This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3358919

A Comparative Study on Fixed-order Event Sequence Visualizations: Gantt, Extended Gantt, and Stringline Charts

Junxiu Tang, Fumeng Yang, Jiang Wu, Yifang Wang, Jiayi Zhou, Xiwen Cai, Lingyun Yu, and Yingcai Wu

***Abstract*—We conduct two in-lab experiments (N=93) to evaluate the effectiveness of Gantt charts, extended Gantt charts, and stringline charts for visualizing fixed-order event sequence data. We first formulate five types of event sequences and define three types of sequence elements: point events, interval events, and the temporal gaps between them. Our two experiments focus on event sequences with a pre-defined, fixed order and measure task error rates and completion time. The first experiment shows single sequences and assesses the three charts' performance in comparing event duration or gap. The second experiment shows multiple sequences and evaluates how well the charts reveal temporal patterns. The results suggest that when visualizing single fixed-order event sequences, 1) Gantt and extended Gantt charts lead to comparable error rates in the duration-comparing task; 2) Gantt charts exhibit either shorter or equal completion time than extended Gantt charts; 3) both Gantt and extended Gantt charts demonstrate shorter completion times than stringline charts; 4) however, stringline charts outperform the other two charts with fewer errors in the comparing task when event type counts are high. Additionally, when visualizing multiple point-based fixed-order event sequences, stringline charts require less time than Gantt charts for people to find temporal patterns. Based on these findings, we discuss design opportunities for visualizing fixed-order event sequences and discuss future avenues for optimizing these charts.**

***Index Terms*—Gantt chart, stringline chart, Marey's graph, event sequence, empirical study.**

I. Introduction

FIXED-ORDER event sequence is a type of sequence data that exists in various scenarios, such as industrial logs of products in assembly lines [1]–[5], traveling records of vehicles in public transportation [6]–[9], cascading effects of disaster [10], and biological process like apoptosis [11]. In these event sequences, the order of events is predefined and remains consistent in each sequence. This stands different from sequences where events occur in an unpredictable order [12]–[17]. To illustrate, consider a scenario of manufacturing a particular product within an industrial workflow. Each stage in this process can be considered as an event that needs to be carried out in a predefined order to complete the process. In this context, the production process of an individual product is a fixed-order event sequence. Despite the fixed order, the identical steps in different sequences happen at different timestamps and last for different durations. Visualizing such event sequences can provide valuable insights into understanding, troubleshooting, and optimizing production processes.

Our work thus focuses on fixed-order event sequence visualization. The literature on visualizing fixed-order event sequences is extensive, producing timeline-based visualizations that can encode event types, timestamps, and duration, such as Gantt chart [8], [18] and stringline chart (also known as Marey's graph) [1], [2], [6]. As fixed-order event sequences are a subset of general event sequences where the event orders are inconsistent or random, other timeline-based visualizations for general sequences such as extended Gantt chart (a variant of a Gantt chart in the detailed view in Monroe et al.'s studies [19]–[21]) can also be used to visualize fixed-order event sequences since it can present the event type and temporal information. These timeline-based visualizations can be applied to temporal regulation finding [8], [19], efficiency analysis [2], [6] and visual diagnostics [1], underscoring their strengths in various usage scenarios. Therefore, it is important to understand their performance and select the most appropriate visualization. Understanding them is also a fundamental step to developing more efficient visualization methods in the future.

In this work, we compare three timeline-based visualizations designed for representing fixed-order event sequence data (Fig. 1): Gantt chart (Gantt), extended Gantt chart (ExtGantt), and stringline charts (Stringline). We compare their performance across various scenarios. These visualizations are commonly used for fixed-order event sequence data or event sequences in general [1], [2], [6], [8], [18]–[22]. Other existing event sequence visualizations [22], such as Hierarchy-based, Sankey-based, and Matrix-based, lack the capability to simultaneously show essential elements in fixed-order event sequences (e.g., event type, duration, and timestamps). Thus, these alternatives are not considered within the scope of our work.

To begin, we reviewed the literature and summarized five types of fixed-order event sequences, taking into account point events, interval events, and event gaps (Sec. III) as three essential elements. Next, we surveyed related studies and conducted a pilot study to select appropriate tasks for the experiment, and finally chose duration-comparison and temporal-pattern-finding as the two experimental tasks. Our experimental variables include *visualization*, *sequence element type*, *event type count*, *the number of sequences*, and *temporal pattern*. More specifically,

- We conducted two in-lab experiments (N=93) to evaluate

Yingcai Wu is the corresponding author. Manuscript created October, 2020; This work was developed by the IEEE Publication Technology Department.

the effectiveness of Gantt charts, extended Gantt charts, and stringline charts in visualizing fixed-order event sequence data. **The first experiment** examined the effectiveness of the three charts in supporting duration-comparison tasks within a single event sequence. **The second experiment** explored the advantages and disadvantages of these visualizations for temporal patterns (i.e., timestamp and duration patterns) in multiple event sequences.

- We reported the following quantitative results for fixed-order event sequence data. 1) Overall, Stringline required more completion time than Gantt and ExtGantt in the task that compared the event duration in a single sequence. 2) No significant differences were observed between Gantt and ExtGantt in terms of both error rate and completion time. 3) Stringline required less time than Gantt in finding temporal patterns in multiple sequences.
- We summarized design suggestions for choosing appropriate charts for fixed-order event sequence data, which can assist future work in selecting suitable visualizations for the specific scenarios.

II. Related Work

We review visualizations used in previous event sequence studies [22]–[24] to identify those charts which can simultaneously present scheduled event sequences' temporal information, including event types, key timestamps, and duration. After the survey, we categorize these charts into two groups: Gantt chart-like and stringline-like, based on how fixed-order events' types is encoded.

Gantt chart displays time on a timeline axis and uses categorical colormaps or textures to represent event types (Fig. 1 (b)). Each colored rectangle (or line) represents a single event, and the rectangle width represents the event duration. A row of rectangles represents a sequence of fixed-order events, and the space between two rectangles shows the time interval between two adjacent events. Initially designed for tracking the project progress [25], [26], Gantt charts are widely used in project planning [20] and schedule comparison [27], [28]. Jo et al. [8] later improved the scalability of the Gantt chart to visualize large volumes of manufacturing data and flight schedules by reordering the sequence based on event type similarity and grouping events with similar durations. Their work showed that Gantt chart-like visualization performs well when presenting a large number of event sequences through algorithms and interactions.

On the other hand, a few studies proposed and utilized the variants of Gantt charts to visualize various data types effectively. Luz et al. [29], [30] propose mosaic charts to make the visualization more compact. Several subsequent studies compared Gantt and mosaic charts [31], [32] and found that these two charts yield similar reading accuracy, completion time, and subjective assessments. Still, mosaic charts can make good use of viewport spaces and convey the overlapping information of parallel events. Moreover, other visual analysis studies used **extended Gantt chart** to represent complex event sequences, replacing rectangles with triangles to depict point-based events and arranging different event types in different vertical positions. For instance, LifeLines [18] and its follow-up studies [19], [21], [33], [34] used extended Gantt charts for exploring details of event sequences. However, Gantt and ExtGantt charts can be less legible when there are many event types because humans can only simultaneously perceive a limited number (around 12 bins) of colors [35].

Stringline chart, also known as Marey's graph, was originally developed in the 19th century to display a static train schedule from Paris to Lyon [36], [37] and has been widely used among transportation experts to represent fixed schedules [6], [9], [38], [39]. It encodes time and distance between stations on the horizontal and vertical axes [6], [9], and sometimes in reverse [7]. Palomo et al. [6] extended it in visual analytics for transportation. They introduced a system, TR-EX, which used an extended stringline chart to reveal spatiotemporal patterns of public transportation and employed kernel density estimation for readability.

Other studies extended transportation schedule visualization to present the industrial manufacturing process [1], [2], in which the "transport stations" were replaced by "procedures". As illustrated in Fig. 1(b), procedures are placed on the y-axis of a schedule, and a line segment along one procedure slice represents an event within that procedure, with the length indicating the event duration. The line segment between two procedure slices represents the time interval between two events. A polyline consisting of line segments of each procedure represents a sequence of fixed-order events. Xu et al. [1] used an extended Stringline chart to present historical logs of assembly lines. They designed a time-aware outlier-preserving visual aggregation algorithm to diagnose abnormal durations among fixed-order events. Inspired by stringline chart, Ono et al. [37] utilized a similar visual encoding to summarize baseball plays. Tang et al. [2] integrated stringline chart units into their system to display the waiting duration between processing procedures. The existing studies suggest that stringline chart-like visualizations effectively present duration and timestamp patterns among similar or normal event sequences. Still, they may result in visual cluttering when showing large volumes of sequences.

