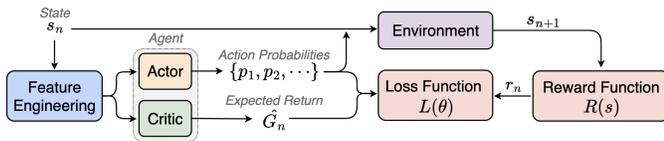


Fig. 2: Overview of the RL-based algorithm in the generation module.

5.3.1 RL-based Algorithm

The RL-based algorithm is based on Proximal Policy Optimization (PPO) [50]. We used PPO as it can strike a balance between performance and ease of tuning hyperparameters, which is suitable for data exploration tasks such as EVA. Fig. 2 shows the overview of the algorithm. First, it represents a state s_n as a vector through feature engineering and then feeds the vector into the RL agent. Next, the agent selects an action a_n from the action space A that is expected to maximize the reward. Specifically, the agent consists of two components:

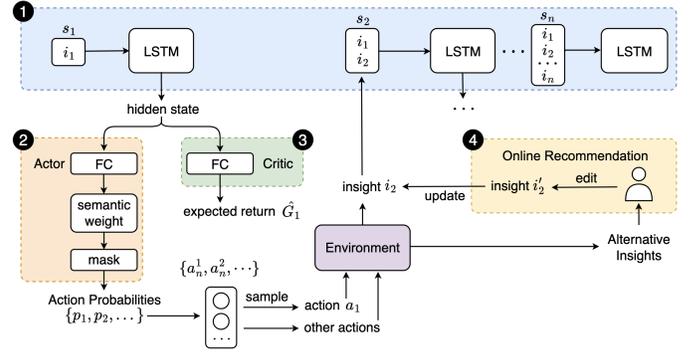


Fig. 3: The network architecture of the agent, which consists of (1) the LSTM-based encoder, (2) the actor, and (3) the critic. (4) illustrates the mechanism of online recommendation based on the network.

the actor and critic. The core concept of the actor-critic mechanism is to utilize the critic to evaluate the actor's prediction and train the actor so that actions yielding greater rewards are more likely to be selected by the actor. Then, the environment conducts the action a_n with the maximum probability on the state s_n to transfer to the next state s_{n+1} . Finally, the generation module evaluates s_{n+1} through the reward function $R(\cdot)$ to obtain the reward r_n towards a_n . The reward r_n , along with action probabilities from the actor and expected return from the critic, serve as the input for the loss function $L(\theta)$. The algorithm then moves on to the next iteration, repeating the process until the generated EVA sequence reaches the maximum length N .

5.3.2 Feature Engineering

To feed a state s_n into the agent, we first represented it as a feature vector through feature engineering. Considering s_n includes insights $\{i_1, i_2, \dots, i_n\}$, each of them can be represented as an insight vector. Regarding its *metadata*, type, subspace, breakdown, and measure are represented as vectors. To represent raw value in its *value*, we adopted CasualCNN [21] to represent raw value using fixed-length embeddings which capture the overall features of time series data as well as the time-dependent relationship between data points. We ignored derived value as it is calculated based on raw value and its information can also be inferred through the vector of raw value. As a result, a state s_n can be represented as a matrix that packs n insight vectors. Assuming that each insight vector has length l , then the feature vector v_n of the state s_n can be represented as a matrix of $n \times l$.

5.3.3 RL Agent

Given the feature vector v_n of the state s_n , the objective of the RL agent is to select the most appropriate action to discover a new insight for the subsequent state. Also, the new insight should be consistent with the insights identified in previous steps. The agent consists of two components (Fig. 3): (1) the actor that predicts the probability of performing each action $a \in A$ at the state s_n , and (2) the critic that predicts the maximum expected return which is the sum of expected rewards when continuing to explore from s_n to the final state s_N .

First, the feature vector of state s_n is fed into a shared LSTM-based encoder, as shown in Fig. 3 (1). The LSTM-based encoder can model the insights and their relationships in the state s_n , through which the agent gains a comprehensive awareness and thus selects an action that maintains such relationship of all insights in the next state s_{n+1} . Specifically, we input n insight vectors of $\{i_1, i_2, \dots, i_n\}$ at the state s_n into the encoder in order. The hidden state of the n -th insight is used as the unified representation of s_n . Such a hidden state includes contextual information by preserving the features of all the insights at s_n and capturing the sequential information between the insights based on the input order.

Then, the hidden state is fed into the actor and critic respectively, as shown in Fig. 3 (2) and (3). The actor and critic have similar network

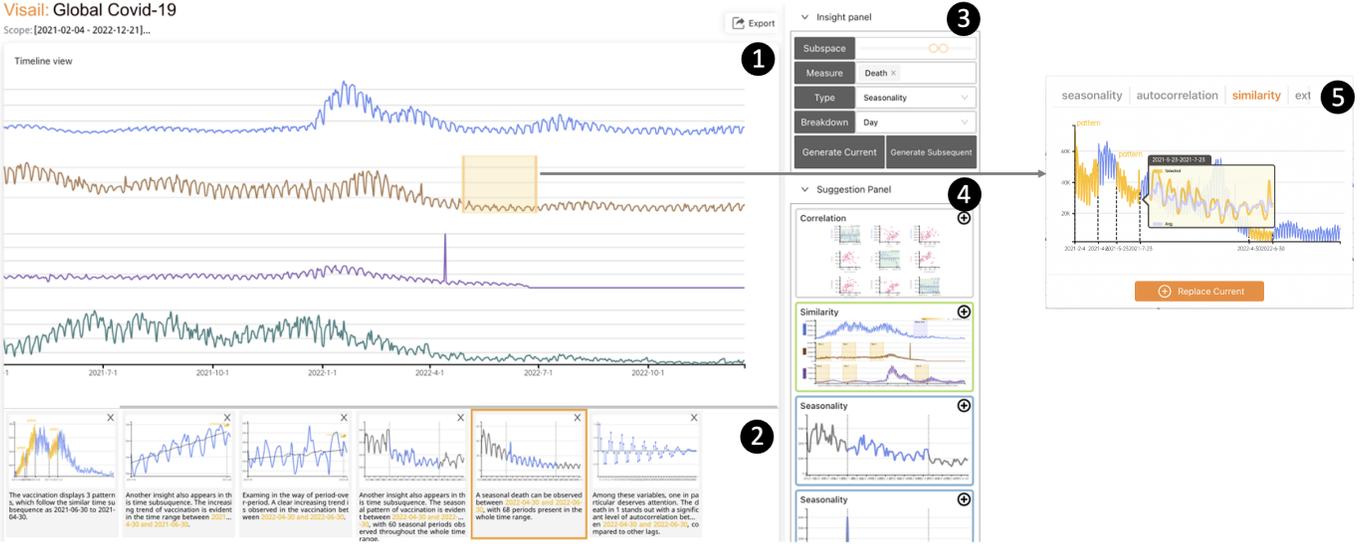


Fig. 4: The interface of Visail consists of five components, including (1) Timeline view, (2) Sequence view, (3) Insight panel, (4) Insight tooltip, and (5) Suggestion Panel.

