Visualizing Large-Scale Spatial Time Series with GeoChron

Zikun Deng, Shifu Chen, Tobias Schreck, Dazhen Deng, Tan Tang, Mingliang Xu, Di Weng, and Yingcai Wu

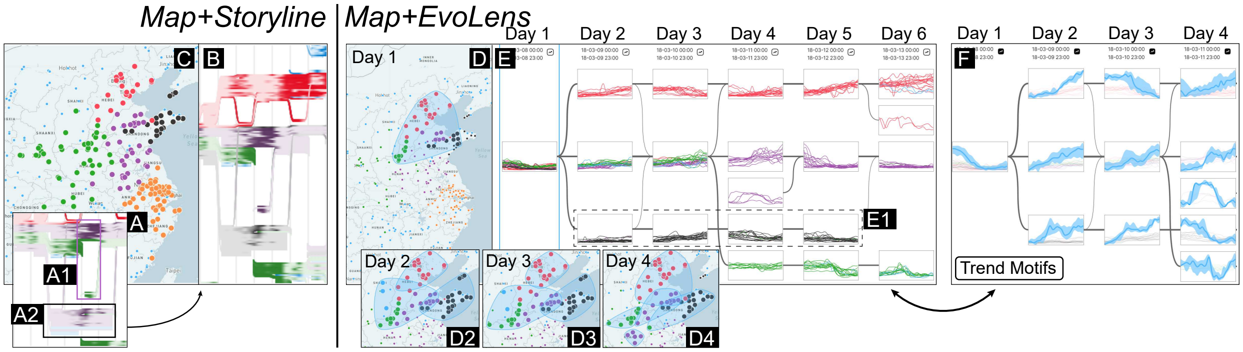

Fig. 1: Two-levenl visualizations (left and right) in GeoChron. (A) A snapshot of Storyline. (B) Entities in (A) are recolored. (C) The spatial distribution of (B). (D) The spatial distribution and EvoLens ((E) line charts and (F) trend motifs) of evolution patterns in (B).

Abstract—In geo-related fields such as urban informatics, atmospheric science, and geography, large-scale spatial time (ST) series (i.e., geo-referred time series) are collected for monitoring and understanding important spatiotemporal phenomena. ST series visualization is an effective means of understanding the data and reviewing spatiotemporal phenomena, which is a prerequisite for in-depth data analysis. However, visualizing these series is challenging due to their large scales, inherent dynamics, and spatiotemporal nature. In this study, we introduce the notion of *patterns of evolution* in ST series. Each evolution pattern is characterized by 1) *a set of ST series* that are close in space and 2) *a time period* when the trends of these ST series are correlated. We then leverage Storyline techniques by considering an analogy between evolution patterns and sessions, and finally design a novel visualization called *GeoChron*, which is capable of visualizing large-scale ST series in an evolution pattern-aware and narrative-preserving manner. GeoChron includes a mining framework to extract evolution patterns and two-level visualizations to enhance its visual scalability. We evaluate GeoChron with two case studies, an informal user study, an ablation study, parameter analysis, and running time analysis.

Index Terms—Spatiotemporal visualization, spatial time series, Storyline

1 Introduction

Spatiotemporal phenomena (e.g., traffic conditions, air pollution, rainfall, and temperature) are continuously monitored by geo-referred sensors, generating large-scale spatial time series (hereafter "ST series") in many domains, such as geography [15], atmospheric science [24, 46], and urban informatics [9, 27, 70]. ST series visualization is one of the important means of understanding spatiotemporal phenomena.

Traditionally, ST series are first depicted in temporal visualizations, e.g., line charts. These visualizations are then either plotted on a map by their geographic positions [54, 65], or displayed in a separate view that is coordinated with a map [43], and thereby can be related back to the geographic context. The above methods, considered as the strategy of direct depiction [37], are not effective for large-scale ST series. An analyst may find it difficult to browse the spatial distribution in the vast space and diverse, dynamically changing trends over a long time to gain insights (e.g., trends, correlation, and anomalies). Another strategy is to transform ST series by summarizing them into summaries [19, 42] (e.g., variations of consecutive records [24]) or by extracting patterns from them [10, 13] (e.g., co-occurrence patterns [23]). These summaries or patterns, rather than massive raw ST series, are then visually exposed. Yet, the trend narrative and dynamics of ST series, the fundamental features of ST series data, are broken into summaries [19] or patterns [9, 23]. Neither patterns nor summaries are depicted over time. Analysts cannot intuitively perceive how the recorded phenomena develop in the geographic space over time. For example, how quickly does air quality in an area deteriorate/improve? How long is the time before it deteriorates again? Is the deteriorating area expanding?

Each ST series is not isolated in space and time; usually, spatially close ST series exhibit correlated trends within a time period when the recorded phenomenon evolves over the region within the time period [13, 23, 76]. For example, when a factory emitted air pollutants, correlated upward trends would be observed in the ST series of air quality sensors nearby the factory. We define the above observation, i.e., *spatially close ST series* exhibit correlated trends in *a time period*, as an *evolution pattern*. These patterns offer the possibility that large-scale ST series can be visualized in a pattern-aware and narrative-preserving manner. First, visualizing these patterns enables users to view multiple ST series meaningfully at once rather than scan many ST series to derive knowledge in their minds. Second, the ST series data are continuous between evolution patterns in adjacent but not overlapping periods. Visualizing such patterns along the timeline can retain the temporal narrative of the trends and dynamics of the involved ST series. Thus, we study how to visualize large-scale ST series based on the notion of evolution patterns and thereby support the analysis of large ST datasets.

The Storyline technique [30, 50] is well-suited for solving the above visualization problem. Originally, the Storyline was designed to visualize the dynamic relationships between entities and applied to reveal the

- *Z. Deng and Y. Wu are with State Key Lab of CAD&CG, Zhejiang University. E-mail: {zikun_rain, ycwu}@zju.edu.cn.*
- *S. Chen, D. Deng, and D. Weng are with School of Software Technology, Zhejiang University. E-mail: {sfchen, dengdazhen, dweng}@zju.edu.cn. D. Weng is the corresponding author.*
- *T. Tang is with School of Art and Archaeology, Zhejiang University. E-mail: tangtan@zju.edu.cn.*
- *T. Schreck is with Graz University of Technology. E-mail: tobias.schreck@cgv.tugraz.at.*
- *M. Xu is with a) School of Computer and Artificial Intelligence, Zhengzhou University, b) Engineering Research Center of Intelligent Swarm Systems, Ministry of Education, and c) National Supercomputing Center in Zhengzhou. E-mail: iexumingliang@zzu.edu.cn.*

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

development of a story, movie, or meeting. For example, Liu et al.'s Storyline [30] depicts characters' or entities' interaction in the same scene over time in a movie. Each entity is represented as a curve along a horizontal timeline. Curves are bundled together to form a session only if the entities are related to each other in the period. By drawing an analogy (Fig. 2) between an evolution pattern and a session (or an ST series and an entity), the Storyline can present evolution patterns over time in a well-organized way and thereby visualize large-scale ST series. However, two challenges must be addressed to apply the Storyline technique to visualize **large-scale** ST series:

Mining of evolution patterns. Evolution patterns should capture implicit spatiotemporal relations between ST series. Any two ST series in a pattern should 1) have correlated temporal trends during the time period of the pattern, and 2) have close geographic positions because a meaningful correlation tends to exist between the ST series close in space. Yet, the temporal trends and spatial context are heterogeneous with different scales and semantics. Fusing them to capture the spatiotemporal relation is non-trivial, let alone to mine the patterns where multiple ST series should have many such relations with each other.

Presentation of spatiotemporal information. The traditional Storyline can neither display the fine-grained temporal trends nor provide the spatial context for ST series analyses. Previous studies [17, 62] attempted to extend Storyline techniques to spatiotemporal scenarios but cannot accommodate large-scale datasets with many locations and long time ranges. The Storyline occupies the most effective visual channel (i.e., position) and is compact in layout. Thus, popular temporal visualizations (e.g., line charts) and spatial visualizations (e.g., a map) are difficult to integrate.

We propose GeoChron (**chron**icle of **geo**-space), an interactive visualization for large-scale ST series. For the first challenge, GeoChron includes a carefully-designed data mining framework. The framework slices the time span and captures reliable correlation relations between ST series in each time slice. Afterward, it employs a network formulation to fuse the correlation relations above and the spatial proximity in a semantics-preserving way and detect communities as evolution patterns. Consequently, the Storyline layout algorithm can be applied by considering each ST series as an entity and considering an evolution pattern as a session. For the second challenge, GeoChron supports a novel two-level visualization mechanism to present the spatiotemporal information of large-scale ST series. At the first level, we revise the Storyline with new interactions and visual encodings, and link it with a geographic map, to present evolution patterns from a high level of perspective. At the second level, we employ an *EvoLens*, a lens placed on the Storyline, to display more details regarding the temporal trends of evolution patterns in a narrative-preserving manner. The map is coordinated with the EvoLens to provide the detailed spatial context of evolution patterns. GeoChron is comprehensively evaluated by a series of studies and analyses. In sum, our contributions are as follows:

- We propose a **technique** called GeoChron by leveraging the evolution pattern notion to meaningfully organize large-scale ST series into spatiotemporal partitions and visualize large-scale ST series in a pattern-aware and narrative-preserving way.
- We develop a **data mining** framework to extract evolution patterns (i.e., sessions) from large-scale ST series.
- We design a two-level **visualization** mechanism based on the classic Storyline representation to support effective visual analysis of large-scale ST series in a spatiotemporal context.
- We demonstrate the successful **application** of GeoChron through two real-world cases, which expands Storyline's impact to the field of big spatiotemporal data analysis.

2 Related Work

This section presents prior studies of visualization of time series, visualization of spatiotemporal data, and Storylines.

In the **visualization of time series**, the temporal trend is undoubtedly one of the important features [45, 74]. Temporal trend analysis is common in the stock market [16, 69], energy [29], and radio [73] fields, and even many geography related fields like meteorology [38, 46, 68], environment [31, 41, 64], and urban sciences [13, 59]. There are plenty of excellent visualizations for time series [2]. This section narrows down the focus to the most related part, i.e., visualization of ST series.

ST series bring their own challenges in visualization. Analysts need to relate temporal trends of (usually multiple) ST series to the spatial context. Geographic maps were used in all visualizations of ST series without exception. Following Andrienko's study on geovisualization [3], visualization of ST series can be classified into three types, namely, direct depiction, summarization, and pattern extraction.

Direct depiction. The direct depiction is the most straightforward and intuitive way. ST series can be plotted on a geographic map by their geographic positions [75]. Such a kind of visualization can be naturally extended to 3D environments, where the timeline was placed along the z-axis [54]. Additionally, ST series can be visualized separately from the map [43] to avoid visual occlusion and then can be related back to the map through user interactions.