These previous studies suggested timeline-based visualizations Gantt, ExtGantt, and Stringline can effectively visualize fixed-order event sequences data, depending on factors such as data structure, volume, and analysis tasks. Although Di Bartolomeo et al. [40] evaluate the effectiveness of different timeline shapes in presenting event sequences, their work mainly focuses on the timestamp values on timelines, different from specific charts with duration encoding in this work. To our knowledge, no comparative study has formally assessed the effectiveness of Gantt, ExtGantt, and Stringline charts in visualizing fixed-order event sequences. While prior studies offer valuable insights into various domains, it remains unclear which is the optimal choice for a given situation. Therefore, we conducted comparative experiments to evaluate the performance of each chart in addressing visual analysis tasks concerning fixed-order event sequences.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3358919

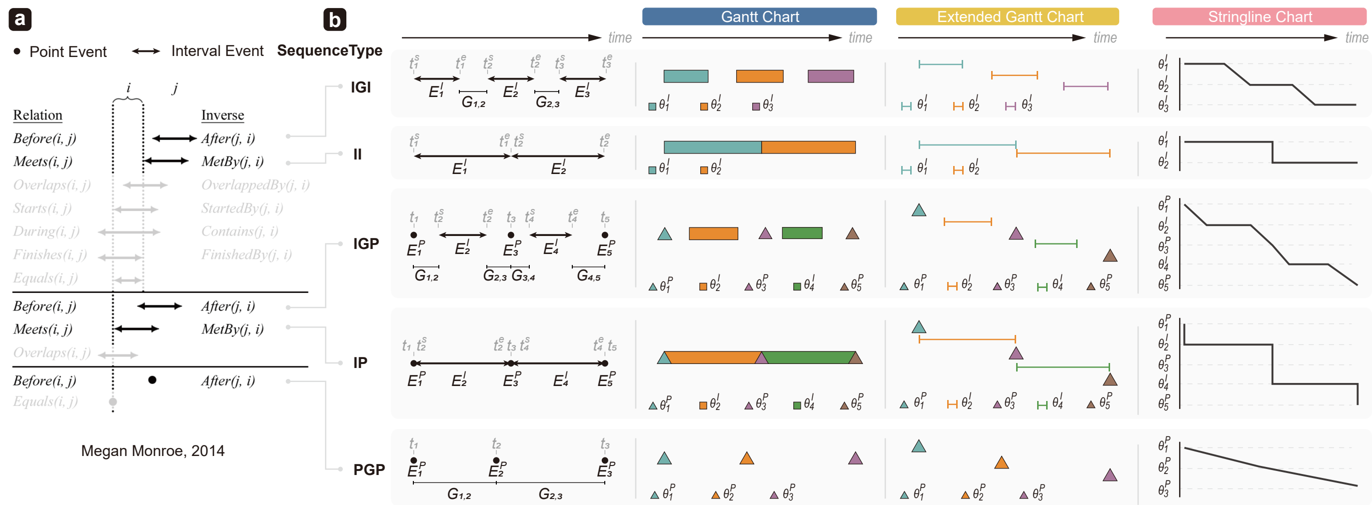


Fig. 1. (a) The interval algebra in Monroe's work [34], which is extended from Allen's version [41]. In this work, we consider five of them in serial event sequences (the black ones) and exclude others with temporal overlapping (the grey ones). (b) The five fixed-order event sequence types and how Gantt, ExtGantt, and Stringline visualize them.

III. Fixed-order Event Sequence Formulation

To present fixed-order event sequences, we adopt Monroe et al.'s classification of interval (**I**) and point (**P**) events [19], [21]. We define an event gap (**G**) between two events, which represents the temporal distance between two adjacent events and is important for identifying temporal patterns [1], [2] in fixed-order event sequences.

We introduce two more concepts, namely, the duration of an event and the event gap between two consecutive events, because these temporal features are commonly used in fixed-order event sequence data [1], [2], [6], [8], [22]. The duration of an interval event E_i^I is defined by $t_i^e - t_i^s$, and the duration of a point event is 0. The event gap $G_{i,i+1}$ between two adjacent events E_i and E_{i+1} is defined by $t_{i+1}^s - t_i^e$.

In this study, we consider the sequences without event overlapping, which widely exist in the usage scenarios of fixed-order event sequence visual analysis in previous studies [1], [2], [6], [8], [22]. Therefore, we select cases without overlapping in Allen's interval algebra [41] and Monroe's extended version [34] for this study (see Fig. 1(a)). The visual representation of events and event gaps in Gantt, ExtGantt, and Stringline charts differ in how events are encoded (see Fig. 1 (b)). Therefore, we further classify the fixed-order event sequences based on the presence of interval events, point events, and event gaps. Note that sequences that only have *Point Events without Gaps* (**PP**) means multiple events occur at similar timestamps, resulting in entirely overlapped point events (see Fig. 1(a)), so we excluded this condition. Finally, we define five types of sequences as below. For more details about the examples of each type, please refer to the supplementary material.

- *Interval Events with Gaps* (**IGI**). In this type, each event is E_i^I, and $t_{i+1}^s > t_i^e$. This type is often found in schedules that involve breaks, such as an airplane schedule [8].
- *Interval Events without Gaps* (**II**). In this type, each event is E_i^I, and $t_{i+1}^s = t_i^e$. Here events occur successively without any gaps between them.
- *Mixed Interval and Point Events with Gaps* (**IGP**). In this type, an event can be either E_i^I or E_i^P, and $t_{i+1}^s > t_i^e$. In these sequences, gaps can occur between events of either type. This sequence type exists in mixed assembly lines such as order processing pipelines [2].
- *Mixed Interval and Point Events without Gaps* (**IP**). In this type, an event can be either E_i^I or E_i^P, and $t_{i+1}^s = t_i^e$. The point events occur at the same time as the start or end of the interval events; thus, there is no gap between two adjacent events.
- *Point Events with Gaps* (**PGP**). In this type, each event is E_i^P, and $t_{i+1} > t_i$. This sequence type only includes point events, with event gaps between adjacent point events. Such sequences are common in public traffic schedules [6] and automated assembly lines [1].

IV. Study Overview

In this section, we first describe our selection of tasks. Next, we outline the variables of fixed-order event sequence data in our study. We then introduce the measurement used in both experiments. Finally, we outline the hypotheses of two respective experiments for single and multiple sequences.

A. Task Selection

We started with surveying the literature on visual analytics task taxonomy for event sequence analysis [22], [42]. We focused on tasks related to temporal information, supported by Gantt, ExtGantt, and Stringline. Tasks that required data mining were excluded from our scope. Since interactive charts bring more conditions to consider, like interaction technology and highlight methods, to narrow our focus and ensure study feasibility, we excluded tasks that require interactions in our pilot study. As a result, we identified the following tasks:

- *Retrieving Value*. Read the value of start or end timestamps of events (or event gaps) and their duration.
- *Finding Extreme Value*. Point out the longest or shortest events (or event gaps) and the earliest or latest events (or event gaps) in one sequence.

- *Comparing*. Compare two timestamps, tell which is earlier, compare two durations, and point out which is longer.
- *Visually Detecting Anomaly*. Find the event or the gap that seems abnormal compared to others.
- *Finding Temporal Patterns*. Investigate and figure out the timestamp and duration patterns among multiple sequences.

We recruited 13 participants from our institution and conducted a pilot study to examine the practicableness of the tasks. All the participants were students (8 males and 5 females). We randomly generated the five types of event sequences defined in Sec. III by varying event type counts and the numbers of sequences. Participants were assigned tasks in a randomized order. From the feedback received, we identified certain tasks within the study that either exhibited redundancy, notable straightforwardness or lacked appropriate definitions. First, participants were always required to engage in value comparison (*Comparing* task) when they were asked to identify extreme or abnormal values (*Finding Extreme Value* and *Visually Detecting Anomaly* tasks). Second, for participants to successfully perform the value comparison task (*Comparing* task), they needed to initially retrieve the values (*Retrieving Value* task). Third, we also noticed that the task was particularly salient in specific comparison scenarios, such as discerning earlier/later events or identifying event gaps. For instance, the sequence of events in fixed-order event sequences exhibited clear patterns — events located near the beginning of the sequence were naturally the earlier ones. At last, we realized the anomaly detection task is a high-level task related to domain scenarios [1], [2], which differs in metrics.

To streamline the study and address these concerns, we made the decision to retain solely the *duration-comparing* and *temporal-pattern-finding* tasks in the formal study. This choice was based on the fact that these tasks inherently encompassed the requirements of the other tasks, obviating the need for participants to obtain specialized domain expertise in order to perform them.