architecture, consisting of multiple fully-connected (FC) layers. Compared with the critic, the actor has an additional output layer consisting of two components, including *semantic weight* and *mask*, which ensure the coherence of newly added insights from a semantic perspective. The first component, semantic weight, is used to indicate the logical validity of executing an action a_n^j from a set of actions $\{a_n^1, a_n^2, \dots\} \in A$ at s_n . The validity is reflected in the semantic relationship between insights before and after executing a_n^j . If this relationship is evident, the probability of corresponding action a_n^j should be increased. The relationship between the two consecutive insights can be reflected in their type and measure, which contain categorical information. Accordingly, two methods can be employed to determine the relationship: one is using the co-occurrence relationships between types and the other is through the association between variable names in measures. Thus, the probability of co-occurrence relationships between two insights (Fig. 1) and the similarity between the vectors of two variable names are considered as the semantic weight of a_n^j . The weight is then multiplied with the output of the FC layers, followed by a *softmax* activation to determine the probability p_j of a_n^j .

The second component of the actor, mask, is used to avoid selecting actions that cannot be executed at the state s_n . For example, if the measure of the n -th insight i_n at the current state contains only one variable, the action DECREASE_MEASURE should not be selected for execution. Mask was selected as it performs better than other methods in solving the problem of invalid actions, such as giving lower rewards [29]. After obtaining the action probabilities $\{p_1, p_2, \dots\}$, we first determined whether the actions can be executed for the n -th insight i_n and marked them as either valid or invalid. For an invalid action a_n^k , we set the k -th value in the mask vector to zero and multiplied the action probability by the mask to exclude the invalid actions. Thus, $\{p_1, p_2, \dots\}$ multiplied by the mask is the final output of the actor.

Finally, the agent outputs two items, the action probabilities computed by the actor and the expected return computed by the critic. Specifically, action probabilities are used to sample an action a_n , which is executed by the environment to explore the next state s_{n+1} , whose feature vector v_{n+1} is then used as the input for the LSTM-based encoder in the next iteration. Regarding the training of the agent, the goal is to make the cumulative reward $\sum_{i=n}^N \gamma^{i-n} r_i$ obtained by a_n as close to the optimal case as possible. Thus, we applied the loss function $L(\theta)$ and utilized Stochastic Gradient Descent to optimize the parameters θ .

$$L(\theta) = E[w \cdot \hat{A}_n], \text{ where } \hat{A}_n = r_n + \gamma \hat{G}_{n+1} - \hat{G}_n \quad (8)$$

where w is a dynamic weight used to limit the range of the loss and prevent excessively large updates to parameters θ , stabilizing the training process. The calculation of w is based on the probabilities of actions predicted by the actor [50]. \hat{A}_n denotes the advantage function that measures the quality of the action a_n . \hat{G}_n and \hat{G}_{n+1} are the maximum expected returns predicted by the critic based on s_n and s_{n+1} , respectively. They evaluate the maximum cumulative reward that can be obtained by generating a complete sequence after s_n and s_{n+1} . r_n is the reward obtained by the action a_n . Thus, the closer the value of \hat{A}_n is to zero, the more likely a_n is the optimal choice at s_n .

5.3.4 Online Recommendation

The design of the RL-based algorithm is well-suited to incorporate user feedback into the generation. As the algorithm automatically generates an EVA sequence, it adds a new insight at each step by executing an action with high probability. It can also generate alternative insights through the actions with lower probabilities, providing real-time recommendations to users, as shown in Fig. 3 (4). A user can select one of the alternatives to replace the current insight i_n with a new insight i'_n , changing the state of the generated EVA sequence from s_n , i.e., $\{i_1, i_2, \dots, i_n\}$ to s'_n , i.e., $\{i_1, i_2, \dots, i'_n\}$. Then, the algorithm can continuously generate the subsequent insights $i'_{n+1}, i'_{n+2}, \dots, i'_N$ based on the updated state s'_n , yielding a new EVA sequence for the user.

5.4 Representation

The representation module shows the generated EVA sequences using a set of carefully designed visualizations and provides interaction to facilitate exploration, along with accompanying textual descriptions.

5.4.1 Visualization and Textual Description

To support communicating the results generated by Visail, we represented each insight of an EVA sequence as an annotated chart along with textual descriptions. To design annotated charts, we first conducted a literature review on visualization design in terms of time series data [9, 27, 32] and annotation [9, 46, 64]. We also investigated news outlets famous for data journalism such as The New York Times and The Economist as well as well-known Business Intelligence tools such as Azure and Tableau to understand how visualization is designed for the general public. The survey from both academia and industry allows us to design annotated charts by following the design principles of simplicity, aesthetics, consistency, and contrast [43]. For example, Fig. 4 (5) shows an annotated chart depicting a similarity insight. The design highlights the visual representation of similar time subsequences

by assigning distinctive hues, as described in the derived value of the insight. When hovering over the chart, details are displayed.

Visail complements annotated charts with textual descriptions of their content, as narrative can serve as a cognitive tool for situated understanding [15, 47]. We used a template-based approach based on the formation of insight and different insight types [53]. For example, when describing a trend insight, its syntax is as follows, between $\{\{\text{subspace}\}\}$, the $\{\{\text{measure}\}\}$ is $\{\{\text{derived value}\}\}$ over time. The textual description of a trend insight, $\{\{\text{“trend”}\}, \{\{[2021-04-30, 2021-6-30]\}, \{\{\text{day}(\text{Date})\}\}, \{\{\text{vaccination cases}\}, \{\{[2275501, 2959008, \dots]\}, \{\{\text{“extent”}\}: \{\{\text{“+0.7”}\}\}\}$ is written as *“between 2021-04-30 and 2021-6-30, the daily vaccination cases increased 70% over time”*.