Summarization. In tradition, ST series are commonly summarized into multiple heatmaps by time, and each heatmap presented the spatial distribution of the recorded readings in a time interval [31, 38, 64]. For example, Meshesha et al. [36] used four heatmaps to depict the distribution of dissolved oxygen in a river over four seasons. Visualization researchers designed visualizations to summarize ST series into more in-depth features [4, 19, 20, 24]. For example, Li et al. [24] derived and visualized variations in recorded climate data to discern climate changes. Clustering can also help summarize ST series [60]. The ST series of the same cluster can be visually summarized by a representative one because they are similar [74]. Yet, existing methods treated spatial and temporal dimensions separately [60] and thus cannot capture the spatiotemporal relationship that shifts in both time and space. Besides, considering the whole time span may ignore local features. To this end, the sliding window strategy is helpful [1, 40]. A window slides along time, and its size is far less than the series' length. Clustering is performed on the parts of time series covered by the window.

Pattern extraction. ST series exhibit various patterns, e.g., correlation [35], co-occurrence [61], propagation [25], cascading [10], and causality [76], which are difficult to comprehend due to inherent uncertainty, heterogeneity, or dynamics [11]. Visual analytics solutions have been developed to assist analysts in interactively exploring and visually reasoning these valuable patterns in spatiotemporal contexts [9, 13, 23].

If large-scale ST series are depicted directly, analysts may find it hard to gain insights into data due to the sheer number of ST series to be investigated. By means of summarization or pattern extraction of large-scale ST series, analysts may not easily follow the temporal trends and dynamics of ST series between summaries or patterns. We define the evolution pattern in ST series and develop a framework to detect the patterns. Visualizing these patterns over time can retain the temporal trends of ST series. We aim to propose an evolution pattern-aware and narrative-preserving visualization for large-scale ST series.

Visualization of spatiotemporal data can be classified into two main categories, namely integrated view and linked views, based on the composition of space and time. On the one hand, an integrated view presents spatial and temporal information in a tightly integrated manner. The integration way is case-by-case. For example, the temporal information can be embedded into the map [48] or tightly placed around the map [24]. Glyph-based designs [14, 66, 67] are commonly used to integrate spatial and temporal information. In Zeng et al.'s glyph [71], the temporal distribution of inflow and outflow in a region was wrapped on a pie chart that counted the POI classes in the region. On the other hand, linked views adopt two (sometimes multiple) coordinated visualizations to present spatial and temporal information, respectively [6, 22, 28, 29, 63, 72].

Our design employs linked views. The Storyline is already compactly laid out and takes up many effective visual channels. Thus, the geographic information is hard to integrate.

Storylines have been applied to various scenarios, such as software analysis [39], compound event reviewing [32], fiction presentation [30], video moderation [53], and meeting recalling [44]. Initially, Storylines were generated by hand. To automate the generation, researcher developed many algorithms, such as, based on genetic algorithms [50] or hybrid optimization [30]. Tang et al. [52] realized that a hand-drawn storyline is more aesthetically pleasing and developed an authoring tool iStoryline with a new layout optimization framework. Most recently, Tang et al. [51] improved iStoryline via reinforcement learning.

In the above studies, the relationships between entities are explicit. For example, two entities have a relationship if the characters of entities appear in the same scene [30]. By contrast, we need to model implicit relationships between large-scale ST series and construct sessions. Yagi et al. [62] made an initial attempt on a small dataset with a few ST series and short-term observations. Their approach ignored the spatial context and the temporal sensitivity of relationship modeling, and might require tedious processes with trial and error to tune session generation.

ST series are also associated with the spatial context. Yagi et al. [62] used colors to visually link the curves in Storyline with the locations on a geographic map. Such a manner is not scalable. Hulstein et al. [17] summarized and compared different ways to combine the spatial context and Storyline. Although their study did not focus on large-scale data, their findings provide useful insights into effective visual designs.

3 Overview

This section presents the overview of our study.

3.1 Term Definition

Locations $\mathscr{L} = \{l_1, l_2, \dots\}$ are sensors (e.g., air quality monitoring stations) deployed in geographic space. Each location has a geographic coordinate. A **time period** is a data structure p that has a start timestamp $p.t_s$ and an end timestamp $p.t_e$, i.e., $p = (t_s, t_e)$. In Sec. 4, we will introduce two concepts, **time slice** and **time window**. They both have such a structure. **Spatial time series (ST series)** $\mathscr{V} = \{V_1, V_2, \dots\}$ is a set of time series collected in locations $\mathscr{L}$. The ST series $V_i = \{v_{i,1}, v_{i,2}, \dots\}$ comprises chronological records in the location l_i. These records are collected at a series of regular timestamps. We use $V_i(p)$ to denote the subpart of an ST series V_i that belong to the time period p from $p.t_s$ to $p.t_e$, i.e., $V_i(p) = \{\dots, v_{i,t}, \dots\}, p.t_s \leq t \leq p.t_e$. An **evolution pattern** is characterized by 1) a group of ST series $\mathscr{V}_g \subset \mathscr{V}$ geographically close, and 2) a time period p such that $\{V_i(p) | V_i \in \mathscr{V}_g\}$ have correlated trends. We say the ST series in an evolution pattern have **spatiotemporal relations** with each other.

3.2 Background and Research Problem

This study resulted from a long-term experience in spatiotemporal analysis. We have collaborated with many experts in environmental science, geography, and urban computing fields in various projects. We found that although many visual analytics methods have been proposed for ST series analysis [9, 10, 13, 23], there is still a lack of an intuitive and effective visualization to display large-scale ST series to provide the awareness or understanding of data. As mentioned in Sec. 2, ST series have two fundamental features to visualize, namely, the **temporal trend** and **spatial context**. When dealing with large-scale ST series, analysts have a vast geographic space to explore, and many long time series to browse. Existing visualization methods become limited. They are either unable to highlight meaningful knowledge and avoid the intractable seeking process of information in numerous ST series by analysts, or cannot clearly display the inherent temporal trends and dynamics of ST series. Instead of a complicated application, this paper studies a fundamental problem, i.e., effectively visualizing the **temporal trends** of **large-scale ST series** with a **spatial context**.

We formed a team to study this problem, including visualization researchers as well as three spatiotemporal analysis experts. Each of the three experts has at least five years of experience in spatiotemporal data analysis. The target users, including the experts, are any analysts who need to review, monitor, or analyze ST time series. The team participates in iterating the design. As we are studying a general and fundamental visualization problem, no additional experts in the specific fields are involved in design iteration and evaluation.

According to our observations, each ST series is not isolated and is oftentimes correlated with others nearby due to the influence of certain events, exhibiting the evolution patterns defined above. Evolution patterns serve as an effective and meaningful entry to analyze multiple ST series at once. Furthermore, analysts can recover the trends of ST series by examining the patterns in two adjacent time periods if the patterns are placed over time. In this way, the narrative of temporal trends and dynamics is preserved between adjacent evolution patterns. Finally, evolution patterns may imply causation since they are built on correlation. Visualizing these patterns can aid in further analysis.

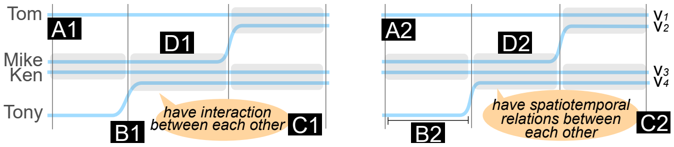
Tom
Mike
Ken
Tony
A1
D1
B1
C1
have interaction between each other
A2
D2
B2
C2
have spatiotemporal relations between each other
V1
V2
V3
V4

Fig. 2: Analogy illustration. (left) Telling a story with Storyline. (right) Visualizing the evolution patterns of ST series.

By now, our research problem has been specified on how to present evolution patterns in large-scale ST series in a narrative-preserving manner and thereby visualize the ST series effectively.

3.3 Storyline-based Solution

The Storyline is favored for its narrative representation of dynamic relationships [30]. We introduce the core concepts of the Storyline based on its popular application in the storytelling of films.

- *Entity*. An entity refers to a character in a story and is represented as a curve that progresses from left to right (Fig. 2A1).
- *Relationship*. Two characters (i.e., entities) have a relationship if they interact with each other at the same scene (Fig. 2C1), and vice versa. The relationship between two characters can evolve over time according to the story's progression.
- *Time Frame*. A time frame is when the relationship between any two entities changes (Fig. 2B1), which depends on the evolution of a known story. Therefore, the interval between timestamps can be uneven and irregular.
- *Session*. A session is defined as the interaction of multiple entities between two adjacent time frames (Fig. 2D1).

We develop GeoChron adapted from the Storyline to address the research problem in the following two stages.

Transforming ST series into sessions. We first draw an analogy between an entity and an ST series (Fig. 2A2). In this way, a session where entities have spatiotemporal relations with each other in the time slice due to a certain event (Fig. 2C2) is exactly an evolution pattern (Fig. 2D2). Leveraging the analogy, the Storyline offers a narrative-preserving manner to visualize evolution patterns. The **first challenge** is that the spatiotemporal relations between ST series are implicit and dynamic in both space and time. Determining the relations effectively is non-trivial, let alone detecting the complex sessions with multiple such relations. To address this, GeoChron includes a data mining framework to detect evolution patterns from large-scale ST series.

Incorporating spatiotemporal visualizations. Evolution patterns are well organized in a narrative way by analogy to sessions in a Storyline. The **second challenge** is the limited space in the compact Storyline hinders the embedding of the spatiotemporal visualization of ST series. In addition, Storyline occupies the visual channel of height, which is originally used to reveal the temporal trend in line charts. To this end, GeoChron includes a novel two-level visualization mechanism based on the Storyline to encode the spatiotemporal information from overview to details.

4 Pattern Mining

This section presents the mining framework in GeoChron for detecting evolution patterns, i.e., sessions, from large-scale ST series. The framework should 1) capture stable and reliable correlation relations, and 2) fuse the geographic context and temporal features.

Step 1: Slicing Time Span. The first step is to transform the continuous temporal dimension into discrete slices. After detecting sessions for each slice, we can apply Storyline. Our framework divides the time span evenly into T slices, denoted as $\{s_1, \dots, s_T\}$, with equal size (e.g., Fig. 3A). Each time slice $s = (t_s, t_e)$ has a time period structure. The slice size depends on the domain-specific scenario. For example, air pollution data are often reviewed on a daily basis [55], and the size of one day is appropriate. A smaller size is not enough to capture prominent features, and a larger size may miss the dynamics. Afterward, Steps 2-4 below are executed for each time slice.