- For the *duration-comparing task*, we focus on duration comparisons within one sequence. The reasons are threefold. First, it is a simpler and more atomic task compared with comparing the duration between multiple sequences. The latter task has more conditions and makes the task complex when event type counts and the number of sequences is high. Second, the single sequence appears frequently during a user's highlighting or filtering interactions among multiple sequences. Third, the existing study [43] demonstrates aligning interactions can improve the speed of comparing between *multiple sequences*, so it is not reasonable to compare duration without such interactions in *multiple sequences*. As such, we examine comparison in a *single sequence*.
- For the *temporal-pattern-finding task*, we survey patterns of fixed-order event sequences in the previous studies (see Sec. IV-B), all of which exist in multiple sequence scenarios. Therefore, we examine this task in multiple sequences.

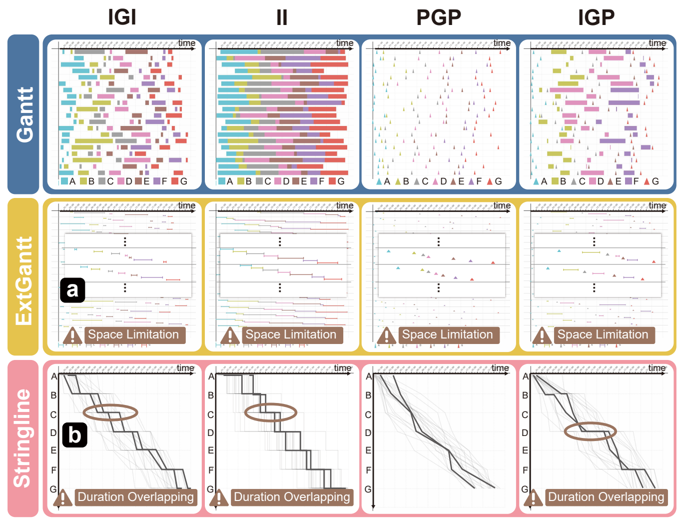


Fig. 2. When presenting multiple sequences, (a) ExtGantt, and (b) Stringline may face readability issues in certain conditions, and therefore we didn't asses them in experiments. The internal test indicates that Gantt is the more appropriate choice for presenting multiple *IGI*, *II*, and *IGP* sequences in a non-interactive environment.

B. Experimental Variables

We have five variables: one is visualization, and the other four variables are related to data, detailed below.

1) Visualization: We chose Gantt, ExtGantt, Stringline in this study as they can encode fixed-order events' temporal information and have been widely used in previous studies or real life (Sec. II). For the duration-comparing task, we chose all three visualizations. For the pattern-finding task, based on our findings in an internal test run within authors, we excluded ExtGantt as it faces space limitations due to its encoding for multiple sequences in a non-interactive environment (Fig. 2 (a)). For more details about the internal test, please refer to our supplementary material.

2) Sequence Element Type: Based on the definitions in Sec. III, there are five sequence types.

For the duration-comparing task, however, we found both *IP* and *II* can only compare interval events' duration. These two sequence types have similar visual forms of interval events, resulting in repetitive tasks. Therefore, we exclude *IP* sequence and keep *II* sequence which is more concise, to reduce the trials and make the study more practical. Overall, we consider four types of sequences in the duration-comparing task: *IGI*, *II*, *IGP*, and *PGP*. Specifically, we differentiated between semantic elements such as events and event gaps. To counterbalance possible conditions, the relative positions of these elements within the II and PGP sequences were considered distinct, because both types allowed the same elements (E or G) to be adjacent. For example, *II-nearI* means the near interval events in the II sequence while *II-apartI* means the apart ones. When dealing with *IGP* sequences, we only considered the gaps as the event tasks are similar to those in *IGI* or *PGP* sequences. Therefore, we had nine sequence element types (Tab. I) for the comparison task. The illustration and more detailed explanation of sequence element types can be found in the supplementary materials.

For the pattern-finding task, based on our findings in an

TABLE I
TRIALS COVERED IN THE TWO EXPERIMENTS.

	Sequence Element Type									Event Type Count			Number of Sequences				Temporal Pattern		Visualization			Trials
	IGI-I	IGI-G	II nearI	II apartI	PGP nearG	PGP apartG	IGP piip	IGP pipi	IGP ipip	Low (4)	Middle (7)	High (10)	Single (1)	Low (10)	Middle (50)	High (100)	Smilar Duration	Smilar Timestamps	Gantt	ExtGantt	Stringline	
E1	●	●	●	●	●	●	●	●	●	●	●	●	●	—	—	—	—	—	9*3*1*3=81	9*3*1*3=81	9*3*1*3=81	243
E2	—		—		●		—			●	●	●	—	●	●	●	●	●	1*3*3*2*3=54	—	1*3*3*2*3=54	108

I: Interval Event P: Point Event G: Event Gap — Each condition has 3 repeats.

internal test run within authors, we filtered the conditions with interval events since multiple sequences with *IGI*, *II*, and *IGP* types cannot be well visualized in Stringline (Fig. 2 (b)). It suffered from severe overlapping issues when presenting interval events, making it difficult to distinguish the duration. For more detailed illustrations, please refer to the supplementary material. As a result, we keep only the *PGP* sequence type for the pattern-finding task, which is consistent with the sequence type used in previous studies [1], [2], [6] involving Stringline.

3) Event type count: It can influence the number of colors used in Gantt and ExtGantt, and the event axes in Stringline. We only considered sequences with no duplicate event types to ensure that the number of conditions is manageable. This means that the number of events (M) within a fixed-order event sequence $\{E_1, E_2, ..., E_m\}$ is the same as the number of event types (M) $\{\theta_1, \theta_2, ..., \theta_m\}$. Overall, we covered three event type counts. Specifically, we used 4 as the smallest event type count, as this was sufficient for gap comparison tasks in *IGP* sequence. We used 10 as the largest event type count because too many colors will increase viewers' cognition burden [35], [44] and a 10-color scheme is commonly used in practice [45]. Then, we decided on 7 as the middle number of events (the median of 4 and 10).

4) Number of Sequence: This variable influenced viewers' inspections of visual patterns. Also, in cases where multiple sequences were visualized, Gantt and ExtGantt require enough viewport spaces to ensure readability. In contrast, Stringline can present multiple sequences in a limited viewport but faces visual overlapping when visualizing a large volume of sequences. Overall, we covered four numbers of sequences. Specifically, we included 10, 50, and 100 sequences for sequences with low, middle, and high numbers of sequences. We selected the maximum number of sequences of 100, considering the constraints imposed by the fixed canvas size.

5) Temporal Pattern: We identified two patterns in events or event gaps: similar timestamps and similar duration, based on previous studies [1], [2], [8]. A similar timestamp pattern refers to that the same event or gap type in multiple sequences has simultaneous start or end times, indicating a batched starting or ending of events (Fig. 3(b1)). A similar duration pattern suggests the same event or gap type in multiple sequences have a similar duration (Fig. 3(b2)).

C. Measures

Following previous studies on visualization technique comparison [46]–[49]. We used two quantitative measures:

- *Error Rate*: the proportion of the number of wrong answers to the total number of trials in that condition.
- *Task Completion Time*: the time interval between a participant first seeing the chart and submitting their answer.

D. Experiment Overview

We conduct two formal experiments based on the above-mentioned tasks and variables. Experiment 1 is to understand which chart performs better for comparison tasks on the single fixed-order event sequences among the Gantt, ExtGantt, and Stringline. Experiment 2 evaluates how Gantt and Stringline perform in revealing temporal patterns for multiple fixed-order event sequences. These two experiments share the same measures.

1) Experiment 1 - Compare Duration in the Single Sequence: Our hypotheses for Experiment 1 are as follows:

Ex1-H1: For different sequence element types, in task error rate and completion time, Gantt and ExtGantt perform better on interval element types, while Stringline performs better on gap-related element types.

We suspect the colors being more visually salient improve performance with Gantt and ExtGantt for intervals. However, because gaps are not directly encoded at all in these two charts, they might perform worse than Stringline where gaps are the slash lines.

Ex1-H2: Gantt and ExtGantt perform worse than Stringline with larger event type counts in task error rate and completion time. We suspect this is because too many colors of encoding various event types in Gantt and ExtGantt can result in a high cognitive load.

2) Experiment 2 - Find Patterns in Multiple Sequences: For the task to find *similar timestamp patterns*, both charts have apparent visual cues (Fig. 3(b1)). Note that we excluded ExtGantt in this experiment, justified in Sec. IV-B. In Gantt, the "line" formed by the same colored triangles indicates this pattern. In Stringline, the lines' intersection point illustrates the pattern. Based on these observations, we had the following hypotheses:

- ***Ex2-H1***: For different event type counts, the two charts exhibit similar performance in error rate and completion time in finding similar timestamp patterns.
- ***Ex2-H2***: For different numbers of sequences, the two charts exhibit similar performance in error rate and completion time in finding similar timestamp patterns.