5.4.2 User Interface and Interaction

The user interface of Visail consists of five main components (Fig. 4): (1) Timeline view for visualizing raw time series data by variable; (2) Sequence view for displaying the generated EVA sequences, where each step is depicted as an annotated chart with a caption; (3) Insight panel that supports editing the details of a selected insight; (4) the tooltip for showing alternative insights of a selected time interval; and (5) the Suggestion Panel recommends alternative insights for a specific step in the EVA sequence.

As a user uploads a time series dataset to Visail, it is displayed in the Timeline view. In this view, the user can explore the raw data by scrolling through different variables and selecting a specific time interval. The user can also manipulate the exploration parameters such as the variables and the time range. Once the parameters are set, he or she can click the “Start” button to check the result of an EVA sequence, which is presented in the Sequence view. In the Sequence view, the user can add, delete, or exchange insights. Specifically, an insight can be edited in three different ways. First, the user can select one insight to edit its *metadata* in the Insight panel, and then click the “Generate Current” button to re-generate the current insight or click the “Generate Subsequent” button to re-generate subsequent insights for a new EVA sequence based on the modification. Second, the user can select one insight to check alternative insights listed in the Suggestion Panel and replace the current one. Third, the user can select one insight to check the time interval being observed, which is highlighted in the Timeline view. When the time interval is selected, all the insights related to it will be extracted and displayed in the tooltip. Then, he or she can add a new insight of interest to the EVA sequence. If the user is satisfied with the results in the Sequence view, he or she can click the “Export” button in the Timeline view to export.

6 EVALUATION

The evaluation of Visail includes an ablation study, a user study, and two case studies. We first conducted an ablation study to evaluate the performance of the core component of Visail (the generation module). Then, we carried out a user study to compare the quality of EVA sequences generated by Visail with that of EVA sequences created by human experts. We further conducted two case studies to understand how Visail is used by domain experts.

6.1 Ablation Study

The ablation study evaluates the performance of the generation module through three comparative experiments focusing on the RL algorithm, LSTM-based encoder, and mask, respectively. We designed a baseline for each of the three experiments. All three baselines and Visail were trained on global Covid-19 dataset (2021-02-04~2022-12-22) [4] with 100,000 steps and a maximum step of five. Following the experiment setting of the PPO algorithm [50], we trained the baselines and Visail 10 times and averaged their rewards, as shown in Fig. 5. The three experiments were conducted on a workstation with Intel i7-10700K CPU, a GeForce RTX 3060Ti GPU, and 32GB memory.

RL Algorithm versus Monte-Carlo Search Tree. Monte-Carlo search tree can effectively explore the insight space and generate data stories composed of insights [53]. Thus, it was designed as a baseline, $Baseline_{searchTree}$, for comparing against the RL algorithm in the task of generating EVA sequences. We constructed $Baseline_{searchTree}$ using

the same action space and reward function as our RL algorithm and input an identical initial insight i_1 . Given a time limit, $Baseline_{searchTree}$ explores the insight space and selects the EVA sequence with the highest reward from the search results as its output. Fig 5 shows that the average reward obtained by $Baseline_{searchTree}$ is lower than that of Visail. The results indicate that within the same time limit, Visail can generate high-quality EVA sequences more efficiently when compared to $Baseline_{searchTree}$.

LSTM-Encoder versus No Encoder To validate the effectiveness of the LSTM-based encoder in capturing the features of multiple insights as well as their sequential information, we designed a baseline, $Baseline_{no-encoder}$, where the encoder was removed from the RL agent. In $Baseline_{no-encoder}$, only the feature vector of the n -th insight i_n at the current state is used as the input for the actor and the critic [5, 52]. As shown in Fig 5, the reward of $Baseline_{no-encoder}$ increased but eventually converged to a smaller value than Visail. The results suggested that considering all the insights from the first step to the n -th step can lead to EVA sequences of high quality.

Mask versus Invalid Penalty To evaluate the effect of using a mask to avoid invalid actions, we used a penalty reward as a baseline, $Baseline_{penalty}$. During the training of $Baseline_{penalty}$, the RL agent will receive a negative reward if an invalid action is selected. Fig 5 shows that the reward of $Baseline_{penalty}$ increases at first, but converged to a smaller value than other methods, indicating that it is challenging for the agent to learn to effectively balance the rewards given the quality of the EVA sequence and the validness of the actions taken.

6.2 User Study

To evaluate the effectiveness of our algorithm in generating high-quality EVA sequences, we conducted a within-subject user study.

6.2.1 Methodology

In the user study, we compared the quality of the EVA sequences that were generated by the RL algorithm (S_{RL}), those generated by the RL algorithm without semantic weight ($S_{no-weight}$), and those created by human experts (S_{human}). Specifically, $S_{no-weight}$ was designed to investigate the impact of semantic weight on the quality of EVA sequences. Previous work suggested that DL can generate insights at a professional level of quality [17, 69]. Thus, an EVA sequence was rated in terms of intra-insight (*comprehensiveness*), inter-insight (*coherence*), path (*effective path*), and goal (*helping develop a clear goal*) qualities. We form hypotheses as follows:

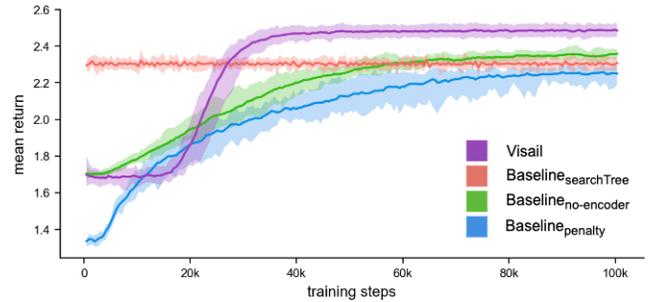


Fig. 5: Ablation study showing the mean returns of Visail and three baselines across training steps.