Step 2: Determining Correlation in a Time Slice. Pearson correlation coefficient is the most widely used correlation metric. The

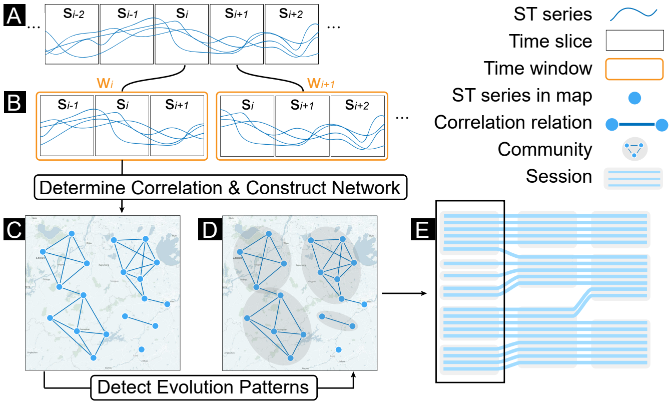

Fig. 3: Framework of evolution pattern mining. (A) The time span is divided into time slices. (B) A time window wraps three consecutive time slices. (C) The spatiotemporal relations in a time slice are determined based on the time window and constitute a relation network. (D) The communities in the relation network of (C) form six evolution patterns. (E) The patterns in (D) are depicted as sessions in a Storyline.

correlation between two ST series V_a and V_b in a time slice s_i can be straightforwardly computed as $\mathbf{r}(V_a(s_i), V_b(s_i))$, the Pearson correlation coefficient between their subparts within the time slice. V_a and V_b are correlated if $\mathbf{r}(V_a(s_i), V_b(s_i))$ is larger than a given threshold th_r. However, the coefficients can be unstable across adjacent time slices, e.g., due to subtle and local event effects, or data measurement errors. For example, the coefficients between V_a and V_b in s_{i-1}, s_i, and s_{i+1} could be 0.75, 0.45, and 0.8, respectively. If $th_r = 0.7$, V_a and V_b becomes correlated in s_i abruptly. An individual line may suddenly break away from a session and then return. As a result, the produced Storyline may be hard to read and even meaningless. To capture prominent patterns and produce readable visualization, we need to smooth out the correlation relationships between ST series over time. However, averaging these coefficients is not advisable because they cannot be added or subtracted arithmetically.

Thus, we use a wrapping window strategy based on a sliding window. A set of wrapping windows $\{w_2, \dots, w_{n-1}\}$ is generated by sliding a three slice-sized window along time. Each one wraps one focus slice and two adjacent slices and also has a time period structure. In Fig. 3B, the time window w_i wraps s_{i-1}, s_i, and s_{i+1}. The correlation relation between V_a and V_b in a time slice s_i exists if $r_{a,b,i} = \mathbf{r}(V_a(w_i), V_b(w_i)) > th_r$ holds true. If $r_{a,b,i}$ is still a sudden compared to $r_{a,b,i-1}$ and $r_{a,b,i+1}$, we smooth it out by replacing it with $r_{a,b,i-1}$ or $r_{a,b,i+1}$.

In this way, we consider more observations in adjacent time slices, while maintaining the analysis at the time slice granularity. If the records in s_i are uncorrelated due to subtle effects, $r_{a,b,i-1}$, $r_{a,b,i}$, and $r_{a,b,i+1}$ tend to be true since the records in s_i are only one-third of the records they consider. If $r_{a,b,i}$ is false due to prominent effects in multiple slices wrapped by w_i, $r_{a,b,i-1}$ and $r_{a,b,i+1}$ tend to be false, because $\mathbf{r}(V_a(w_{i-1}), V_b(w_{i-1}))$ and $\mathbf{r}(V_a(w_{i+1}), V_b(w_{i+1}))$ consider two-thirds of the records that $\mathbf{r}(V_a(w_i), V_b(w_i))$ considers. We do not adopt a sliding window whose size varies adaptively. Considering too many observations may de-emphasize the correlation/uncorrelation under a certain time slice and produce inaccurate results.

In this step, we determine the pairwise correlation relations for all ST series pairs in a time slice.

Step 3: Modeling the Relation Network in a Time Slice. For each time slice, we construct a relation network based on the correlation relations (Fig. 3C). We introduce a threshold th_d to filter out those distant pairs of ST series. Meaningful correlations tend to exist between the observations in close locations. After filtering, the remaining pairs of ST series naturally constitute a relation network, where connected nodes (ST series) are close in space and correlated in the time slice.

Such a network-based formulation fuses the temporal correlation and geographic context well. Assigning weights to each aspect and summing them to form a distance or similarity measure for clustering may be a straightforward approach. Yet, the two aspects have distinct scales and semantics, making weight tuning difficult, and the summation lacks semantic clarity. By contrast, our formulation preserves the semantics in time and space respectively to construct a relation network.

Step 4: Detecting Evolution Patterns. ST series belong to the same session if they are correlated in the time slice and are close in geographic space with each other. Such ST series exactly form a community in the relation network. Therefore, we apply the Louvain algorithm [5], a well-established method, to detect communities. The Louvain algorithm works on the network structure and does not require hyperparameters. Each ST series can only belong to one community. Fig. 3D shows six communities detected from the network of Fig. 3C. The ST series enclosed by a gray ellipse belong to the same community.

An ST series has correlation relations with most other ST series in the same pattern. In this way, patterns' geographic ranges are not limited by th_d. Two remote ST series can be in the same pattern if they both have spatiotemporal relations with most others in the pattern.

Multiple communities (e.g., Fig. 3D) are detected for each time slice by the framework. Communities are then regarded as sessions (e.g., Fig. 3E) in the Storyline. The detection module has the time complexity of $O(|\mathscr{L}|^2 T)$. $|\mathscr{L}|^2$ comes from the Louvain algorithm [56].

5 Original Storyline Layout Method

Among existing Storyline layout methods, we choose StoryFlow [30] because it is easily implemented and widely extended by recent studies [51–53]. StoryFlow has the time complexity of $\mathscr{O}(|\mathscr{L}|^2 T + T^3)$. Below, we briefly introduce StoryFlow [30].

Step 1: Ordering. StoryFlow [30] first transforms the sessions into a layered graph. Each time slice is viewed as a layer and each session as a node. An edge exists between nodes if the two sessions have at least one entity in common. StoryFlow [30] formulates the problem of ordering sessions across time slices as finding the layered layout of the graph with the least number of edge crossings. A good layout of the layered graph can be obtained by the classic Sugiyama's algorithm [47].

Step 2: Aligning. This step aims at minimizing the wiggles of curves by aligning sessions and entities between adjacent time slices. StoryFlow [30] formulated the problem of aligning the sessions in s_i and s_{i+1} respectively as a weighted Longest-Common-Subsequence (LCS) problem [7]. StoryFlow tends to align two sessions if they have sufficient entities in common while trying to maximize the sum of common entities over all pairs of aligned sessions. After the sessions are aligned, the entities can be aligned by finding the common entities.

Step 3: Positioning. Two adjacent entities in the same session should have a space d_{in} between each other, and two adjacent sessions should have a space d_{out} between each other (Fig. 4E). $d_{out} > d_{in}$. Aligned entities in the aligned sessions should have the same y-positions and can be rendered with straight lines. Sessions and entities should be compactly positioned under the constraints above. StoryFlow's positioning method based on the quadratic programming [30] usually runs out of memory for large-scale data [49]. Thus, we adopt a heuristic implementation of StoryFlow's positioning method to position entities by sweeping time slices back and forth.

First, the entities in s_0 are positioned from top to bottom one by one. Second, s_i can be a position reference for s_{i+1} (sweep forth). An entity in s_i can determine the position of its aligned entity in s_{i+1} and the session that contains this aligned entity. If the remaining entities in s_{i+1} cannot be inserted under the constraints above, expand the space between the aligned entities in s_{i+1} so that all entities in s_{i+1} can be positioned (Fig. 4A). Entities in $s_i, s_{i-1}, \dots, s_0$ need to be repositioned (sweep back) with the reference of s_{i+1}.

6 Visual Design

This section presents the design goals and visual designs of GeoChron.

6.1 Design Goals

In most cases, target users may not necessarily have expertise in visualization and visual analytics. Thus, GeoChron should employ intuitive and easy-to-understand designs based on the Storyline representation to facilitate their understanding and interpretation of the large-scale data.

We conduct an iterative design process. In each iteration, we develop a prototype and perform tentative explorations on an air quality dataset (Sec. 8.1.1) via this prototype. Then, we summarize the observed shortcomings and then improve the prototype for the next iteration. After multiple iterations, we conclude five design goals as follows.

G1 **Enable session-based analysis**. Evolution patterns serve as entries for analyzing multiple ST series that are close and correlated. Since every pattern is represented as a session in the Storyline, the visualization should enable interactive analysis based on sessions.

G2 **Support effective tracking**. The correlation relationship between ST series is dynamic and thereby evolution patterns present complex spatial and temporal relationships between each other. Therefore, the design should implement flexible interactions for tracking the evolution patterns as well as ST series in the Storyline.

G3 **Visualize temporal trends**. Time series trends are the basic features in ST series [74]. Trends can facilitate the locating of interesting periods, e.g., when the readings of multiple ST series suddenly rise or fall. Besides, trends help users interpret dynamic correlation relations, e.g., why two ST series are correlated. Hence, the design should visually display the trends of time series over time.

G4 **Link spatial and temporal information**. The spatial information of ST series is important for understanding evolution patterns. A geographic map is difficult to integrate into the compact and already information-dense Storyline. Therefore, linking the spatial and temporal information is more suitable than integrating them.

G5 **Provide multi-level analysis workflow**. Large-scale ST series can exhibit rich information, e.g., the time series trends over a long time, spatial distribution on a vast space, and many-to-many spatiotemporal relations. Multi-level analysis is a well-known effective manner for accommodating a large amount of information. Thus, the design should support a multi-level analysis workflow.

The final design satisfies these goals to effectively visualize large-scale ST series. Below, we introduce the design from two aspects, namely, layout refinement and two-level visualization.

6.2 Layout Refinement

First, we need to further refine the Storyline layout generated by Sec. 4 so that it can compactly and effectively display large-scale ST series.

Loose Alignment. Feeding numerous sessions into the Storyline layout algorithm above may produce a hard-to-read representation. The sessions sparsely distribute vertically and users have to scroll a lot to browse them. The main reason is that when two aligned entities are straightened, the straightening process pushes other sessions up or down entirely over many time slices (Fig. 4A), consuming much vertical space. StoryFlow's solution [30] also has this issue. To alleviate this issue, we discard those session pairs whose number of entities in common is less than a threshold th_c. For example, compared with Fig. 4B1, Fig. 4B2 is more space-efficient since the curve pointed by the red arrow is not aligned.

Session Filtering. Numerous sessions are overwhelming. Thus, we filter out the session if 1) its size is less than a threshold th_s and 2) none of the entities it comprises goes to the sessions with a size $\geq th_s$ in the two neighboring slices. In this way, Fig. 4B2 is transformed to Fig. 4B3. Not only the overwhelmingness is alleviated, but also the vertical space is further saved.

Curve Hiding. Wiggles cannot be totally avoided. What is unbearable is the up-and-down movement of the curve caused by wiggles. Specifically, these nearly vertical parts of the curve lead to visual clutter and break the visual continuity of aligned sessions (G2). Such an issue is particularly bad when the wiggle distance is large. To this end, we hide the main vertical part of a curve if the wiggle distance is larger than a threshold th_w (Fig. 4C). The parts where the wiggle starts and ends are left to reveal where the curve goes.