For the task to find the *similar duration pattern*, the parallel lines in Stringline are more apparent than the distance between two colored triangles in Gantt (Fig. 3(b2)). According to Gestalt Principles [50], the clustered visual patterns of parallel lines can make Stringline more noticeable in this comparison. As such, we formed the following hypotheses:

- ***Ex2-H3***: For different event type counts, Stringline outperforms Gantt in task error rate and completion time in finding similar duration patterns.

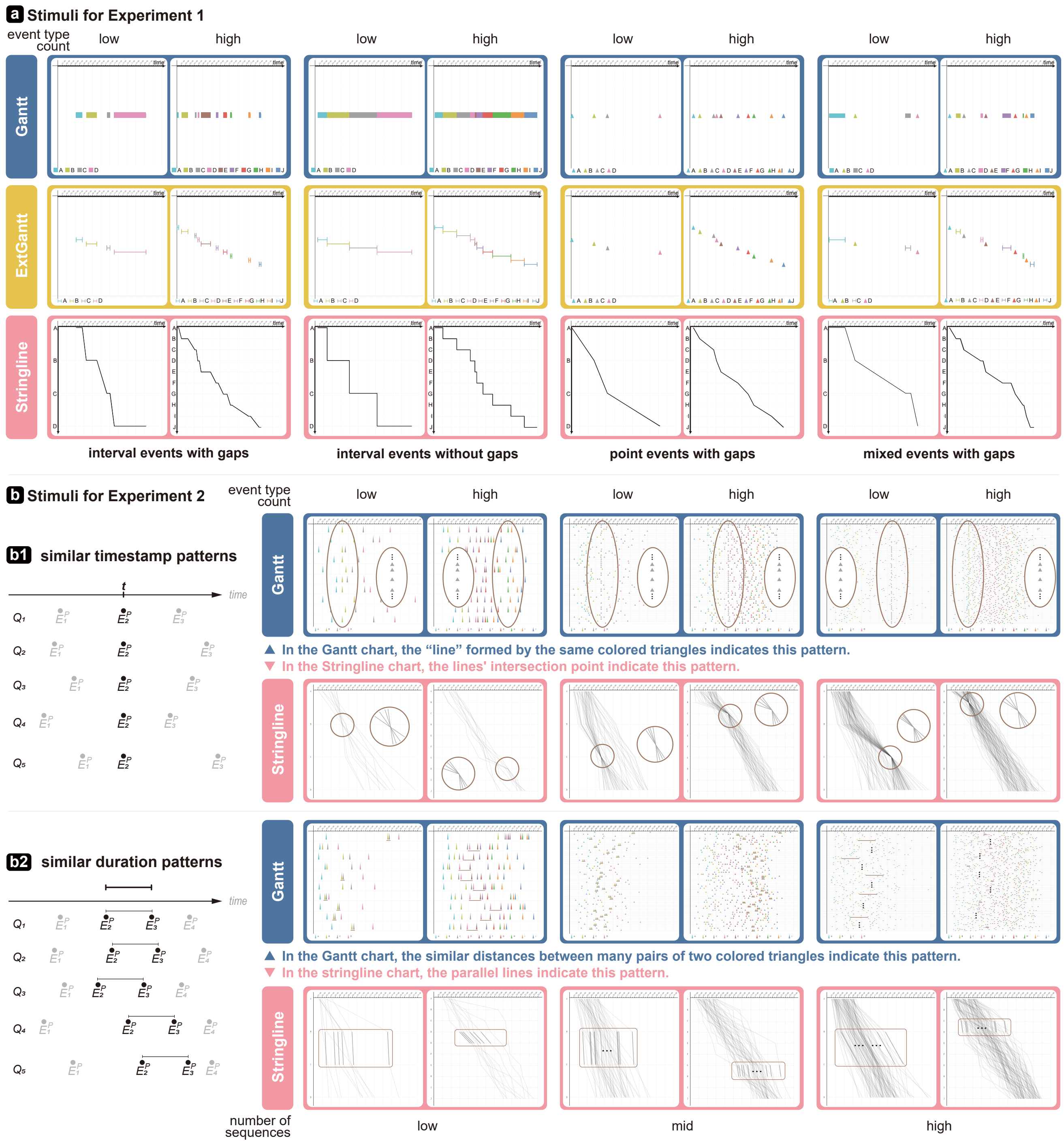


Fig. 3. The example stimuli of low and high event type counts in (a) Experiment 1 and (b) Experiment 2. We scale the main visual contents of trials for better presentation. We provide the original whole set of stimuli, including medium event type counts, in the supplementary materials.

- ***Ex2-H4***: For different numbers of sequences, Stringline outperforms Gantt in task error rate and completion time in finding similar duration patterns.

V. Experiment Design

In this section, we detail the design of the two experiments (Ex1, Ex2), including the procedures, participants, and stimuli. We provided detailed supplementary materials, including tutorial videos, stimuli, experiment interface screenshots and demos, prompts, analysis code, and statistical results.

A. Experimental design and procedures

Our study employed a within-subject design, where each participant completed the two experiments (see in Sec. IV-D) in sequence. The experiment began with a video introducing event sequence data and visualizations. Participants had to watch it before moving on to the next stage. To reduce fatigue, we divided all trials into five sessions, each containing only trials of the same chart type. Therefore, we had three sessions (Gantt, ExtGantt, Stringline) for Experiment 1 and two (Gantt, Stringline) for Experiment 2 (Fig. 4).

First, they watched a tutorial video. For Experiment 1,

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3358919

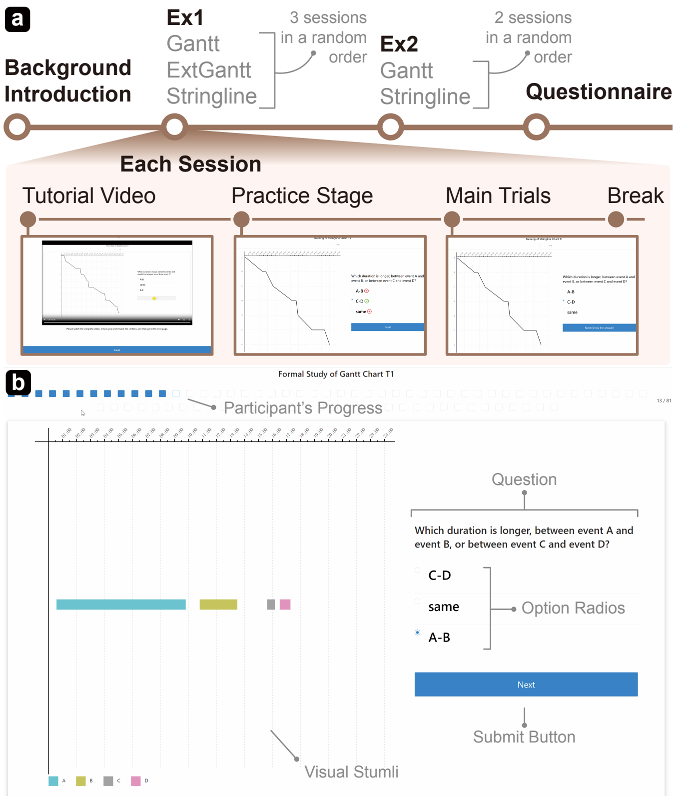


Fig. 4. (a) The procedures for both experiments and (b) the study's interface. For more comprehensive screenshots and demonstration videos, please refer to the supplementary materials.

the video introduced chart encodings and how to use the experiment interface. For Experiment 2, the video explained temporal patterns and how they were visualized in the chart. Second, they did the practice trials to familiarize themselves with the interface and charts. In this practice stage, we provided correct answers for participants after they answered to help them check their understanding. In each session, there were 12 practice trials. Third, they started the main trials. In this stage, we recorded the time participants spent in each trial. To accurately record the time interval for answering, we required participants to read the trial question, chart legends, and axes presented in advance on the canvas and then to click a button to load the stimulus. Then, they should choose one answer from three options and proceed to the next one. After completing each session, participants were suggested to take a break. We randomly assigned participants in the order of chart types. At the end of two experiments, participants completed a demographics questionnaire and provided comments. All of these procedures were notified to participants at the beginning. The two experiments lasted around 100 minutes.

In Experiment 1, we evaluated 3 event type counts and 9 sequence elements, with 3 repeats for each condition. As a result, we had 3 (event type counts)$\times$9 (sequence element types)$\times$3 (repeats)$\times$3 (chart types) = 243 trials. In Experiment 2, we assessed 3 event type counts and 3 numbers of sequences, with 3 repeats for each. As such, we had 3 (event type counts)$\times$3 (numbers of sequence)$\times$3 (repeats)$\times$2 (pattern types)$\times$2 (chart types) = 108 trials.