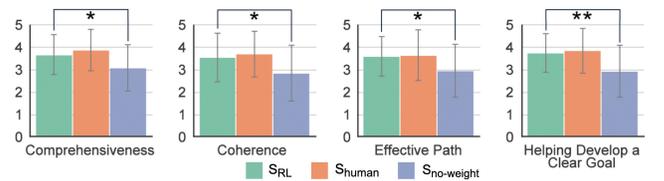


Fig. 6: Means and standard errors of each measurement in S_{human} , S_{RL} , and $S_{no-weight}$ conditions (*: $p < .05$, **: $p < .01$).

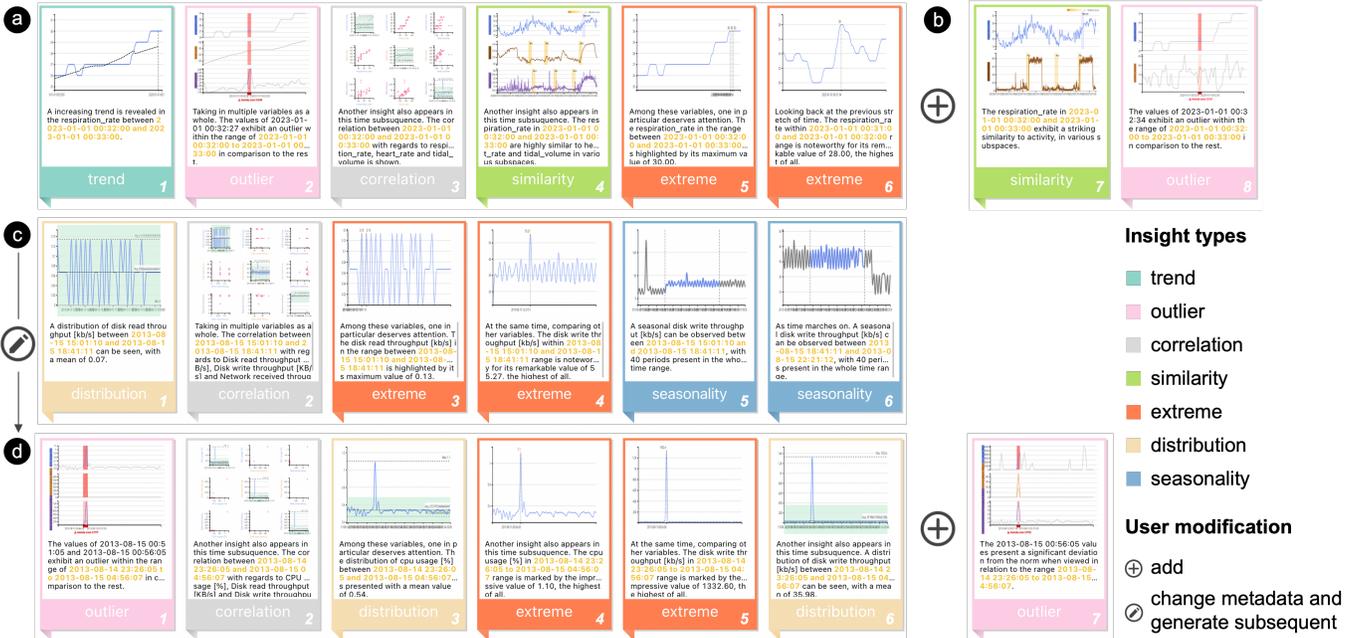


Fig. 7: Results of Case I (a)(b) and Case II (c)(d). Here, (a) and (c) display EVA sequences generated before user modification while (b) and (d) show EVA sequences generated after user modification. Specific types of user modification are highlighted.

H1 S_{RL} are perceived to be as *effective, coherent, comprehensible, and directive* as S_{human} .

H2 S_{RL} are perceived to be significantly more *effective, coherent, comprehensible, and directive* than $S_{no-weight}$.

Data and Material In each of the three conditions, S_{RL} , S_{human} , and $S_{no-weight}$, the dataset used the ablation study served as data input. Regarding S_{human} , we invited four data analysts with five or more years of professional experience to manually create EVA sequences. They used Visail without the generation module and were allowed to use any tools they are familiar with, such as Grafana, to assist their exploration. First, they identified an insight of interest as the starting point of an EVA sequence using the Insight panel. Then, they continued their exploration until the length of the EVA sequence reached six. This process resulted in a total of eight EVA sequences from four experts. To ensure a fair comparison, we used the eight initial insights selected by the experts as the input to both S_{RL} and $S_{no-weight}$ to generate another 16 EVA sequences of the same length of six. Finally, Visail visualized all 24 EVA sequences, which were divided into eight groups, with three EVA sequences sharing the same initial insight included in each group.

Task and Procedure The study began with a 10-minute introduction explaining the goal of our study and the details of the datasets. The study task required participants to read and rate the 24 EVA sequences. Each participant was presented with two groups of EVA sequences, one sequence at a time. The order of the EVA sequences was randomized to avoid learning effects. Participants were asked to rate each EVA sequence using a 5-point Likert scale and explain their ratings. The study lasted about 20-25 minutes for each participant.

Participants We recruited 20 participants (11 females) aged between 22 and 45 ($M = 26.3$, $SD = 5.0$) for our user study, including college students, researchers, and professionals from diverse backgrounds such as computer science, data science, and electronic engineering. All the participants reported that they have experience in analyzing time series and their expertise in visualization is as follows: novice: 5 (25%), advanced beginner: 3 (15%), competent: 12 (60%).

6.2.2 Results and Analysis

To evaluate **H1**, we tested the following null hypothesis: **H0**: the condition (S_{RL} or S_{human}) has no effect on the four metrics. The Wilcoxon

signed-rank test showed that S_{RL} earned similar scores with S_{human} and the differences were not significant ($p = .88, .51, .30, .70$, respectively) (Fig. 6). Thus, the null hypothesis is accepted, providing support for **H1**. To evaluate **H2**, we tested the following null hypothesis: **H0**: the condition (S_{RL} or $S_{no-weight}$) has no effect on the four metrics. Similarly, we compared S_{RL} and $S_{no-weight}$ through the Wilcoxon signed-rank test and found that S_{RL} performed significantly better than $S_{no-weight}$ ($p = .019, .016, .018, .002$, respectively) (Fig. 6). Thus, the null hypothesis is rejected, providing support for **H2**. By analyzing the quantitative feedback collected from the participants when explaining their ratings, we found that they were positive about the quality of EVA sequences generated by the RL algorithm. P11 noted, “it’s easy to understand, it first describes the distribution of confirmed and vaccination cases, then it highlights the outliers in the next insight.” The participants also suggested that the EVA sequence generated with semantic weight constituted a coherent exploration path. “It tries to find if this outlier is caused by this sudden increase in the trend of the vaccination” (P13).