These methods can improve readability but also bring different degrees of information loss. Users need to make a trade-off between information completeness and readability on demand. Moreover, the readability should be perceived by users. For example, how much do the curves affect the visual continuity? How much do users have to scroll to browse patterns? Do the patterns overwhelm users? Therefore, GeoChron allows users to specify the above parameters via sliders.

Enforce Alignment of Sessions. Previous methods [30] enable users to straighten the curve of an entity they are interested in. Without visual wiggles, users track the entity's evolution more easily. We extend this idea from an entity level to a session level (G1) by enabling straighten multiple entities as much as possible for ST series tracking (G2).

After a user clicks a session, for each time slice, the session with the largest intersection with the clicked session is called the target session. Then, Step 2: Aligning will be re-run as follows. For each of two adjacent time slices s_i and s_{i+1}, we first align the two target sessions in s_i and s_{i+1} if the loose alignment constraint is satisfied. Afterward, the sessions in s_i (and in s_{i+1}) are divided into a top part and a bottom part by the target session. We use the LCS-based method [7] to align the two top parts and two bottom parts, respectively. For example, after the session 2 in Fig. 4D1 is clicked, the layout is transformed to Fig. 4D2.

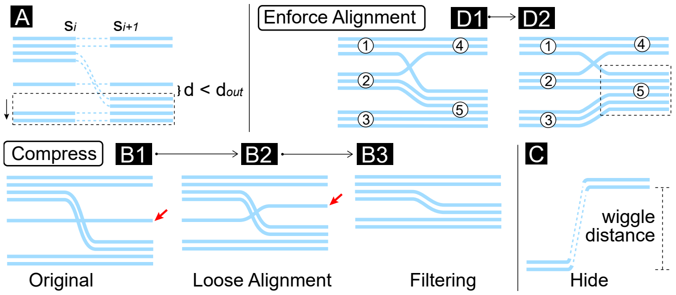

Fig. 4: Layout refinement. (A) Vertical space consumed by the positioning process. (B) The layout after loose alignment and session filtering. (C) Curve hiding. (D) Enforced alignment of sessions ② and ⑤.

6.3 Two-level Visualization

GeoChron employs a two-level visualization mechanism (G5) to present evolution patterns in a spatiotemporal context.

6.3.1 Tracking Overall Evolution Patterns

At the first level, we modify the visual encodings in the Storyline and add new encodings to the Storyline to present evolution patterns from a macro perspective (Fig. 5A and D). We still call it Storyline, since it retains the visual identity of the classic Storyline. Users can obtain an overview of ST series and locate a time period or patterns of interest.

Visualizing time series trends (G3). Traditionally, time series are visualized with line charts, and the trends are reflected by the heights of lines. However, in the Storyline, the visual channel of vertical positions is occupied with encoding the relationship among ST series over time. Finally, we encode the trends with shades of curves.

Here, we take the blue color for illustration. We first select two shades of blue, light blue and dark blue, and generate a linear interpolation between them. Then, users can specify a piecewise linear mapping function $shade(v)$ via the user interface of Fig. 5B. $shade(v)$ can map each value v in a time series to a shade value between 0 and 1. 1 corresponds to the dark blue shade and 0 corresponds to the light blue shade. Subsequently, the shade value is used to generate a color that falls between the light and dark shades of blue. For example, given the mapping of Fig. 5B, higher values have colors with darker shades. For each curve C_i, we evenly sample a series of values from V_i and obtain the corresponding colors. We finally generate a gradient color stroke for C_i using these colors to reflect the trends of the time series.

Moreover, we set d_{in} (the space between the adjacent entities in a session) as 0. In this way, the trends of the ST series in a session can be clearly revealed (G1). For example, the session in Fig. 5A5 clearly indicates that multiple ST series have correlated upward trends during the period. In addition, the layout is further compressed to save space.

Linking spatial context with the Storyline (G4). We implement a geographic map, where each ST series is plotted as a dot according to its geographic position (Fig. 5C). The map hovers above the Storyline. Users can interactively drag and drop it. Afterward, we need to link the dots on the map to the curves in the Storyline. Due to a large number of involved dots, it is inappropriate to assign each dot a unique color like previous methods [10, 62]. Moreover, our study focuses on evolution patterns (i.e., sessions) rather than individual ST series (i.e., entities). Thus, we design an interactive session-based coloring (G1).

Initially, all curves and dots have the same default color (blue in our study). If users identify an interesting session, they may want to obtain

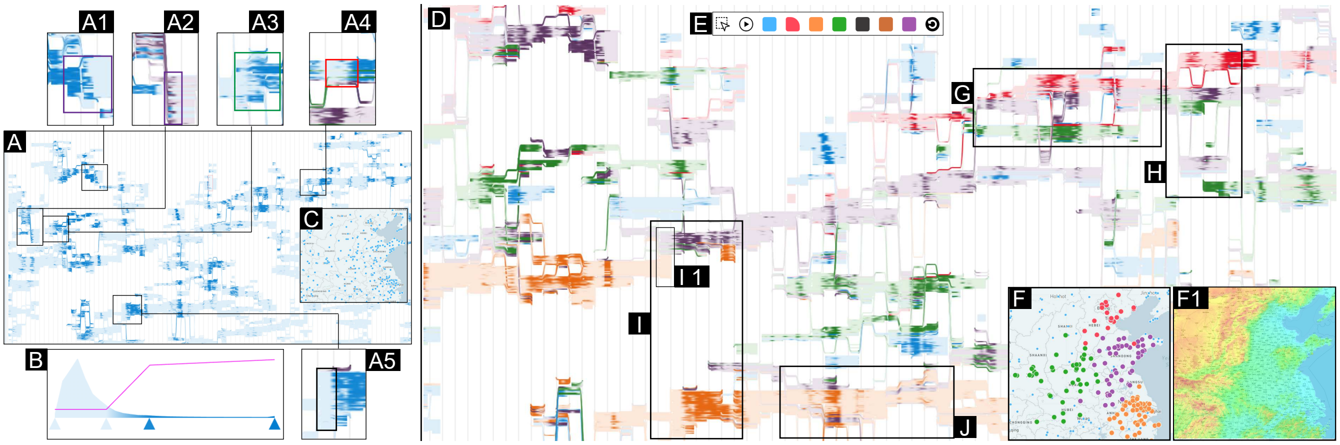

Fig. 5: Tracking overall evolution patterns in the first case. (A) Storyline with (B) the shade mapping. (C) A draggable map. (D) Colored Storyline of (A). (F) The spatial distribution of colored ST series. (G, H, I, J) Interesting evolution patterns observed in the first case.

its geographic distribution and track how the entities evolve over time (G2). To do that, they can pick up a color (Fig. 5E) and click a session to color the entities that are comprised by the clicked session. The dots of these colored entities will also be colored on the geographic map. We provide multiple optional colors, and users can color multiple sessions differently. In this way, users are enabled to establish the geographic distribution of sessions and compare the evolution of different groups of ST series. Fig. 5D shows the colored Storyline from Fig. 5A by mainly coloring the sessions highlighted in Fig. 5A1, A2, A3, A4, and A5 with purple, purple, green, orange, and red, one by one. Fig. 5F shows the geographic distribution of the colored entities.

The coloring process is difficult to automate, nor should it be automated. There are innumerable coloring results with different visual patterns. We leave the process of visual pattern discovery to users because users know what their concerns are better than the machine.

6.3.2 Drilling Down Into Evolution Patterns

After locating a time period and sessions of interest, users drill down to perform detailed analyses. We design an EvoLens (illustrated in Fig. 6C) to support detailed analyses. Users first brush a rectangle on the Storyline to select a time period and sessions. Then, the EvoLens pops up on the right side of the map. The EvoLens together with the map present spatial and temporal details of evolution patterns.

Showing detailed temporal trends (G3). The EvoLens (e.g., Fig. 6C) can be viewed as a len that enlarges the brushed sessions (e.g., Fig. 6A). Each session (e.g., Fig. 6A①) is replaced with a line chart (e.g., Fig. 6Cⓐ). If the entities belong to the same session, the corresponding time series are displayed as lines in the same line chart. Each line has the same color (without shades) as the curve in Storyline. The y-axis of all line charts represents the value range of all records. In this way, the recorded values in ST series are encoded by the height rather than the gradient shade and thus can be perceived more accurately. The line chart is the most common temporal visualization in people's daily life and most people are very familiar with it.

Evolution patterns consider the correlated trends between time series. However, correlated ST series are sometimes not necessarily similar by values (e.g., Fig. 6E) but should have similar trends. To reveal their correlative trends, we design a trend motif (Fig. 6F). We denote the subpart of an ST series in a line chart as a sub-ST series. For a line chart, we respectively normalize each sub-ST series according to its max and min values. For each time stamp in the line chart, we have a set of values, each of which is from a normalized sub-ST series. Afterward, we obtain the lower quartile $q_n(0.25)$, median, and upper quartile $q_n(0.75)$ for these values. Finally, the lower and upper quartiles in all time stamps are visualized with an area. The medians in all time stamps form a median line. The line is superimposed with the area, constituting a trend motif. Via the trend motif, the correlative trends of multiple ST series can be revealed in circumstances where the ST series are dissimilar in values from each other. For example, the trend motif of Fig. 6F clearly shows an upward trend of the ST series in Fig. 6E.

Preserving the narrative between patterns. We maintain the analysis context of entities' transition and session alignment to preserve the narrative of evolution patterns (G2). These line charts are positioned according to sessions' positions in the Storyline as follows. First, two charts are horizontally aligned (e.g., Fig. 6Cⓐ and Cⓒ) if their corresponding sessions are aligned (e.g., Fig. 6A① and A③). Second, charts are vertically aligned (Fig. 6Cⓐ and Cⓑ) if their sessions belong to the same slice (Fig. 6A① and A②). Third, the charts in the same slice are placed in the same order as their sessions in the Storyline. Finally, if there are entities passing through two sessions in the Storyline, the charts are connected with a Bezier curve from left to right. For example, Fig. 6Cⓐ is connected to Fig. 6Cⓒ and Cⓓ. The width encodes the number of entities that pass through them.

At the top of each column are the start and end times of the time slice. There is also a button that allows users to choose whether to show the motifs on the line charts.

Investigating evolution patterns in space and time. EvoLens is coordinated with the map (G4). When users hover over a time slice (Day 3 in Fig. 6C), the map displays the geographic distribution of each session in this time slice (Fig. 6B). Specifically, we generate a convex hull on the map for each session to cover the dots the session comprises. The number of sessions in this view is less than in the Storyline so that we can overlay visual elements on the map. When users hover over a line chart, the borders of both the line chart and the hull become thicker. Via the above interactions, users can interactively investigate how ST time series evolve correlatively from a spatiotemporal perspective.