B. Participants

We recruited participants from an introductory visualization class and the university's internal bulletin board system, providing them with partial course credit or financial compensation (12 USD per hour). Participants were sent a link to the web-based experiment system and asked to complete it on their laptop or desktop. As a result, this could cause variations in environmental conditions, such as screen size and lightness, among participants. However, existing literature demonstrated the validity of crowd sourcing studies for graphical perception [51].

By the recruitment deadline, we recruited 93 participants. Among them, there are 60 males, 31 females, and 2 chose not to disclose their biological sex. Their ages ranged from 19 to 27, with an average of 21.16. 87% (81/93) of participants reported prior experience with event sequence visualization (*know but not use: 57, know and use: 19, knowledgeable: 4, expertise: 1*). Specifically, 77 have seen or used Gantt, 56 for ExtGantt, and 53 for Stringline.

C. Stimuli

1) Trial Data: We used synthetic data for the study to maintain a reasonable number of trials and ensure consistency across conditions (Fig. 3). To ensure readability while maintaining randomness, we adhered to certain constraints (Sec. V-C3) during the generation of event durations and timestamps within a single sequence. Under the readability constraint, we randomly generated the duration of the events and their timestamps within [00:00, 24:00] in one sequence. Once the event timestamps are determined, the duration and timestamps of event gaps are decided accordingly. Further detail of our pattern generation in Experiment 2 is included as part of the supplemental material.

2) Trial Questions: For each trial, the question and the three options were generated randomly. In Experiment 1, the question is *"Which duration is longer, event X or event Y?"* or *"Which duration is longer, between event X and event Y, or between event Y and event Z?"*. We randomly selected two events or gaps to be compared in the question. The options presented to participants consisted of each of these selected events or gaps, along with one option labeled as *"the same."* In Experiment 2, the question for similar timestamp patterns is like *"Which event has the most sequences with the same timestamp?"*, while the question for similar duration patterns is like *"Between which two events, the duration between them are always the same?"*. The options included the patterned event or gap and two other randomly generated events or gaps within the trial sequences.

3) Stimuli Charts: To have a consistent visual searching area for all the conditions, we rendered the stimuli charts on a square canvas with a fixed ratio (0.85) to the screen height. Moreover, to ensure the duration of interval events and event gaps are readable in the three charts, we fixed the shortest horizontal length of visualization shape (rectangles in Gantt, line in ExtGantt, and line in Stringline) to 10 pixels. In order to ensure visibility and differentiation through color mapping, the height of the triangles in the Gantt chart was set to be larger than 5 pixels. If a participant's equipment did not meet these readability constraints, they would be unable to take part in the study.

For all the trials, we utilized the color palettes provided by Tableau 10 for visual representation [45], which was widely used in visualization practice. No interactions were incorporated into the stimuli chart (see Sec. IV-A). Regarding the event labels, we used alphabet letters as they could correspond to a predefined order established for the events. This choice facilitates intuitive identification and understanding of the event sequence, enabling participants to interpret the chart easily. To assist participants in retrieving the time values, we provide vertical dashed lines that align with the ticks on the time axis. These lines serve as visual references, helping participants locate time.

VI. Analyses and Results

A. Analyses

We analyze, report, and interpret the results using interval estimates [52]. We first average the results of three repeats for each condition. For each condition, we report the means of error rate and completion time. We calculate all the 95 confidence intervals (CIs) through BCa bootstrapping with 10,000 iterations. We also compare the error rate and completion time between each two of the tested charts. Following the pairwise comparison methods that previous studies adopt [46], [52], we present the differences in error rates and the ratio of completion time. Note that in our analysis, we log-transformed the completion time and anti-logged them in our reporting.

We also perform significance testing as additional analysis. Because our data violates the normality assumption, we choose the Friedman test for Experiment 1 and the Wilcoxon test for Experiment 2, with Bonferroni-corrected p-values for statistical tests. We planned all the analysis methods before we practiced the experiments. All the codes and data can be found in the supplementary materials.

B. Experiment 1 Results

We examine duration-comparing tasks regarding sequence element types and event type counts, as we hypothesized in Sec. IV-D1.

Ex1-H1 (Fig. 5 (a)): The results show that Stringline has lower mean error rates than Gantt and ExtGantt for all interval-event-related elements (e.g., the mean error rate of Stringline is *0.48 [0.0, 1.19]%* (*IGI-I*), *0 [0, 0]%* (*II-nearI*), *0.36 [0.0, 0.72]%* (*II-apartI*)). We do not have conclusive results about the error rate winner between Gantt and ExtGantt, as the pairwise CIs overlay the zero. On the other hand, Stringline has a higher mean completion time than Gantt and ExtGantt across all sequence types, particularly in the three *IGP* sequence element types (e.g., the completion time of Stringline is *4.19 [3.83, 4.6]s* (*IGP-piip*), *4.91 [4.47, 5.46]s* (*IGP-pipi*), *5.92 [5.41, 6.54]s* (*IGP-ipip*). Gantt has similar mean completion times to ExtGantt for element types such as *IGI-G* (Gantt: *3.63 [3.31, 4.02]s*), *PGP-apartG* (Gantt: *3.5 [3.18, 3.9]s*), *IGP-piip* (Gantt: *3.26 [2.96, 3.67]s*), and *IGP-pipi* (Gantt: *3.75 [3.42, 4.11]s*), but lower mean completion times for other element types. These findings partially support our hypothesis ***Ex1-H1***: Stringline requires more time than the other two charts in all the element types except for the pairwise comparison with ExtGantt in *II-nearI* (*1.03 [0.97, 1.12]*$\times$). However, Stringline has lower error rates than the other two charts in interval-event-related elements but no significant difference in gap-related elements, which mismatches our initial hypothesis.

Ex1-H2 (Fig. 5 (b)): Compared to Gantt and ExtGantt, Stringline has marginally higher mean error rates for low (Stringline: *1.43 [0.96, 2.03]%*) and middle (Stringline: *2.95 [2.31, 3.58]%*) event type counts but lower mean error rates for high event type counts (Stringline: *1.95 [1.31, 2.67]%*). However, across all event type counts, Stringline showed longer completion times than both Gantt and ExtGantt (e.g., the completion time of Stringline is *3.3 [3.06, 3.62]s* (low), *4.07 [3.78, 4.44]s* (mid), *4.47 [4.09, 4.92]s* (high), and the completion time ratio of Stringline and the other charts decreases with event type counts increasing. For the ratio in completion time, ExtGantt needs a bit more (*1.09 [1.03, 1.15]*$\times$) than Gantt in the middle event type count. No other significant differences are observed between Gantt and ExtGantt. These findings partially support our hypotheses that Gantt and ExtGantt perform worse than Stringline with larger event type counts in error rate.

a) Qualitative Feedback: In this experiment, participants first needed to find the events mentioned in the chart and then compare the duration of them. Participants were required to understand the chart encoding, including visual elements that represent the event types and indicate the duration length. Several participants expressed their preference for the encoding of event types and the easiness of comparing gap duration. First, four participants felt Stringline is more user-friendly for distinguishing event types compared to Gantt and ExtGantt. They reported that performing tasks with Gantt and ExtGantt required extra time on color mapping. Particularly when confronted with a multitude of event types, they found that "*in comparison to the color mapping utilized in Gantt, the visual identification of event types in Stringline is more direct and perceptible due to its positional and visual search simplicity.*" Specifically, as shown in Fig. 3(a) they had first to look up the color legend, and then match two same colors, finally retrieve event type information from the color legend, which is at the bottom of the stimuli, far away from the visualization. In contrast, Stringline allows participants to directly follow the horizontal lines aligned to the vertical event type marks, which are much closer and are at regular positions. While ExtGantt does organize events of different types along the vertical positions, they are mainly encoded by color rather than position. Therefore, the vertical position serves only as a reference rather than a concrete way of accessing event types. On the other hand, three participants argued that Gantt and ExtGantt's colored symbols are more prominent and noticeable than the uncolored lines in Stringline. Second, five participants found it easier to compare near event gaps in PGP sequences (*PGP-nearG*) using Stringline compared to Gantt and ExtGantt. In Stringline, event gaps are represented by slash lines, whereas, in Gantt and ExtGantt, event gaps are shown as the empty spaces between two colored symbols. This is because Stringline provided double visual references: the horizontal length of the