6.3 Case Studies

To evaluate the usefulness of Visail, we conducted two case studies with two domain experts, who were invited to use the system to conduct visual analysis tasks on two new datasets, including physiological measurements [63] and Bitbrains datacenter traces [61], respectively. The expert (**E1**) for Case I is a rheumatologist from a hospital, with more than ten years of clinical experience. The expert (**E2**) for Case II is a data scientist from a cloud computing service team, with more than five years of operation and maintenance experience.

6.3.1 Case I: Physiological Measurements

The first case study uses a physiological measurement dataset collected when a subject performs exercises, such as heart rate and respiration rate. An EVA sequence was generated by Visail for **E1** (Fig. 7 (a)). It first shows a significant increase in respiration rate between minutes 32 and 33 (step 1). Then, outliers in respiration rate, heart rate, and tidal volume were detected (step 2) and **E1** noted a strong positive correlation between these measurements (step 3). She also observed that multiple time subsequences in heart rate and tidal volume had similar increasing trends as the time subsequence in respiration rate (step 4). Then, **E1** found that the respiration rate had reached the maximum, 30 (step 5). Finally, the EVA sequence showed the fluctuation in respiration rate

from minutes 31 to 32 when approached the maximum at 31:34 (step 6). After gaining an overview of the dataset, **E1** intended to focus on the relationship between respiration rate and activity volume. To this end, she added a similarity insight to the original EVA sequence (step 7), as shown in Fig. 7 (b). The insight shows that a few time subsequences of activity volume are similar to that of respiratory rate (minutes 32-33). She noted that the increasing trend in these subsequences also implied that the subject started exercising. Finally, by referring to the recommendation list, **E1** found that outliers also exist in the two measurements and added this insight to the EVA sequence (step 8).

After analyzing the EVA sequence, **E1** concluded that a correlation exists between respiration rate, heart rate, and tidal volume, which is consistent with medical knowledge. Also, **E1** mentioned that the time subsequence with an increasing trend in heart rate is followed by a similar increase in tidal volume. The reason might be that the increase in heart rate (oxygen consumption) leads to an increase in tidal volume (oxygen inhalation). The longer the difference in time between the two increases, the longer the body is located in hypoxia. Thus, **E1** proposed to analyze the time difference between the time subsequences with increasing trends in tidal volume and heart rate, combined with the rapid rise of measurements such as respiration and heart rates, suggesting more effective exercise plans to subjects.

6.3.2 Case II: Bitbrains Datacenter Traces

The second case study uses a time series dataset (2013-08-12~2013-08-16) from Bitbrains, which involves the performance metrics of virtual machines (VMs) such as disk read throughput and CPU usage. Visail first generated an EVA sequence, as shown in Fig. 7 (c). **E2** first noticed that the disk read throughput from 15:11 to 18:40 on 8.15 is fluctuating and hypothesized that this is likely a normal periodic fluctuation (step 1). He then moved his attention to the distribution of multiple variables including disk read throughput, disk write throughput, and network received rate (step 2), as well as the maximum values of disk read throughput (step 3) and disk write throughput (step 4). He found that these values all lie within the normal interval. Through further investigation, **E2** confirmed that these fluctuations occur periodically throughout the time series (steps 5 and 6), which verified his hypothesis. After exploration, **E2** gained a brief understanding of the VM's performance in its normal state. Also, he observed a few outliers in step 2 and noted that in step 5, the waveform of disk read throughout in the previous time period deviated significantly from others.

To further investigate the outliers, **E2** selected a new initial insight, outlier, with the time range from 8.14 23:26 to 8.15 4:56 and with measures of CPU usage, disk read throughput, and disk write throughput. After that, he clicked the "Generate Subsequent" button. This prompted Visail to generate a new EVA sequence (Fig. 7 (d)), which focused more on showing the metrics in anomalies. The EVA sequence shows outliers in steps 1 and 2. **E2** then examined steps 3 to 6 and found that the distribution and maximum values of the disk read and write throughput were located in the abnormal interval. He hypothesized that these outliers might be caused by external requests. To validate his hypothesis, he selected three variables related to external requests including memory usage, network received throughput, and network transmitted throughput to conduct further analysis. He found that all three variables were anomalous to varying degrees and added this insight in step 7. Based on the analysis of the original and updated EVA sequences, **E2** concluded: the VM is running certain services with low load under normal conditions. However, these services are not independent timed tasks and are affected by external requests that cause the VM to read and write data from/to the disk and use CPU resources. These burst requests pose a performance challenge for the VM. Thus, **E2** proposed to investigate the logs when the anomalies occurred to identify the cause of the burst requests and then optimize the performance of the running services.

6.3.3 User Feedback

Both experts from the case studies, **E1** and **E2**, provided perceptive feedback regarding generation, visualization, and interaction.

Generation. Both **E1** and **E2** agreed that the original sequences generated by Visail are reasonable and provide inspiration for data exploration and analysis. **E1** mentioned: "it's common to get an overview followed by a detailed analysis. It follows such a path of analysis. That's quite impressive." **E2** thought Visail is about to mimic human analysis behaviors, "it analyzes the association between variables by their distribution after observing outliers, and finally compares the differences of outliers in different variables."

Visualization and Interaction. Both **E1** and **E2** acknowledged the simplicity and expressiveness of the visualization design. They emphasized that the annotated charts of outlier, distribution, and extreme are intuitive and clear to communicate insights. **E1** pointed out that "although the sequence is designed in the order of the analysis steps, I prefer to arrows to indicate the analysis flow". Regarding the textual descriptions on each step of the sequence, **E2** believed that the text can help analysts to see more detailed information and play a role in promoting and assisting analysis.