7 Implementation

GeoChron is a web-based application with a backend and a frontend. The backend is implemented with Python 3.8. It serves the generation procedures of sessions and Storyline layout. In particular, the backend caches pairwise correlation coefficients among all ST series in every time window. Relation networks thereby can be constructed quickly after receiving the parameters. The Louvain algorithm is supported by the package *sknetwork*. All parameters are default, as they worked well in our multiple trials (See Sec. 8). We also employ a parallel framework *Multiprocessing* to detect communities for multiple time slices simultaneously. The frontend is implemented with TypeScript. It provides interactive visualizations for users. The Storyline is rendered on HTML5 canvas rather than HTML5 svg for efficiency.

8 Evaluation

GeoChron is evaluated with real-world case studies, an informal user study, an ablation study, parameter analysis, and running time analysis.

8.1 Case Studies

Two experts together perform case studies on two real-world datasets by using GeoChron in person. The insights gained are confirmed by another expert. Below, "we" refers to the two experts.

Two real-world case studies evaluate GeoChron as follows. First, the effectiveness and usability of GeoChron are illustrated. GeoChron enables users to view and analyze large-scale ST series intuitively, which is important in many scenarios, such as data review and real-time monitoring. Initial hypotheses regarding the evolution of spatiotemporal phenomena can also be established. Second, the pattern mining framework is justified as the evolution patterns presented in the cases have correlated trends of spatially close ST series. Third, the Storyline has a readable layout, which shows the heuristic positioning is acceptable.

8.1.1 Air Quality in China

In the first case study, we analyzed how the air quality in China evolved. A full demonstration of this case is available in the supplemental video.

Dataset. The dataset comprises 448 ST series covering nearly all of China. Each ST series records the air quality index (AQI) of a region from January 1 to July 3, 2018, at an hourly granularity. There are 448 (ST series) × 4,416 (timestamps) records, ranging from 0 to 500.

Parameters. Prior to the analysis, we set the size of the time slice to one day to obtain daily patterns and obtained 184 time slices in total. The values of this dataset had a skewed distribution (Fig. 5B). We adjusted the shade mapping and exploited the shade channel to highlight the serious pollution periods. The final mapping is shown in Fig. 5B. There were five parameters to be tuned interactively to obtain an ideal layout with many aligned sessions and little visual clutter. First, th_d could be determined based on how far the wind can blow in a time slice. Second, a too-large th_r would cause few ST series to be considered correlated and few ST series in the patterns, and vice versa. Third, we focused on the patterns with larger sizes, so we tended to increase th_s more than th_c (See Sec. 8.3 for how th_s and th_c work). Finally, we tuned th_w to reduce clutters after the layout had been determined. GeoChron can compute layouts and render results within seconds (See Sec. 8.4). After several trials, we got satisfactory parameter settings: $th_d = 300\text{km}$, $th_r = 0.7$, $th_c = 3$, $th_s = 7$, and $th_w = 140\text{px}$.

Tracking Overall Evolution Patterns.

Coloring. In the Storyline (Fig. 5A), we first noticed several sessions with large sizes and dark strokes, denoted as Fig. 5A1-5, respectively. We were interested in their spatial distributions and temporal evolution, and colored them one by one. A1 and A2 are colored purple. A3, A4, and A4 are colored green, red, and black, respectively. Generally, the sessions with more ST series in common tend to receive the same color. The final result was in Fig. 5D, where most of ST series were colored.

Spatial distribution. The map shows that these ST series were mainly in the North China Plain and the Yangtze Plain (Fig. 5F and F1). On flat terrain, air pollutants are easily transported (e.g., by winds) or diffuse. ST series with different colors constituted different geographic clusters. For example, the red and orange ST series were located in the BTH (Beijing-Tianjin-Hebei) region and Yangtze River Delta, respectively.

Temporal dynamics. The Storyline revealed how ST series correlatively evolved over a long time span. Overall, correlations tended to exist between geographically adjacent ST series. For example, the red ST series (BTH region) evolved with the purple ones (Shandong Province) during the period of Fig. 5G. Another example was found in the orange ST series. They were strongly correlated with each other over the whole time span. They sometimes were correlated with the northern purple ST series (Fig. 5I and J).

Drilling Down into Evolution Patterns.

Air Quality in North China. In Fig. 5H, we observed that a group of ST series gradually splits into four over time. The red, green, and purple ST series all exhibited an increasing trend over time as their strokes became darker. In particular, the red ST series covered Beijing, the capital of China. Fig. 1A enlarges Fig. 5H. To reveal clear patterns, we further colored Fig. 1A1 with purple and color Fig. 1A2 with black and then obtained Fig. 1B and the spatial distributions in Fig. 1C. We brushed Fig. 1B and obtained the details shown in Fig. 1D, E, and F.

First, the purple and green ST series were distributed in the center and geographically close (Fig. 1D, D2, and D3). Their readings had correlatively increased and reached a high level of air pollution within the first three days (Fig. 1E and F). Afterward (Day 4), the purple and green ST series became uncorrelated as the purple ST ones continuously increased while the green ones fluctuated. They separated into left and right regions (Fig. 1D4). Second, the black ST series were less correlated with others and the readings did not rise very high (Fig. 1E1). It might be because of the sea breeze blowing from three sides blew away air pollution quickly. Finally, the red ST series experienced a decrease on the third day, followed by an increase during the final three days, whereas the other ST series remained on a downward trend (Fig. 1E and F). A hypothesis suggests that during the last three days, an additional pollution source may have polluted the BTH region. Testing the hypotheses requires additional data and in-depth analysis [10, 13].

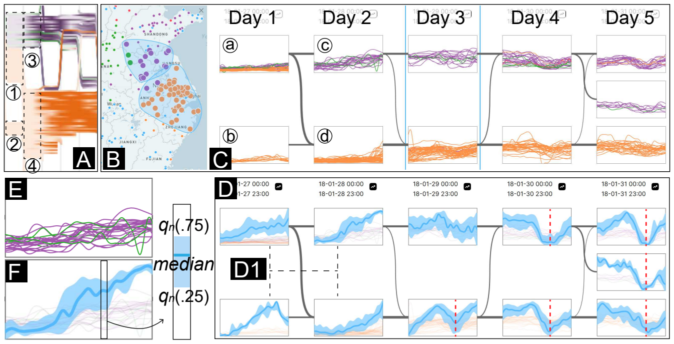

Fig. 6: Air quality in Yangtze River Delta. (A) Interesting evolution patterns in Storyline. (B) The map coordinated with the EvoLens. (C) Line charts and (D) trend motifs of the patterns of (A). (E) A line chart of an evolution pattern and (F) its trend motif.

Air Quality in Yangtze River Delta. During the period of Fig. 5I, serious pollution occurred in the Yangtze River Delta and its northern region. We wondered whether there was a propagation process between them. We wanted the orange and purple ST series to be placed closer. Thus, we enforced the alignment on the session of Fig. 5I1 and obtained the layout of Fig. 6A. The sessions with more orange and purple entities were aligned with a higher priority. Fig. 6C shows the details of Fig. 6A.

At first, half of the orange ST series were correlated with the purple ones (Day 1 in Fig. 6C). Afterward, the orange ST series separated into another session because they had a subtle drop during the period of Fig. 6D1. As a result, the orange ST series reached a high level of air pollution one day later than the purple one. In the last three days (Days 3, 4, and 5 in Fig. 6C), the orange and purple ST series had slightly different trends and both exhibited subtle declines around 4 pm each day, indicated by the red dashed line in Fig. 6D. The above observations implied that there might be propagation processes of air pollution from north to south in the first two/three days. Afterwards, the air in both regions was filled with air pollutants. The fluctuation/decline trends in the orange and purple ST series were different since the industrial structure and pollution purification capabilities of the two regions were different. We consulted historical weather data, and at that time Yangtze River Delta did receive strong cold air from the north, which caused blizzards and brought massive air pollutants.

8.1.2 Temperature in China

We visualized the temperature time series in China in the second case.

Dataset. The dataset comprises 393 ST series covering nearly all of China. Each series records the temperature of a region from January 1 to December 29, 2020, at a daily granularity. There are 393 (ST series) × 364 (timestamps) records, ranging from -37.8 °C to 43.2 °C.

Parameters. We set the size of the time slice to 7 days. In the analysis of temperature data, one of the most concerning issues is the cold wave [33]. Thus, the shade mapping was adjusted to focus on the range between 0 °C and 20 °C (Fig. 7A1). We finally obtained Fig. 7A. We kept th_d as 300 km in the first case. We increased th_r to 0.9 because the change in temperature is often stable and is less subject to local disturbance. Also because of this, the patterns with small sizes are fewer than in the first case study. We decreased th_s to 4. Other parameters were $th_c = 4$ and $th_w = 280\text{px}$ after multiple trials.

Analyzing Overall Evolution Patterns. Few patterns were detected during the summer (the middle part of Fig. 7A). Most parts of China were very hot in summer and the summer monsoon did not bring changes in temperature but rainfall. We then noticed that the

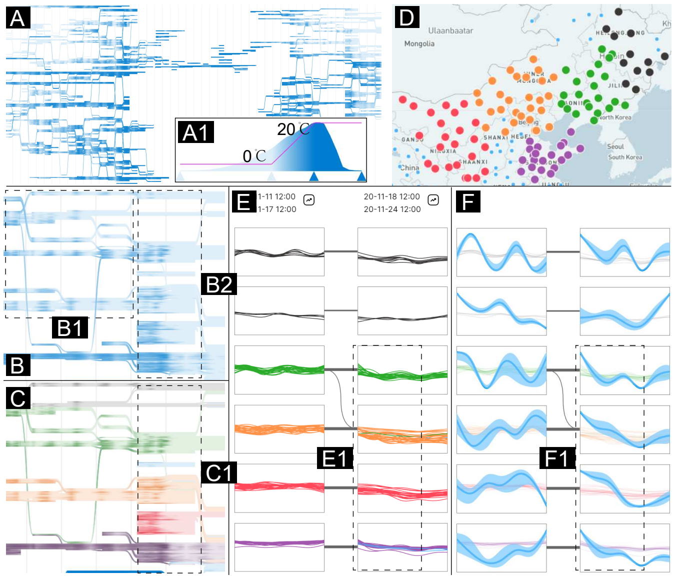

Fig. 7: Temperature in China. (A) Storyline with (A1) the shade mapping. (B) Upper right corner of (A), where (B1) periodicity and (B2) drops are observed. (C) Colored result of (B). (D) The spatial distribution of colored ST series in (C). (E) Line charts and (F) trend motifs of (C1), where obvious drops are observed in (E1, F1).

entities in the upper right corner of Fig. 7A) were lightened. The time was early winter. Fig. 7B was the enlarged part of there. There was periodic warming and cooling over the period in Fig. 7B1. Besides, a significant drop in temperature occurred during the two weeks in Fig. 7B2. Compared with the summer monsoon, the winter monsoon blowing from Siberia made the ST series correlated because it caused large-scale cooling. We colored these sessions (Fig. 7C) and found that they were mainly distributed in northern China (Fig. 7D).