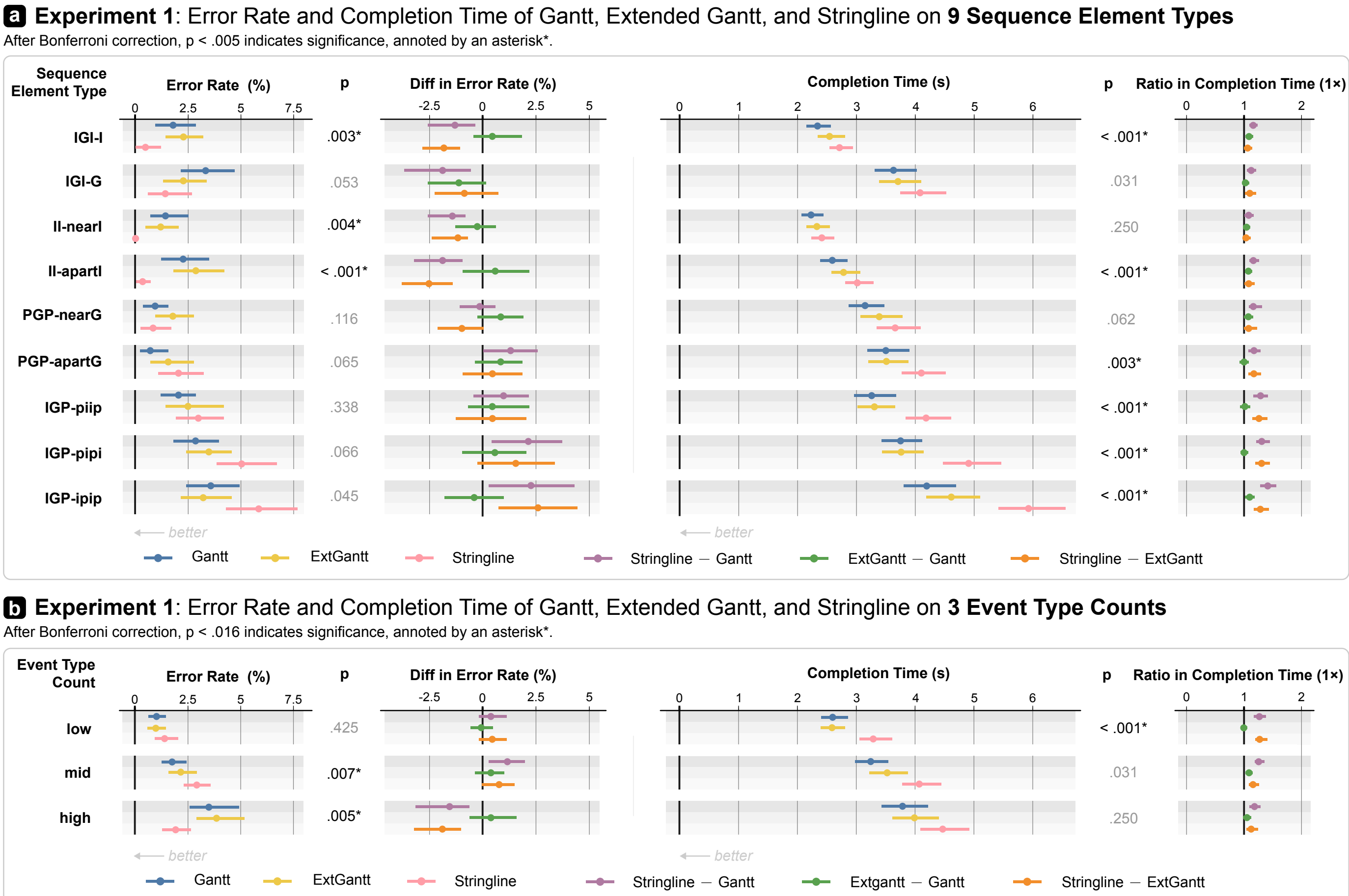


Fig. 5. The quantitative result of the duration-comparing task in Experiment 1 about (a) nine element types and (b) three event type counts. Left: Mean Error Rate and pairwise comparisons (difference). Right: Mean Completion Time and pairwise comparisons (ratio). We computed means, 95% bootstrap confidence intervals, and the p values of the Friedman test.

slash and the slope of the line. A bigger angle formed by the line and the horizontal axes indicates a shorter duration of an event gap. Thirteen participants commented that the slope is useful when comparing two adjacent slashes, as they could draw a conclusion through whether the angle formed by the adjacent slashes is obtuse or not. However, one participant mentioned that the slope difference could not work well if the compared duration was too similar.

b) Discussion: For the duration-comparing task in single sequences, the effectiveness of the three charts depends on the compared elements' relative positions, the encoding of sequence elements, and the familiarity of these charts. First, comparing elements with a common reference is easier than those without one. For example, the error rate and completion time of the three charts in *II-nearI* are lower than in *II-apartI*. Similar cases are also observed for *PGP-nearG* and *PGP-apartG* sequences. Second, when the number of events is high, excessive use of colors may impact the effectiveness of the charts. Both encoding event types by color, Gantt and ExtGantt, have similar error rates and completion time performance when the event type count is low, both lower than Stringline. In contrast, when the event type count is high, Gantt and ExtGantt have significantly higher error rates than Stringline. This may be attributed to the extra perceptual burden of distinguishing too many colors in Gantt and ExtGantt in high event type count situations. Another evidence of the influence that color issues may bring is that though Stringline consistently requires more time than the other two charts, the difference decreases when the event type count is higher. Third, illustrated by the demographic questionnaire result, almost half of the participants never knew about Stringline (37) and ExtGantt (40), compared to Gantt (16). As a result, Stringline requires more completion time than the other charts. In contrast, the performance of ExtGantt is similar to Gantt in conditions such as low event type counts, etc. Participants' prior Knowledge affected their comprehension and utilization of these charts.

C. Experiment 2 Results

We tested two temporal patterns: *similar timestamps* pattern and *similar duration* pattern. In each pattern, we analyzed Gantt's and Stringline's performance in different sequence and event type counts (Sec. IV-D2).

Similar Timestamp Pattern

Ex2-H1 (Fig. 6 (a1)): Stringline has a lower mean error rate (*0.24 [0, 0.60]%*) than Gantt (*3.82 [2.75, 5.26]%*) in the high event type count condition, and less mean completion time (e.g., high: *1.99 [1.90, 2.11]s*) than Gantt (e.g., high: *5.58 [5.22, 6.02]s*) across all three event type counts, which partially rejects our hypothesis that the two charts have no significant difference in the performance in finding similar timestamp patterns for different event type counts.