7 DISCUSSION

Design Requirements for Guided EVA Tools. We found that when presented with the results generated by Visail, the participants occasionally encounter questions regarding multivariate insights, such as "beautiful visuals, cool interaction, but what does this mean?". Thus, we suggest that future guided EVA tools should increase the explainability of visual analytic recommendations using data-driven storytelling [13]. First, after analyzing the data, the experts frequently desire a narrative structure to organize the extracted insights, and further enhance engagement, communicate ideas, or persuasion [33]. Second, we found that the experts often wish to augment the interpretability of the EVA sequences by adding animation [55, 72], interaction [54], or iconic representation [56, 70]. For example, when comparing different 'outliers' in multiple EVA sequences, animation techniques such as *glow* and *swing* [55] can be added to the insight.

Generalizability of Collaborative EVA with Deep Learning. We observed that our Visail users mostly hold positive opinions toward the results generated by DL. They appreciated that DL can "relieve the workload of analysis" and "generate high-quality results". We believe that the learning-based approach can be further generalized to more data types. For example, using such an approach can support quickly generating EVA sequences for network data and telling data-driven stories in the form of chart sequences [39]. Also, Visail generates EVA sequences partially depending on the initial coding of the 80 EDA notebooks. Although such an approach can ensure the quality of the results, it limits insight types and corresponding relationships to time series data. In our future work, Large Language Models can be used to learn from human behaviors in the data exploration process to enable conversational interaction with data.

Limitations. We also observed several limitations in this work. First, both familiarity and curiosity in the reward function consider the impact of different attributes (e.g., type and measure, insight types) being the same. Our future work plans to conduct a series of user studies to investigate how analysts decide the weights when performing different visual analysis tasks. Second, the scale of a dataset can impact the results generated by the RL-based algorithm; when a dataset contains millions of time series, the resulting EVA sequence of limited length may not cover a large time range of data and thus miss certain insights.

8 CONCLUSION

In this work, we present Visail, an RL-based system that generates EVA sequences to guide the exploration of time series data. Visail presents each step of an EVA sequence as a data insight with an annotated chart and textual descriptions. Visail also allows users to modify a certain step and then updates the sequence accordingly. The results of our evaluation suggested that Visail is able to generate effective EVA sequences in terms of effective path, coherence, comprehensiveness, and goal development. Also, Visail is useful and easy to use for users who lack visual analytic expertise. Future work includes developing guided EVA tools for data-driven storytelling and deploying Visail in real data exploration work.

ACKNOWLEDGMENTS

Nan Cao is the corresponding author. This work was supported by NSFC 62061136003, 62072338, 62002267, and Huawei Cloud.