Drilling Down into Patterns. Fig. 7E or F shows the details of Fig. 7C1. First, the correlation relationship of the ST series was stable. Second, in the first week, the ST series fluctuated and no obvious decrease was observed. In the second week, the ST series, except for the black ones, exhibited obvious drops of around 10 °C (Fig. 7E1). The trends can be further validated in Fig. 7F1. This cold wave was considered one of China's top ten weather and climate events in 2020 by the China Meteorological Administration.

8.2 Informal User Study

We perform an informal user study to evaluate the intuitiveness and legibility of the proposed visualizations.

Participants. We recruited six participants (P1-6) from the same school. P1 knows both Storyline and spatiotemporal data. P2 and P3 know Storyline but not spatiotemporal data. P4 and P5 know spatiotemporal data but not Storyline. P6 knows nothing about both. P1-4 are graduates majoring in computer science. P5 and P6 are undergraduates majoring in geographic information science and agricultural engineering, respectively, and they have no visualization expertise.

Procedure. We first introduced GeoChron's visual encodings and interaction. We then showed each participant the GeoChron of the first case. In particular, GeoChron was colored as Fig. 5D and then was subjected to enforced alignment according to Fig. 5I1. The participant used GeoChron via desktop in person to finish three tasks: 1) "*Locating the period when the air quality in the red, green, and purple locations are generally correlated with each other*." 2) "*Locating the period when the air quality in a large area of the Yangtze River Delta Plain deteriorates significantly*." 3) "*Following task 2, describing how the air quality deteriorates (spatiotemporal trends in the first three days) by days using EvoLens*." For task 3, they were only required to describe the objective observations with text. These open-ended tasks required participants to imitate the process of experts seeking information and analyzing ST series. Finally, each one filled out the ICE-T questionnaire [57] to rate GeoChron from **I**nsight, **C**onfidence, **E**ssence, and **T**ime perspectives (7-point Likert scale). We used think-aloud protocol to collect their feedback. The procedure lasted around 40 minutes.

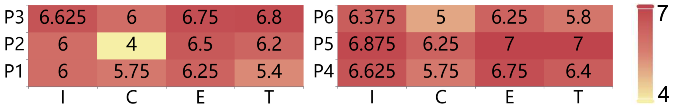

Fig. 8: ICE-T questionnaire results.

Results. All participants quickly and correctly located the periods in Fig. 5G and H for task 1 and the period in Fig. 6A for task 2. For task 3, all participants made descriptions similar to the case presented in Sec. 8.1.1. In particular, they all noticed 1) a small decline in the orange ST series on the first day (Fig. 6D1), 2) a rise in the orange ST series a day after the rise in the purple ST series, and 3) the medium/high level all ST series reached on the third day (Day 3 in Fig. 6C). It can be seen that all participants, even P6, did understand GeoChron's visualizations.

Fig. 8 summarizes the ICE-T scores. 95% C.I. are 6.41 ± 0.26 (I), 5.45 ± 0.60 (C), 6.58 ± 0.22 (E), and 6.27 ± 0.44 (T). All participants agreed that GeoChron can provide insights, facilitate exploration, and reveal the essence of data, as the scores of **I**nsight, **E**ssence, and **T**ime are about 6. The **C**onfidence scores are relatively low, which is reasonable. We assume the dataset is cleaned and thus do not design visualizations for data quality. P2 and P6 also commented, "*GeoChron is somehow misleading if ST series that are close in space are not placed together in the Storyline.*" Their confusion was dispelled during the study after we clarified the encoding of the vertical position again.

All participants agreed that GeoChron's visualizations are intuitive and easy to understand. According to participants' comments, the intuitiveness and ease of learning benefit from the proper use of existing good visualizations. The classic Storyline itself is easy to understand. The two-level visualizations built on it also have intuitive and simple encodings. In the first level, GeoChron uses the gradient shade and color to encode the temporal trends and geographic distribution, respectively. In the second level, GeoChron employs line charts to visualize the temporal trends in a familiar way.

Due to the limited paper space, we leave more study details in an appendix, including interface screenshots when doing the tasks, participants' descriptions for task 3, and the filled ICE-T questionnaires.

8.3 Ablation Study and Parameter Analysis

We refine the traditional Storyline via 1) **curve hiding**, 2) **loose alignment**, and 3) **session filtering**. In addition, we claim the 4) **sliding window** strategy could improve visual quality in Sec. 4. We figure out how they affect the layout based on the dataset in Sec. 8.1.1.

We test the **curve hiding** and **sliding windows** via ablation studies following the first case. First, we set $th_w = +Infinity$ to disable the curve hiding. Fig. 9A shows a snapshot of the result. The nearly vertical curves bounce up and down on the screen, resulting in visual distractions. Moreover, some aligned sessions are meant to be visually continuous, but this continuity is broken by these curves (e.g., Fig. 9B). Smaller the th_w, the less clutter. Second, we directly use the correlation coefficients in time slices rather than sliding windows and generate the layout in Fig. 9C with the same parameters in the first case ($th_s = 7$ and $th_c = 3$). We remove the gradient shade and focus on the layout. Compared with the time window-based results in Fig. 5D, the correlation relationship between ST series is more dynamic. In particular, the dashed ellipses in Fig. 9D and E highlight the unexpected deviation of curves. Entities part suddenly and come back together again, which may be due to the slightly offset time slices, subtle pollution events, or errors. Such deviations lead to visual clusters and prevent users from locating prominent visual patterns. It can be concluded that the sliding window does improve the readability of representations.

Afterward, we study how **loose alignment** and **session filtering** affect the layout of GeoChron via parameter analysis on th_s and th_c. We obtain different layouts generated under different combinations of th_s and th_c. Then, we use the average height of all time slices to quantify each layout's height (Fig. 10A). The layout is high when these two parameters are small (e.g., Fig. 10F). Users may have difficulty browsing the visualization. Increasing the two parameters can compact the layout. They work by different principles. On the one hand, in-

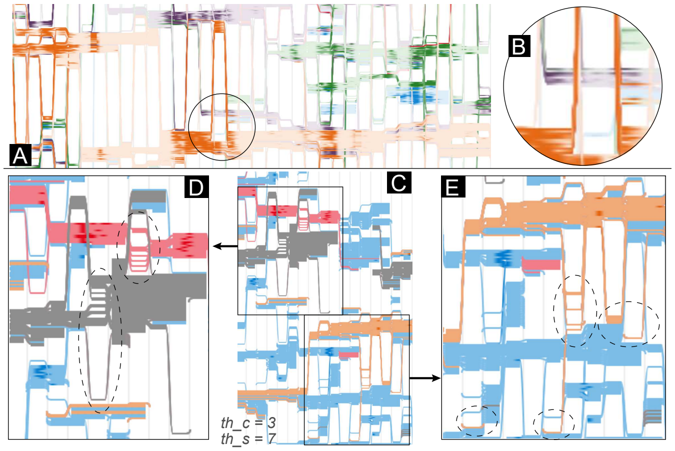

Fig. 9: Ablating (A,B) curve hiding and (C,D,E) sliding windows.

creasing th_c results in the layout with looser alignment (e.g., Fig. 10G). Users can still analyze those sessions with small sizes. On the other hand, increasing th_s results in the layout of Fig. 5A or D, where fewer sessions are displayed (Fig. 10B).

8.4 Running Time Analysis

Recall that users may need to refine the layout, e.g., for less clutter, fewer small patterns, or more aligned patterns. We report the running times of four backend modules (evolution pattern detection, ordering, aligning, and positioning) to clarify the fluency of user interaction when refining the layout. The experiments were performed on a desktop running Ubuntu 20.04 with Intel Core i7 3.70GHz CPU, and 16 GB RAM. The detection and aligning modules are accelerated via *Multiprocessing* with 12 pools. Recall that determining correlation does not require high efficiency since the results are cached in advance in the backend.

Detection and Ordering. Users interact with these two modules when adjusting th_d and th_r. We report the running times under different th_d and th_r (Fig. 10H). Fig. 10H (left) and (right) are the run times of pattern detection and ordering, respectively. With the increasing of th_d and decreasing of th_r, these two modules consume more time because more pairs of ST series are considered correlated. The total running time in the loosest case in Fig. 10H is 12 seconds.

Aligning and Positioning. Users interact with these two modules when adjusting th_s and th_c. The running time of the aligning module depends on the number of sessions the module needs to process (Fig. 10C). With a larger th_s, fewer sessions are left and this module runs faster. The running time of the positioning module mainly depends on the number of aligned session pairs. Intuitively, aligned pairs are the constraints of the positioning process. The sweeping process is more likely to return backward if there were more constraints. If th_s and th_c are smaller, there are more aligned pairs (Fig. 10D), and the positioning module consumes more time (Fig. 10E). The aligning and positioning modules in our experiments consume no more than 10 seconds.

Generally, the backend can respond to users within seconds, so users can easily tune the layout and obtain an ideal one for the dataset with 448 ST series and 184 time slices.

9 Discussion and Conclusion

Lessons. We have two lessons. First, *interactive layout refining is useful*. There are many possibilities for the visual layout of large-scale data. An ideal layout for the proposed visualization is hard to determine automatically. Users may have personalized analysis interests that are hard to quantify for automated optimization. In GeoChron, users are able to interactively refine the layout according to their needs, making a trade-off between information completeness and readability. In this way, the smoothness of layout adjustment needs to be guaranteed.

Second, *multilevel visualization is suitable for large-scale data*. GeoChron adopts the two-level visualization in a compactly laid-out Storyline such that the Storyline becomes applicable to large-scale ST series. We suggest that the multilevel strategy [12, 18, 21, 34] could be the first choice to improve the scalability of big data analysis.

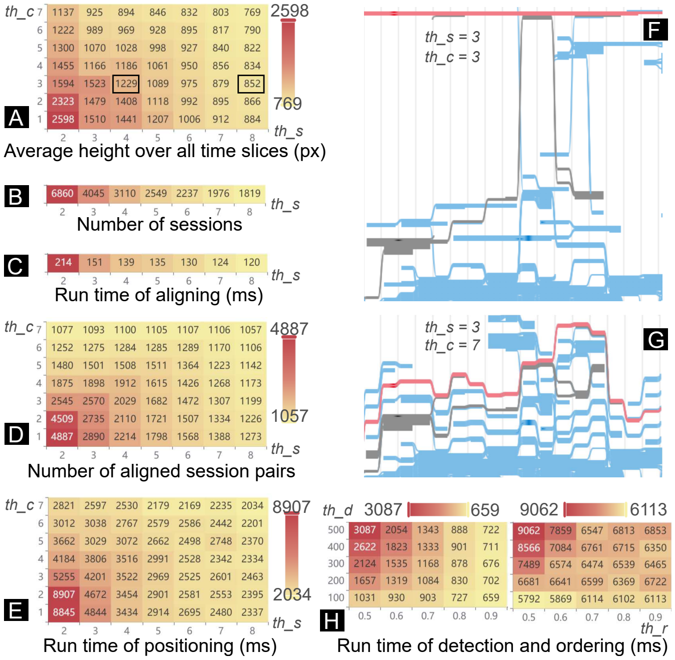

Fig. 10: Quantitative experiment results. (A) Average height over all time slices with different th_c and th_s. (B) Number of sessions left and (C) run times of aligning with different th_s. (D) Number of aligned session pairs and (E) run times of positioning with different combinations of th_c and th_s. (F) Snapshot when $th_c = 3$ and $th_s = 3$. (G) Snapshot when $th_c = 7$ and $th_s = 3$. (H) Run times of detection (left) and ordering (right) with different combinations of th_d and th_r.