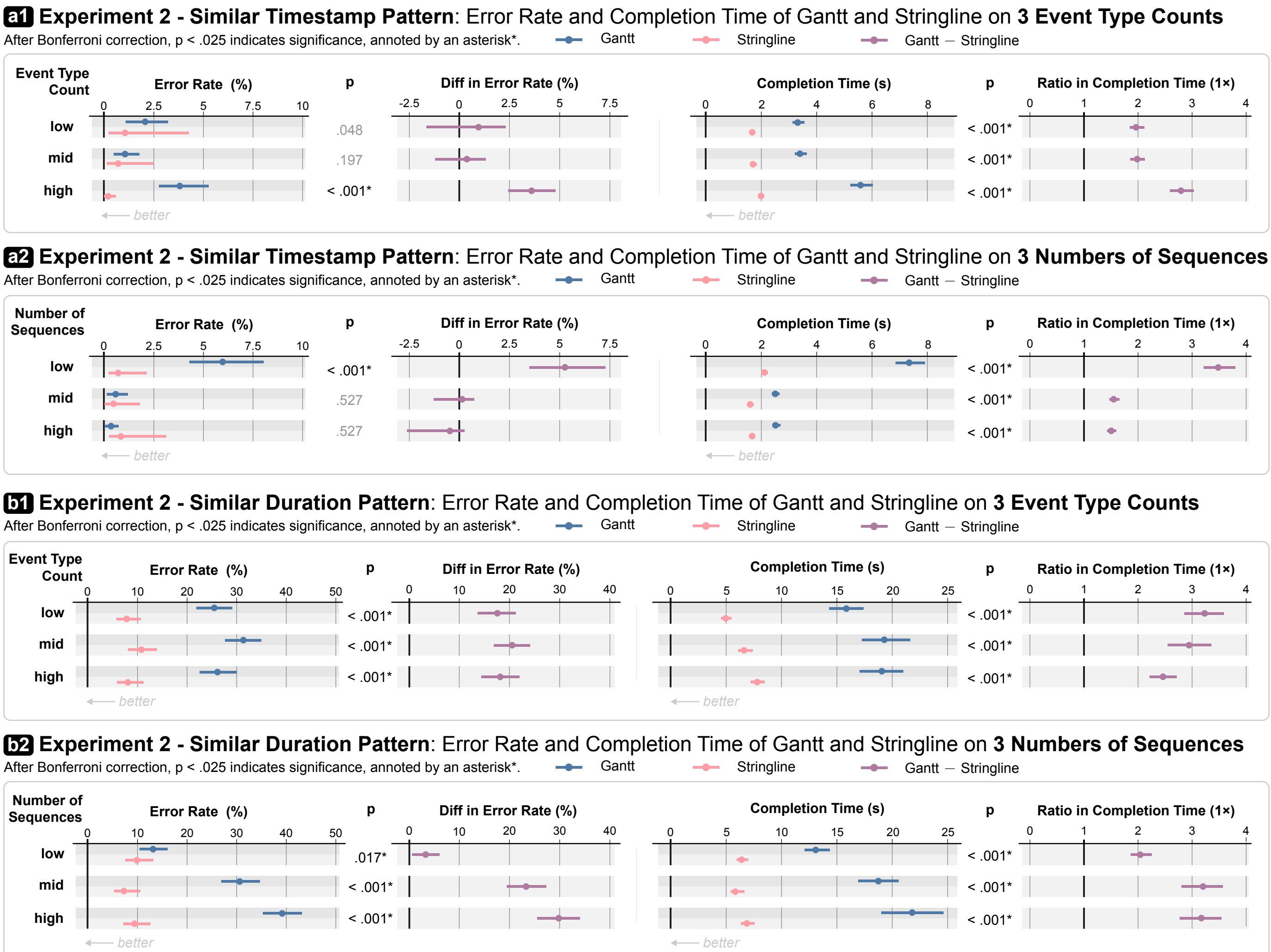


Fig. 6. The quantitative result of the temporal-pattern-finding task in Experiment 2. (a) Similar timestamp pattern in (a1) three event type counts and (a2) three numbers of sequence. (b) Similar duration pattern in (b1) three event type counts and (b2) three numbers of sequence. Left: Mean Error Rate and pairwise comparisons (difference). Right: Mean Completion Time and pairwise comparisons (ratio). We computed means, 95% bootstrap confidence intervals, and the p values of the Wilcoxon test.

Ex2-H2 (Fig. 6 (a2)): Stringline has a lower mean error rate (*0.72 [0.24, 2.15]%*) than Gantt (*5.97 [4.3, 8.0]%*) when the number of sequences is low. Moreover, Stringline (e.g., low: *2.10 [1.99, 2.23]%*) has a lower mean completion time than Gantt (e.g., low: *7.33 [6.85, 7.90]%*) across all three numbers of sequences. Thus, the result rejects the hypothesis that for different numbers of sequences, the two charts exhibit no significant differences in task error rate and completion time in this task. The completion time ratio of Stringline to Gantt is larger in the low number of sequence (*3.49 [3.22, 3.8]*$\times$) compared to the middle (*1.55 [1.47, 1.66]*$\times$) and the high number of sequence (*1.5 [1.42, 1.59]*$\times$).

Similar Duration Pattern

Ex2-H3 (Fig. 6 (b1)): For all three event type counts, Stringline outperforms Gantt, with lower mean error rates (e.g., Stringline: *7.89 [5.73, 10.75]%* (low)) and shorter mean completion time (e.g., Stringline: *4.91 [4.50, 5.40]%* (low)). This confirms our hypotheses that for different event type counts, Stringline performs better than Gantt in both error rate and time in this task. Another pattern is that the ratio of completion time between the two charts becomes smaller with the event type count increasing.

Ex2-H4 (Fig. 6 (b2)): Stringline has lower mean error rates (e.g., Stringline: *9.44 [7.17, 12.54]%* (high)) and shorter mean completion times (e.g., Stringline: *6.87 [6.32, 7.57]s* (high)) than Gantt across all three event type counts, thus supporting this hypothesis that for different event type counts, Stringline performs better than Gantt in both error rate and time in this task. Notably, in the low number of sequence conditions, the difference in mean error rate between the two charts (*3.23 [0.48, 5.97]%*) is smaller than in the middle (*23.3 [19.47, 27.36]%*) and high numbers of sequence (*29.75 [25.45, 34.05]%*). As the number of sequences increases, the difference in mean error rate becomes larger.

a) Qualitative Feedback: In this experiment, participants first needed to find the temporal pattern mentioned in the chart and then recognize the event type. They were required to understand the encoding of event types and timestamps. Several participants reported their strategies for recognizing patterns.

For similar timestamp patterns, seven participants reported that their strategy in Gantt was to find the vertical line formed by the same colored symbols, while in Stringline, they looked for the intersection of the polylines, because "*the same position on the timeline indicates the same timestamp.*" The ease of finding this pattern was related to the number of sequences, as more sequences increased their confidence in the finding. One participant mentioned, "*...when the number of sequence is low, I'm not sure if there is a pattern at first glance and have to check each sequence individually...*" Overall, they found this pattern easy to identify because when events occur at the same timestamp, both charts' visual cues were intuitive and apparent.

On the other hand, for similar duration patterns, six participants reported the pattern apparent in Stringline, while 42 found it too difficult to identify it in Gantt. One participant commented, "*It seems challenging to find the two events with similar distances among many colored symbols.*" Another participant complained the sequences seemed too crowded in Gantt. Agreed with the color clutter, another participant thought that to complete such a task correctly in Gantt, interactions like filtering or aligning might help. For this pattern, participants shared their strategies. In Stringline, they found the lines with similar slopes, while in Gantt, they attempted to figure out a band between the same composition of two colors in each row. One participant also mentioned that she sometimes moved her head back to see the chart overview and to find if the lines formed by colored symbols had a similar shape, which indicates a similar gap duration in multiple sequences. However, if she moved too far from the screen, the symbols became too small to distinguish the colors.

b) Discussion: For the temporal-pattern-finding task, the effectiveness of the two charts is closely related to the clarity of visual cues. First, the density of visual cues tied to the number of event sequences, as reported by some participants who felt they found patterns more quickly when the number of sequences was higher. The results in similar timestamp patterns also indicate this phenomenon. A low number of sequences leads to sparse visualization content, obscuring cues to locate patterns. However, this does not mean the higher the number of sequences, the easier it is for viewers to find temporal patterns, as a large number of sequences bring scalability issues for both charts. For Gantt, a canvas can only present limited sequences, necessitating extra assistance like interactions like filtering and alignment, or a larger canvas for an overview search of all the sequences for patterns. For Stringline, sequences can generate overlapping on each other, therefore masking important cues for pattern finding. Second, the encoding of the two charts also influences the appearance of visual cues, further impacting their effectiveness in revealing temporal patterns. The layout of Stringline, such as the horizontal event type axes, provides viewers with a reference for comparing the duration of event gaps. The consistent vertical distance between these axes also improves the readability of event gaps. In contrast, the non-interactive Gantt lacks a reference and can be limited by the clutter caused by an overabundance of colors.

VII. General Discussion

A. Design Opportunities

Stringline can be used in comparing tasks in *IGI* and *II* fixed-order sequence types to get a lower error rate. According to the experiment result reported in Sec. VI-B, Stringline can have a lower task error rate for interval-event-related fixed-order sequence types (*IGI* and *II*) when performing duration-comparing tasks in a single sequence. However, there is a trade-off between a lower error rate or less completion time in practice because, given completion time, Stringline is the slowest compared to other charts in such sequence types.

Gantt and ExtGantt can be used in comparing tasks in fixed-order event sequences with a few event type counts. As the results indicate, Gantt and ExtGantt perform better in error rate and completion time than Stringline in low (n=4, marginally in error rate) and middle (n=7) event type counts but have higher error rates in high (n=10) event type counts. Also, the results show that sequences with fewer event type counts in both Gantt and ExtGantt, which rely on colors as the event category encoding, have a lower error rate and require less completion time than those with higher event type counts. If there are too many events, aggregating related or adjacent events can effectively address the color clutter issue, as applied in previous work [19], [21].

Stringline can be used to reveal temporal patterns in multiple PGP fixed-order event sequences. The result shows that Stringline is more suitable for finding temporal patterns in *PGP* fixed-order event sequences, especially when multiple event types are presented. This is because Gantt and ExtGantt may face issues like space limitations and color clutter. Furthermore, if the patterns have not been precisely known, an apparent visual cue in visualization would be helpful for pattern identification. In such cases, Stringline can provide an overview without requiring extra alignment, making it superior to the other two charts. However, if the number of sequences is too large, all three charts may fail to reveal patterns. Instead, layout algorithm like KDE [6] and aggregation [1], [8], [21], or interactions [2] might be utilized to address visual clutter.

B. Reflections on Fixed-order Event Sequence Visualization

We may **transform fixed-order event sequence types for better presenting the source data.** The data attributes recorded in real scenarios and domain requirements could affect the sequence type of a dataset. For example, to decrease data redundancy, the *II* sequence dataset can only record the event's start timestamp because the end timestamp is the start timestamp of the next event (see Fig. 1(b)), resulting in the same data format as PGP sequences. This means that the duration between two timestamps can be either considered as an interval event or an event gap. Generally, whether it is an interval event or event gap is based on domain knowledge. However, suppose domain requirements allow semantic transformation between events and event gaps. In that case, transforming them can lead to a broader perspective to explore the source data because the visualization of these two sequence types has different visualization forms, resulting in more possibilities for visualizing the source data and a concise choice based on the design opportunities aforementioned.