REFERENCES

- [1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer London, 2011. doi: 10.1007/978-0-85729-079-3 2
- [2] S. Alspaugh, N. Zokaei, A. Liu, C. Jin, and M. A. Hearst. Futzing and moseying: Interviews with professional data analysts on exploration practices. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):22–31, 2019. doi: 10.1109/tvcg.2018.2865040 3
- [3] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 111–117. IEEE, 2005. doi: 10.1109/infovis.2005.24 2
- [4] J. Assaker. Covid-19 global dataset. <https://www.kaggle.com/datasets/josephassaker/covid19-global-dataset>, 2022. Accessed: 2023. 7
- [5] O. Bar El, T. Milo, and A. Somech. Automatically generating data exploration sessions using deep reinforcement learning. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 1527–1537, 2020. doi: 10.1145/3318464.3389779 1, 3, 4, 7
- [6] L. Battle and J. Heer. Characterizing exploratory visual analysis: A literature review and evaluation of analytic provenance in tableau. *Computer Graphics Forum*, 38(3):145–159, 2019. doi: 10.1111/cgf.13678 1, 2
- [7] M. Behrisch, F. Korkmaz, L. Shao, and T. Schreck. Feedback-driven interactive exploration of large multidimensional data supported by visual classifier. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 43–52. IEEE, 2014. doi: 10.1109/vast.2014.7042480 2
- [8] R. Bellman. A markovian decision process. *Indiana University Mathematics Journal*, 6(4):679–684, 1957. doi: 10.1512/iumj.1957.6.56038 3
- [9] C. Bryan, K.-L. Ma, and J. Woodring. Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):511–520, 2017. doi: 10.1109/tvcg.2016.2598876 1, 2, 6
- [10] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):111–120, 2017. doi: 10.1109/tvcg.2016.2598468 1
- [11] D. Ceneda, T. Gschwandtner, and S. Miksch. A review of guidance approaches in visual data analysis: A multifocal perspective. *Computer Graphics Forum*, 38(3):861–879, 2019. doi: 10.1111/cgf.13730 2
- [12] C. Chatfield. *The Analysis of Time Series*. Chapman and Hall/CRC, 2003. doi: 10.4324/9780203491683 3
- [13] S. Chen, J. Li, G. Andrienko, N. Andrienko, Y. Wang, P. H. Nguyen, and C. Turkay. Supporting story synthesis: Bridging the gap between visual analytics and storytelling. *IEEE Transactions on Visualization and Computer Graphics*, 26(7):2499–2516, 2020. doi: 10.1109/tvcg.2018.2889054 9
- [14] Y. Chen, J. Yang, and W. Ribarsky. Toward effective insight management in visual analytics systems. In *2009 IEEE Pacific Visualization Symposium*, pp. 49–56. IEEE, 2009. doi: 10.1109/pacificvis.2009.4906837 2
- [15] Z. Cui, S. K. Badam, M. A. Yalçın, and N. Elmqvist. DataSite: Proactive visual data exploration with computation of insight-based recommendations. *Information Visualization*, 18(2):251–267, 2018. doi: 10.1177/1473871618806555 2, 7
- [16] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight. *Proceedings of the VLDB Endowment*, 10(12):1937–1940, 2017. doi: 10.14778/3137765.3137813 2
- [17] D. Deng, A. Wu, H. Qu, and Y. Wu. DashBot: Insight-driven dashboard generation based on deep reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):690–700, 2022. doi: 10.1109/tvcg.2022.3209468 2, 3, 7
- [18] Z. Deng, D. Weng, X. Xie, J. Bao, Y. Zheng, M. Xu, W. Chen, and Y. Wu. Compass: Towards better causal analysis of urban time series. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1051–1061, 2022. doi: 10.1109/tvcg.2021.3114875 2, 5
- [19] R. Ding, S. Han, Y. Xu, H. Zhang, and D. Zhang. Quickinsights: Quick and automatic discovery of insights from multi-dimensional data. In *Proceedings of the 2019 International Conference on Management of Data*, pp. 317–332. ACM, 2019. doi: 10.1145/3299869.3314037 1, 2, 4, 5
- [20] A. S. C. Ehrenberg and J. W. Tukey. Exploratory data analysis. *Applied Statistics*, 28(1):79, 1979. doi: 10.2307/2346818 1
- [21] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in Neural Information Processing Systems*, 32, 2019. doi: 10.48550/ARXIV.1901.10738 4, 5
- [22] J. Friedman and J. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23(9):881–890, 1974. doi: 10.1109/t-c.1974.224051 1
- [23] T. Fujiwara, Shilpika, N. Sakamoto, J. Nonaka, K. Yamamoto, and K.-L. Ma. A visual analytics framework for reviewing multivariate time-series data with dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1601–1611, 2021. doi: 10.1109/tvcg.2020.3028889 2
- [24] A. Gogolou, T. Tsandilas, T. Palpanas, and A. Bezerianos. Comparing similarity perception in time series visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):523–533, 2019. doi: 10.1109/tvcg.2018.2865077 2
- [25] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pp. 315–324, 2009. doi: 10.1145/1502650.1502695 3
- [26] T. Guo, S. Chang, M. Yu, and K. Bai. Improving reinforcement learning based image captioning with natural language prior. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 751–756. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1083 3
- [27] M. Hao, M. Marwah, H. Janetzko, R. Sharma, D. A. Keim, U. Dayal, D. Patnaik, and N. Ramakrishnan. Visualizing frequent patterns in large multivariate time series. In *Visualization and Data Analysis 2011*, vol. 7868, pp. 194–203. SPIE, 2011. doi: 10.1117/12.872169 6
- [28] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 159–166. ACM, 1999. doi: 10.1145/302979.303030 2
- [29] S. Huang and S. Ontañón. A closer look at invalid action masking in policy gradient algorithms. *The International FLAIRS Conference Proceedings*, 35, 2022. doi: 10.32473/flairs.v35i.130584 6
- [30] A. Key, B. Howe, D. Perry, and C. Aragon. Vizdeck: self-organizing dashboards for visual analytics. In *Proceedings of the 2012 acm sigmod international conference on management of data*, pp. 681–684, 2012. doi: 10.1145/2213836.2213931 2
- [31] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *1995 international conference on acoustics, speech, and signal processing*, vol. 1, pp. 181–184. IEEE, 1995. doi: 10.1109/icassp.1995.479394 3
- [32] N. Kong and M. Agrawala. Graphical overlays: Using layered elements to aid chart reading. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2631–2638, 2012. doi: 10.1109/tvcg.2012.229 6
- [33] X. Lan, X. Xu, and N. Cao. Understanding narrative linearity for telling expressive time-oriented stories. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pp. 1–13. ACM, 2021. doi: 10.1145/3411764.3445344 9
- [34] P.-M. Law, R. C. Basole, and Y. Wu. Duet: Helping data analysis novices conduct pairwise comparisons by minimal specification. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):427–437, 2019. doi: 10.1109/tvcg.2018.2864526 2
- [35] P.-M. Law, S. Das, and R. C. Basole. Comparing apples and oranges: Taxonomy and design of pairwise comparisons within tabular data. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, p. 1–12. ACM, 2019. doi: 10.1145/3290605.3300409 2
- [36] P.-M. Law, A. Endert, and J. Stasko. Characterizing automated data insights. In *2020 IEEE Visualization Conference (VIS)*, pp. 171–175. IEEE, 2020. doi: 10.1109/vis47514.2020.00041 2, 3
- [37] P.-M. Law, A. Endert, and J. Stasko. What are data insights to professional visualization users? In *2020 IEEE Visualization Conference (VIS)*. IEEE, 2020. doi: 10.1109/vis47514.2020.00043 2
- [38] D. J.-L. Lee, V. Setlur, M. Tory, K. Karahalios, and A. Parameswaran. Deconstructing categorization in visualization recommendation: A taxonomy and comparative study. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4225–4239, 2022. doi: 10.1109/tvcg.2021.3085751 2
- [39] W. Li, S. Schöttler, J. Scott-Brown, Y. Wang, S. Chen, H. Qu, and B. Bach. Networknarratives: Data tours for visual network exploration and anal-