Generalizability. GeoChron can be applied to various kinds of data in addition to *geo-spatial time series*. First, visualizing *cross-domain geo-spatial time series* can reveal in-depth knowledge. For example, air quality is oftentimes correlated to temperature or traffic conditions [61]. To accommodate potential heterogeneity, the correlation modeling method should be replaced with one suitable for heterogeneous data.

Second, if using general spatial relationships, GeoChron is potentially applicable for *general spatial time series* [26,29,58]. For example, in a coal-fired power plant, sensors have a logical relationship to each other depending on where they are located in the workflow [29].

Limitations and Future Work. Our study has two limitations. First, GeoChron may take up a lot of horizontal space if there are too many time slices, requiring the user to scroll a lot horizontally. Although this issue can be potentially solved by increasing the size of slices, it may reduce the granularity and result in information loss. Besides, GeoChron may take up a lot of vertical space when the number of ST series increases. Although we allow users to refine and compress the layout, small but important patterns may be hidden in this way. In the future, we plan to study level-of-detail rendering to support larger scale data [8], for example, by abstracting the representation further [73].

Second, one ST series can be in multiple patterns. To reflect the situation, the curve should be split into multiple sessions. Yet, the splitting curve may exacerbate visual clutter. Addressing such an issue needs further studies of the layout algorithm and visualization.

GeoChron can also be further improved as follows. The algorithm can be improved to adaptively identify the timestamps when the correlation relations change, and obtain reasonable time slices. Query and more pattern mining methods can be incorporated. Then, GeoChron highlights query or mining results to facilitate ST series exploration.

Conclusion. This paper presents GeoChron, an effective visualization for large-scale ST series. We formulate the problem of visualizing large-scale ST series as an evolution pattern visualization problem that Storyline techniques can solve well. To apply Storyline techniques, GeoChron includes a mining framework for extracting evolution patterns from ST series and a two-level visualization mechanism for improved visual scalability. As a result, GeoChron enables pattern-aware and narrative-preserving visualization of large-scale ST series.

Acknowledgments

The work was supported by National Key R&D Program of China (2022YFE0137800) and NSFC (U22A2032), and the Collaborative Innovation Center of Artificial Intelligence by MOE and Zhejiang Provincial Government (ZJU).

References

[1] S. R. Aghabozorgi, A. S. Shirkhorshidi, and Y. W. Teh. Time-series clustering - A decade review. *Information System*, 53:16–38, 2015. doi: 10.1016/j.is.2015.04.007 2

[2] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*, vol. 4. Springer, 2011. 2

[3] G. L. Andrienko, N. V. Andrienko, J. Dykes, S. I. Fabrikant, and M. Wachowicz. Geovisualization of dynamics, movement and change: Key issues and developing approaches in visualization research. *Information Visualization*, 7(3-4):173–180, 2008. doi: 10.1057/ivs.2008.23 2

[4] B. Bach, C. Shi, N. Heulot, T. M. Madhyastha, T. J. Grabowski, and P. Dragicevic. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):559–568, 2016. doi: 10.1109/TVCG.2015.2467851 2

[5] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, oct 2008. doi: 10.1088/1742-5468/2008/10/P10008 4

[6] G. Chatzigeorgakidis, K. Patroumpas, D. Skoutas, S. Athanasiou, and S. Skiadopoulos. Visual exploration of geolocated time series with hybrid indexing. *Big Data Research*, 15:12–28, 2019. doi: 10.1016/j.bdr.2019.02.001 2

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill Book Company, 2001. 4, 5

[8] Z. Deng, S. Chen, X. Xie, G. Sun, M. Xu, D. Weng, and Y. Wu. Multi-level visual analysis of aggregate geo-networks. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–16, 2022. Early Access. doi: 10.1109/TVCG.2022.3229953 9

[9] Z. Deng, D. Weng, J. Chen, R. Liu, Z. Wang, J. Bao, Y. Zheng, and Y. Wu. AirVis: Visual analytics of air pollution propagation. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):800–810, 2020. doi: 10.1109/TVCG.2019.2934670 1, 2, 3

[10] Z. Deng, D. Weng, Y. Liang, J. Bao, Y. Zheng, T. Schreck, M. Xu, and Y. Wu. Visual cascade analytics of large-scale spatiotemporal data. *IEEE Transactions on Visualization and Computer Graphics*, 28(6):2486–2499, 2022. doi: 10.1109/TVCG.2021.3071387 1, 2, 3, 5, 7

[11] Z. Deng, D. Weng, S. Liu, Y. Tian, M. Xu, and Y. Wu. A survey of urban visual analytics: Advances and future directions. *Computational Visual Media*, 9(1):3–39, 2023. doi: 10.1007/s41095-022-0275-7 2

[12] Z. Deng, D. Weng, and Y. Wu. You are experienced: Interactive tour planning with crowdsourcing tour data from web. *Journal of Visualization*, 26(2):385–401, 2023. doi: 10.1007/s12650-022-00884-1 9

[13] 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 1, 2, 3, 7

[14] Y. Dong, I. Oppermann, J. Liang, X. Yuan, and V. N. Quang. User-centered visual explorer of in-process comparison in spatiotemporal space. *Journal of Visualization*, 26(2):403–421, 2023. doi: 10.1007/s12650-022-00882-3 2

[15] M. Evers, K. Huesmann, and L. Linsen. Uncertainty-aware visualization of regional time series correlation in spatio-temporal ensembles. *Computer Graphics Forum*, 40(3):519–530, 2021. doi: 10.1111/cgf.14326 1

[16] H. Guo, M. Liu, B. Yang, Y. Sun, H. Qu, and L. Shi. Rankfirst: Visual analysis for factor investment by ranking stock timeseries. *IEEE Transactions on Visualization and Computer Graphics*, 2022. doi: 10.1109/TVCG.2022.3209414 2

[17] G. Hulstein, V. P. Araya, and A. Bezerianos. Geo-Storylines: Integrating maps into storyline visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):994–1004, 2023. doi: 10.1109/TVCG.2022.3209480 2, 3

[18] P. Karnick, D. Cline, S. Jeschke, A. Razdan, and P. Wonka. Route visualization using detail lenses. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):235–247, 2010. doi: 10.1109/TVCG.2009.65 9

[19] P. Köthur, M. Sips, J. Kuhlmann, and D. Dransch. Visualization of geospatial time series from environmental modeling output. In *Proceedings of Eurographics Conference on Visualization (Short Papers)*. Eurographics Association, 2012. doi: 10.2312/PE/EuroVisShort/EuroVisShort2012/115-119 1, 2

[20] P. Köthur, M. Sips, A. Unger, J. Kuhlmann, and D. Dransch. Interactive visual summaries for detection and assessment of spatiotemporal patterns in geospatial time series. *Information Visualization*, 13(3):283–298, 2014. doi: 10.1177/1473871613481692 2

[21] F. Lekschas, M. Behrisch, B. Bach, P. Kerpedjiev, N. Gehlenborg, and H. Pfister. Pattern-driven navigation in 2d multiscale visualizations with scalable insets. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):611–621, 2020. doi: 10.1109/TVCG.2019.2934555 9

[22] C. Li, G. Baciu, Y. Wang, J. Chen, and C. Wang. DDLVis: Real-time visual query of spatiotemporal data distribution via density dictionary learning. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1062–1072, 2022. doi: 10.1109/TVCG.2021.3114762 2

[23] J. Li, S. Chen, K. Zhang, G. L. Andrienko, and N. V. Andrienko. COPE: Interactive exploration of co-occurrence patterns in spatial time series. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2554–2567, 2019. doi: 10.1109/TVCG.2018.2851227 1, 2, 3

[24] J. Li, K. Zhang, and Z. Meng. Vismate: Interactive visual analysis of station-based observation data on climate changes. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pp. 133–142, 2014. doi: 10.1109/VAST.2014.7042489 1, 2

[25] X. Li, Y. Cheng, G. Cong, and L. Chen. Discovering pollution sources and propagation patterns in urban area. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1863–1872, 2017. doi: 10.1145/3097983.3098090 2

[26] Y. Li, X. Li, S. Shen, L. Zeng, R. Liu, Q. Zheng, J. Feng, and S. Chen. DTBVis: An interactive visual comparison system for digital twin brain and human brain. *Visual Informatics*, 7(2):41–53, 2023. doi: 10.1016/j.visinf.2023.02.002 9

[27] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng. GeoMAN: Multi-level attention networks for geo-sensory time series prediction. In *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 3428–3434, 2018. doi: 10.24963/ijcai.2018/476 1

[28] D. Liu, P. Xu, and L. Ren. TPFlow: Progressive partition and multidimensional pattern extraction for large-scale spatio-temporal data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):1–11, 2019. doi: 10.1109/TVCG.2018.2865018 2

[29] S. Liu, D. Weng, Y. Tian, Z. Deng, H. Xu, X. Zhu, H. Yin, X. Zhan, and Y. Wu. ECoalVis: Visual analysis of control strategies in coal-fired power plants. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1091–1101, 2023. doi: 10.1109/TVCG.2022.3209430 2, 9

[30] S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. StoryFlow: Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2436–2445, 2013. doi: 10.1109/TVCG.2013.196 1, 2, 3, 4, 5

[31] Y. Liu, J. Wu, and D. Yu. Characterizing spatiotemporal patterns of air pollution in china: A multiscale landscape approach. *Ecological Indicators*, 76:344–356, 2017. doi: 10.1016/j.ecolind.2017.01.027 2

[32] X. Lu, Y. Xu, G. Li, Y. Chen, and G. Shan. MVST-SciVis: Narrative visualization and analysis of compound events in scientific data. *Journal of Visualization*, 26(3):687–703, 2023. doi: 10.1007/s12650-022-00893-0 2

[33] S. Ma and C. Zhu. Extreme cold wave over east asia in january 2016: A possible response to the larger internal atmospheric variability induced by arctic warming. *Journal of Climate*, 32(4):1203 – 1216, 2019. doi: 10.1175/JCLI-D-18-0234.1 7

[34] J. Magallanes, T. Stone, P. D. Morris, S. Mason, S. Wood, and M. Villa-Uriol. Sequen-c: A multilevel overview of temporal event sequences. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):901–911, 2022. doi: 10.1109/TVCG.2021.3114868 9

[35] A. Malik, R. Maciejewski, N. Elmqvist, Y. Jang, D. S. Ebert, and W. Huang. A correlative analysis process in a visual analytics environment. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pp. 33–42, 2012. doi: 10.1109/VAST.2012.6400491 2

[36] T. W. Meshesha, J. Wang, and N. D. Melaku. Modelling spatiotemporal patterns of water quality and its impacts on aquatic ecosystem in the cold climate region of alberta, canada. *Journal of Hydrology*, 587:124952, 2020. doi: 10.1016/j.jhydrol.2020.124952 2

[37] M. Monmonier. Strategies for the visualization of geographic time-series data. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 27:30–45, 10 1990. doi: 10.3138/U558-H737-6577-8U31 1

[38] F. K. Muthoni, V. O. Odongo, J. Ochieng, E. M. Mugalavai, S. K. Mourice, I. Hoesche-Zeledon, M. Mwila, and M. Bekunda. Long-term spatial-

temporal trends and variability of rainfall over eastern and southern africa. *Theoretical and Applied Climatology*, 137(3):1869–1882, 2019. doi: 10.1007/s00704-018-2712-1 2

[39] M. Ogawa and K. Ma. Software evolution storylines. In *Proceedings of ACM Symposium on Software Visualization*, pp. 35–42, 2010. doi: 10.1145/1879211.1879219 2

[40] P. Patel, E. J. Keogh, J. Lin, and S. Lonardi. Mining motifs in massive time series databases. In *Proceedings of IEEE International Conference on Data Mining*, pp. 370–377, 2002. doi: 10.1109/ICDM.2002.1183925 2

[41] Y. Qin, J. Li, K. Gong, Z. Wu, M. Chen, M. Qin, L. Huang, and J. Hu. Double high pollution events in the yangtze river delta from 2015 to 2019: Characteristics, trends, and meteorological situations. *Science of The Total Environment*, 792:148349, 2021. doi: 10.1016/j.scitotenv.2021.148349 2

[42] H. Qu, W. Chan, A. Xu, K. Chung, A. K. Lau, and P. Guo. Visual analysis of the air pollution problem in hong kong. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1408–1415, 2007. doi: 10.1109/TVCG.2007.70523 1

[43] N. Rodrigues, R. Netzel, K. R. Ullah, M. Burch, A. Schultz, B. Burger, and D. Weiskopf. Visualization of time series data with spatial context: Communicating the energy production of power plants. In *Proceedings of International Symposium on Visual Information Communication and Interaction*, pp. 37–44. ACM, 2017. doi: 10.1145/3105971.3105982 1, 2

[44] Y. Shi, C. Bryan, S. Bhamidipati, Y. Zhao, Y. Zhang, and K. Ma. MeetingVis: Visual narratives to assist in recalling meeting context and content. *IEEE Transactions on Visualization and Computer Graphics*, 24(6):1918–1929, 2018. doi: 10.1109/TVCG.2018.2816203 2

[45] G. Shirato, N. V. Andrienko, and G. L. Andrienko. Identifying, exploring, and interpreting time series shapes in multivariate time intervals. *Visual Informatics*, 7(1):77–91, 2023. doi: 10.1016/j.visinf.2023.01.001 2

[46] J. Spinoni, M. Lakatos, T. Szentimrey, Z. Bihari, S. Szalai, J. Vogt, and T. Antofie. Heat and cold waves trends in the carpathian region from 1961 to 2010. *International Journal of Climatology*, 35(14):4197–4209, 2015. doi: 10.1002/joc.4279 1, 2

[47] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981. doi: 10.1109/TSMC.1981.4308636 4

[48] G. Sun, R. Liang, H. Qu, and Y. Wu. Embedding spatio-temporal information into maps by route-zooming. *IEEE Transactions on Visualization and Computer Graphics*, 23(5):1506–1519, 2017. doi: 10.1109/TVCG.2016.2535234 2

[49] Y. Tanahashi, C. Hsueh, and K. Ma. An efficient framework for generating storyline visualizations from streaming data. *IEEE Transactions on Visualization and Computer Graphics*, 21(6):730–742, 2015. 4

[50] Y. Tanahashi and K. Ma. Design considerations for optimizing storyline visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2679–2688, 2012. doi: 10.1109/TVCG.2012.212 1, 2

[51] T. Tang, R. Li, X. Wu, S. Liu, J. Knittel, S. Koch, L. Yu, P. Ren, T. Ertl, and Y. Wu. PlotThread: Creating expressive storyline visualizations using reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):294–303, 2021. doi: 10.1109/TVCG.2020.3030467 2, 4

[52] T. Tang, S. Rubab, J. Lai, W. Cui, L. Yu, and Y. Wu. iStoryline: Effective convergence to hand-drawn storylines. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):769–778, 2019. doi: 10.1109/TVCG.2018.2864899 2, 4

[53] T. Tang, Y. Wu, Y. Wu, L. Yu, and Y. Li. VideoModerator: A risk-aware framework for multimodal video moderation in e-commerce. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):846–856, 2022. doi: 10.1109/TVCG.2021.3114781 2, 4

[54] S. Thakur and A. J. Hanson. A 3D visualization of multiple time series on maps. In *Proceedings of IEEE International Conference on Information Visualisation*, pp. 336–343, 2010. doi: 10.1109/IV.2010.54 1, 2

[55] The World Air Quality Index project. New York, USA Air Pollution: Real-time Air Quality index. `https://aqicn.org/city/usa/newyork/`. Last accessed on 28.03.2023. 3

[56] V. A. Traag. Faster unfolding of communities: Speeding up the louvain algorithm. *Physical Review E*, 92(3):032801, 2015. doi: 10.1103/PhysRevE.92.032801 4

[57] E. Wall, M. Agnihotri, L. E. Matzen, K. Divis, M. Haass, A. Endert, and J. T. Stasko. A heuristic approach to value-driven evaluation of visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):491–500, 2019. doi: 10.1109/TVCG.2018.2865146 8

[58] Y. Wang, Z. Zhu, L. Wang, G. Sun, and R. Liang. Visualization and visual analysis of multimedia data in manufacturing: A survey. *Visual Informatics*, 6(4):12–21, 2022. doi: 10.1016/j.visinf.2022.09.001 9

[59] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. van de Wetering. Visual traffic jam analysis based on trajectory data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2159–2168, 2013. doi: 10.1109/TVCG.2013.228 2

[60] X. Wu, C. Cheng, R. Zurita-Milla, and C. Song. An overview of clustering methods for geo-referenced time series: from one-way clustering to co- and tri-clustering. *International Journal of Geographical Information Science*, 34(9):1822–1848, 2020. doi: 10.1080/13658816.2020.1726922 2

[61] Y. Wu, D. Weng, Z. Deng, J. Bao, M. Xu, Z. Wang, Y. Zheng, Z. Ding, and W. Chen. Towards better detection and analysis of massive spatiotemporal co-occurrence patterns. *IEEE Transactions on Intelligent Transportation Systems*, 22(6):3387–3402, 2021. doi: 10.1109/TITS.2020.2983226 2, 9

[62] S. Yagi, T. Itoh, and M. Takatsuka. A layout technique for storyline-based visualization of consecutive numerical time-varying data. In *Proceedings of International Symposium on Visual Information Communication and Interaction*, pp. 156–157. ACM, 2015. doi: 10.1145/2801040.2801067 2, 3, 5

[63] C. Yang, Z. Zhang, Z. Fan, R. Jiang, Q. Chen, X. Song, and R. Shibasaki. EpiMob: Interactive visual analytics of citywide human mobility restrictions for epidemic control. *IEEE Transactions on Visualization and Computer Graphics*, 2022. doi: 10.1109/TVCG.2022.3165385 2

[64] W.-F. Ye, Z.-Y. Ma, and X.-Z. Ha. Spatial-temporal patterns of pm2.5 concentrations for 338 chinese cities. *Science of The Total Environment*, 631-632:524–533, 2018. doi: 10.1016/j.scitotenv.2018.03.057 2

[65] X. Yi, Z. Duan, R. Li, J. Zhang, T. Li, and Y. Zheng. Predicting fine-grained air quality based on deep neural networks. *IEEE Transactions on Big Data*, 8(5):1326–1339, 2022. doi: 10.1109/TBDATA.2020.3047078 1

[66] L. Ying, X. Shu, D. Deng, Y. Yang, T. Tang, L. Yu, and Y. Wu. MetaGlyph: Automatic generation of metaphoric glyph-based visualization. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):331–341, 2023. doi: 10.1109/TVCG.2022.3209447 2

[67] L. Ying, T. Tang, Y. Luo, L. Shen, X. Xie, L. Yu, and Y. Wu. GlyphCreator: Towards example-based automatic generation of circular glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):400–410, 2022. doi: 10.1109/TVCG.2021.3114877 2

[68] Y. Yu, D. Kruyff, J. Jiao, T. Becker, and M. Behrisch. PSEUDo: Interactive pattern search in multivariate time series with locality-sensitive hashing and relevance feedback. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):33–42, 2023. doi: 10.1109/TVCG.2022.3209431 2

[69] X. Yue, J. Bai, Q. Liu, Y. Tang, A. Puri, K. Li, and H. Qu. sportfolio: Stratified visual analysis of stock portfolios. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):601–610, 2020. doi: 10.1109/TVCG.2019.2934660 2

[70] G. G. Zanabria, J. Silveira, J. Poco, A. Paiva, M. B. Nery, C. T. Silva, S. Adorno, and L. G. Nonato. CrimAnalyzer: Understanding crime patterns in são paulo. *IEEE Transactions on Visualization and Computer Graphics*, 27(4):2313–2328, 2021. doi: 10.1109/TVCG.2019.2947515 1

[71] W. Zeng, C. Fu, S. M. Arisona, S. Schubiger, R. Burkhard, and K. Ma. Visualizing the relationship between human mobility and points of interest. *IEEE Transactions on Intelligent Transportation Systems*, 18(8):2271–2284, 2017. doi: 10.1109/TITS.2016.2639320 2

[72] W. Zhao, G. Wang, Z. Wang, L. Liu, X. Wei, and Y. Wu. A uncertainty visual analytics approach for bus travel time. *Visual Informatics*, 6(4):1–11, 2022. doi: 10.1016/j.visinf.2022.06.002 2

[73] Y. Zhao, L. Ge, H. Xie, G. Bai, Z. Zhang, Q. Wei, Y. Lin, Y. Liu, and F. Zhou. ASTF: Visual abstractions of time-varying patterns in radio signals. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):214–224, 2023. doi: 10.1109/TVCG.2022.3209469 2, 9

[74] Y. Zhao, Y. Wang, J. Zhang, C. 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, 5

[75] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li. Forecasting fine-grained air quality based on big data. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2267–2276, 2015. doi: 10.1145/2783258.2788573 2

[76] J. Y. Zhu, C. Zhang, H. Zhang, S. Zhi, V. O. K. Li, J. Han, and Y. Zheng. pg-Causality: Identifying spatiotemporal causal pathways for air pollutants with urban big data. *IEEE Transactions on Big Data*, 4(4):571–585, 2018. doi: 10.1109/TBDATA.2017.2723899 1, 2