- ysis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2023. doi: 10.1145/3544548.3581452 9
- [40] J. Lu, J. Wang, H. Ye, Y. Gu, Z. Ding, M. Xu, and W. Chen. Illustrating changes in time-series data with data video. *IEEE Computer Graphics and Applications*, 40(2):18–31, 2020. doi: 10.1109/mcg.2020.2968249 1
- [41] C. L. Ren, K. Ren, H. Zhang, J. Fu, and Y. Lin. PEVis: visual analytics of potential anomaly pattern evolution for temporal multivariate data. *Journal of Visualization*, 25(3):575–591, 2021. doi: 10.1007/s12650-021-00807-6 2
- [42] P. Ma, R. Ding, S. Han, and D. Zhang. Metainsight: Automatic discovery of structured knowledge for exploratory data analysis. In *Proceedings of the 2021 International Conference on Management of Data*, pp. 1262–1274, 2021. doi: 10.1145/3448016.3457267 4
- [43] M. Moshagen and M. T. Thielsch. Facets of visual aesthetics. *International Journal of Human-Computer Studies*, 68(10):689–709, 2010. doi: 10.1016/j.ijhcs.2010.05.006 6
- [44] A. Personnaz, S. Amer-Yahia, L. Berti-Equille, M. Fabricius, and S. Subramanian. Balancing familiarity and curiosity in data exploration with deep reinforcement learning. In *Fourth Workshop in Exploiting AI Techniques for Data Management*, pp. 16–23, 2021. doi: 10.1145/3464509.3464884 1, 4
- [45] V. Pham, N. Nguyen, and T. Dang. Contimap: Continuous heatmap for large time series data. In *2020 Visualization in Data Science (VDS)*, pp. 42–51. IEEE, 2020. doi: 10.1109/vds51726.2020.00009 2
- [46] D. Ren, M. Brehmer, B. Lee, T. Höllerer, and E. K. Choe. Chartaccent: Annotation for data-driven storytelling. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 230–239. IEEE, 2017. doi: 10.1109/pacificvis.2017.8031599 6
- [47] M. O. Riedl and R. M. Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010. doi: 10.1613/jair.2989 7
- [48] A. Rule, A. Tabard, and J. D. Hollan. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12. ACM, 2018. doi: 10.1145/3173574.3173606 2
- [49] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017. doi: 10.1109/tvcg.2016.2599030 2
- [50] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/ARXIV.1707.06347 5, 6, 7
- [51] K. Sedig and O. Ola. The challenge of big data in public health: An opportunity for visual analytics. *Online Journal of Public Health Informatics*, 5(3), 2014. doi: 10.5210/ojphi.v5i3.4933 1
- [52] D. Shi, Y. Shi, X. Xu, N. Chen, S. Fu, H. Wu, and N. Cao. Task-oriented optimal sequencing of visualization charts. In *2019 IEEE Visualization in Data Science (VDS)*, pp. 58–66. IEEE, 2019. doi: 10.1109/vds48975.2019.8973383 2, 7
- [53] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao. Calliope: Automatic visual data story generation from a spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):453–463, 2021. doi: 10.1109/tvcg.2020.3030403 2, 3, 4, 7
- [54] Y. Shi, T. Gao, X. Jiao, and N. Cao. Breaking the fourth wall of data stories through interaction. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):972–982, 2023. doi: 10.1109/tvcg.2022.3209409 9
- [55] Y. Shi, X. Lan, J. Li, Z. Li, and N. Cao. Communicating with motion: A design space for animated visual narratives in data videos. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pp. 1–13. ACM, 2021. doi: 10.1145/3411764.3445337 9
- [56] Y. Shi, P. Liu, S. Chen, M. Sun, and N. Cao. Supporting expressive and faithful pictorial visualization design with visual style transfer. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):236–246, 2022. doi: 10.1109/tvcg.2022.3209486 9
- [57] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications*. Springer International Publishing, 2017. doi: 10.1007/978-3-319-52452-8 2
- [58] M. Sips, P. Köthür, A. Unger, H.-C. Hege, and D. Dransch. A visual analytics approach to multiscale exploration of environmental time series. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2899–2907, 2012. doi: 10.1109/tvcg.2012.191 1, 2
- [59] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):672–681, 2019. doi: 10.1109/tvcg.2018.2865145 2
- [60] H. Tang, S. Wei, Z. Zhou, Z. C. Qian, and Y. V. Chen. Treeroses: outlier-centric monitoring and analysis of periodic time series data. *Journal of Visualization*, 22(5):1005–1019, 2019. doi: 10.1007/s12650-019-00586-1 2
- [61] TUDelft. Gwa-t-12 bitbrains. <http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains>, 2020. Accessed: 2023. 8
- [62] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: Efficient data-driven visualization recommendations to support visual analytics. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, vol. 8, p. 2182. NIH Public Access, 2015. doi: 10.14778/2831360.2831371 2
- [63] M. Vollmer, D. Bläsing, M. Nisser, A. Buder, et al. Simultaneous physiological measurements with five devices at different cognitive and physical loads. *PhysioNet*, 2020. doi: 10.13026/CHD5-T946 8
- [64] J. Walker, R. Borgo, and M. W. Jones. Timenotes: a study on effective chart visualization and interaction techniques for time-series data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):549–558, 2016. doi: 10.1109/tvcg.2015.2467751 6
- [65] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. DataShot: Automatic generation of fact sheets from tabular data. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):895–905, 2020. doi: 10.1109/tvcg.2019.2934398 2, 3, 4
- [66] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016. doi: 10.1109/tvcg.2015.2467191 3
- [67] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pp. 2648–2659, 2017. doi: 10.1145/3025453.3025768 2
- [68] K. Wongsuphasawat and B. Shneiderman. Finding comparable temporal categorical records: A similarity measure with an interactive visualization. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pp. 27–34. IEEE, 2009. doi: 10.1109/vast.2009.5332595 2
- [69] A. Wu, Y. Wang, M. Zhou, X. He, H. Zhang, H. Qu, and D. Zhang. MultiVision: Designing analytical dashboards with deep learning based recommendation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):162–172, 2022. doi: 10.1109/tvcg.2021.3114826 2, 7
- [70] S. Xiao, S. Huang, Y. LIN, Y. Ye, and W. Zeng. Let the chart spark: Embedding semantic context into chart with generative model. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–10, 2023. doi: 10.48550/ARXIV.2304.14630 9
- [71] K. Xu, J. Yuan, Y. Wang, C. Silva, and E. Bertini. Mtseer: Interactive visual exploration of models on multivariate time-series forecast. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–15. ACM, 2021. doi: 10.1145/3411764.3445083 2
- [72] L. Yang, X. Xu, X. Lan, Z. Liu, S. Guo, Y. Shi, H. Qu, and N. Cao. A design space for applying the freytag’s pyramid structure to data stories. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):922–932, 2022. doi: 10.1109/tvcg.2021.3114774 9
- [73] L. Yu, A. Lu, W. Ribarsky, and W. Chen. Automatic animation for time-varying data visualization. *Computer Graphics Forum*, 29(7):2271–2280, 2010. doi: 10.1111/j.1467-8659.2010.01816.x 1
- [74] J. Zhao, F. Chevalier, and R. Balakrishnan. Kronominer: using multi-foci navigation for the visual exploration of time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1737–1746. ACM, 2011. doi: 10.1145/1978942.1979195 1, 2
- [75] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, 2011. doi: 10.1109/tvcg.2011.195 2
- [76] J. Zhao, M. Fan, and M. Feng. ChartSeer: Interactive steering exploratory visual analysis with machine intelligence. *IEEE Transactions on Visualization and Computer Graphics*, 28(3):1500–1513, 2022. doi: 10.1109/tvcg.2020.3018724 2
- [77] Y. Zhao, Y. Wang, J. Zhang, C.-W. Fu, M. Xu, and D. Moritz. KD-box: Line-segment-based KD-tree for interactive exploration of large-scale time-series data. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):890–900, 2022. doi: 10.1109/tvcg.2021.3114865 2