In addition to a PGP sequence, **Stringline might be used to visualize multiple fixed-order event sequences of other types.** As indicated in Fig. 2(b), Stringline can face overlapping issues when visualizing interval events in the multiple sequence condition. Instead, previous work used to visualize interval event sequences (such as *IGI* and *II*) by Gantt or ExtGantt [2], [8], [21]. Nonetheless, our study found that Stringline conditionally outperforms others. Specifically, it has advantages in comparing interval events in a single sequence and revealing temporal patterns of point events and event gaps in multiple sequences.

Given its advantages in revealing temporal patterns and disadvantages in interval event overlapping, there are values to optimize Stringline or its variants for interval events in multiple sequences. Such optimization can involve color encodings and layout rearrangements inspired by related methods in line-based temporal visualization proposed in previous studies, such as colored waterfall [53] and stacked lines [54], [55]. More effective designs for visualizing fixed-order event sequences in Stringline require further studies.

We can **make use of the unchanging event order of fixed-order event sequence data in Gantt and ExtGantt.** Gantt and ExtGantt encounter challenges related to color clutter when event type counts are high. Nevertheless, fixed-order event sequences possess a distinct advantage: their unchanging event order can serve as a latent reference for encoding event types, for instance, through sequential color mapping, thereby conveying sequential information. However, selecting an appropriate sequential color mapping method necessitates further investigation. This is due to the potential similarity between nearby colors in a sequential color map, which could impose a visual burden on people to discern specific event types and potentially impact temporal pattern recognition.

C. Limitations and Future Work

Our study is an initial step towards bridging the gap between the theoretical literature and visual analysis practice for fixed-order event sequence data visualization. The findings verify the design used in previous studies and inspire potential for new applications. Below we discuss the limitations of our work and opportunities for future work.

First, all of the participants in our experiments are students from one university, and most of them are from a data visualization course. The advantage is the guarantee of data quality, while the disadvantage is the narrow range of occupations, which could have biased how they understand and approach the tasks. Besides, as we discussed in Sec. VI-B, the differences in task completion time may be due to the user's less familiarity with the charts. Although we prepare tutorials and practice for participants to ensure they understand the encoding of the chart, we cannot enable them to use the chart in a short time skillfully. Further investigation is needed to provide more evidence for the completion time difference between these charts over the long term.

Second, the tutorial video could potentially influence how participants comprehend the charts and find patterns. We didn't mention the visual cues or patterns in the tutorial video and question. Instead, our trial questions focused on the data itself (see Sec. V-C2). However, we circled roughly where the patterns were for illustration purposes. While participants' comments indicate their interpretation of data and chart encoding, some of them might have taken a shortcut and used similar visual cues during the formal trials.

Third, as the static version is basic for viewers' reading visualizations, we excluded the interaction variables in this work to make conditions under a practical level, but charts in visual analysis systems can be interactive. How interactions can help to finish visual analysis tasks in these charts is worthy of evaluation in future work. Moreover, to keep the constant, we present charts in a limited and fixed canvas, but realistic scenarios can be more diverse with various canvas sizes and length-width ratios.

Forth, we selected Gantt, ExtGantt, and Stringline and employed their fundamental encodings in our study. These choices were influenced by their prior use in relevant studies or their suitability for visualizing fixed-order event sequences [1], [2], [6], [8], [18]–[21]. However, due to limitations in terms of space and readability, we had to exclude a list of conditions related to visualization and sequence element variables, i.e., extended Gantt chart for multiple sequences and stringline chart for interval-based event sequences. In our internal test, Gantt appeared to be the best choice for multiple interval sequences. Yet, we also believe that addressing these limitations in ExtGantt and Stringline will necessitate future research. Conclusions drawn from this study may require validation when improved designs to overcome these limitations become available.

Another limitation is that we used synthetic data to comprise the readability of stimuli on one screen and real-world data complexity. We only evaluated 1, 10, 50, and 100 sequences with 4, 7, and 10 events, and fixed the shortest horizontal length of the visualization shape to ensure the duration of events and gaps are readable. If certain conditions within a specific chart become unreadable, it signals that the chart might not be the optimal choice and could benefit from necessary optimization. Future work might extend and validate our results on real-world datasets.

Finally, this study evaluates Gantt, extended Gantt, and stringline charts on five kinds of serial fixed-order event sequences, which differ in the composition of interval events, point events, and event gaps. While the scope of sequence data is broad, our work focuses on a specific branch of event sequences with two characteristics. First, events occur in a sequential order without temporal overlaps. Second, event types follow a predetermined and fixed order without interleaving and the number of events is the same as the number of event types in a sequence. Such sequences resemble linear progress found in various contexts, such as automatic pipelines in industry, cascading processing workflows in logistics, a schedule of public transportation, and the growing status in biology, demonstrating the common existence of fixed-order event sequences and the importance of understanding its visualization. The two data characteristics enable the stringline chart to visualize such data, as indicated by previous studies [1], [2], [6]. This explains why we specifically target such event sequences and include the stringline chart in addition to the Gantt chart and extended Gantt chart, which is designed for more general sequence data [2], [8], [21].

However, it is important to note that the findings of this study may not apply to other types of sequence data without these two characteristics (like sequences with unpredefined and inconsistent event orders), as the stimuli for the sequences can differ from those considered under experimental conditions. The application of these visualizations to other types of sequence data and their performance in those contexts require detailed discussion on unpredictable event orders, event overlapping, etc., and further investigation of visualization literacy, which is beyond the scope of this study.

VIII. Conclusion

This work presents a comparative study on Gantt, ExtGantt, and Stringline charts in visualizing fixed-order event sequence data. Our study scope encompasses five types of event sequences, which serve as the basis for conducting two experiments involving a total of 93 participants. These experiments aim to evaluate the effectiveness of these visualizations in terms of comparing duration and revealing duration and timestamp patterns.

Our experiments find evidence that for the duration-comparing task, Gantt and ExtGantt require less completion time than Stringline, but Stringline has lower error rates in interval-event-related sequence elements and high event type count conditions. For the temporal-pattern-finding task, Stringline chart exhibits less completion time and lower error rates compared to Gantt. Specifically, for the similar timestamp pattern, Stringline had a lower error rate than Gantt in two conditions: high event type counts and low number of sequences. For the similar duration pattern, Stringline outperformed Gantt in both error rate and completion time in all experimental conditions. Based on our findings, we provide several reflections on the utilization of these charts for different fixed-order event sequence data and tasks, as well as future avenues for their improvement. Our study outcomes can serve as a valuable reference for future research and practitioners seeking to employ appropriate visualizations for fixed-order event sequence data.

IX. Acknowledgments

The work was supported by NSFC (U22A2032, 62272396), Key “Pioneer” R&D Projects of Zhejiang Province (2023C01120), and NSF 2127309 to the Computing Research Association for the CIFellows Project.

References

[1] P. Xu, H. Mei, L. Ren, and W. Chen, “Vidx: Visual diagnostics of assembly line performance in smart factories,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 291–300, 2017.

[2] J. Tang, Y. Zhou, T. Tang, D. Weng, B. Xie, L. Yu, H. Zhang, and Y. Wu, “A visualization approach for monitoring order processing in e-commerce warehouse,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 857–867, 2022.

[3] F. Zhou, X. Lin, C. Liu, Y. Zhao, P. Xu, L. Ren, T. Xue, and L. Ren, “A survey of visualization for smart manufacturing,” *Journal of Visualization*, vol. 22, p. 419–435, 2019.

[4] F. Chen, J. Li, F. Wang, S. Liu, X. Wen, P. Li, and M. Zhu, “Warehouselens: visualizing and exploring turnover events of digital warehouse,” *Journal of Visualization*, vol. 26, p. 977–998, 2023.

[5] Y. Wang, Z. Zhu, L. Wang, G. Sun, and R. Liang, “Visualization and visual analysis of multimedia data in manufacturing: A survey,” *Visual Informatics*, vol. 6, no. 4, pp. 12–21, 2022.

[6] C. Palomo, Z. Guo, C. T. Silva, and J. Freire, “Visually exploring transportation schedules,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 170–179, 2016.

[7] M. Barry and B. Card, “Visualizing MBTA data: An interactive exploration of boston’s subway system,” Website, June 2014, retrieved January 5th, 2024 from https://mbtaviz.github.io/. [Online]. Available: https://mbtaviz.github.io/

[8] J. Jo, J. Huh, J. Park, B. Kim, and J. Seo, “Livegantt: Interactively visualizing a large manufacturing schedule,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2329–2338, 2014.

[9] W. Zhao, G. Wang, Z. Wang, L. Liu, X. Wei, and Y. Wu, “A uncertainty visual analytics approach for bus travel time,” *Visual Informatics*, vol. 6, no. 4, pp. 1–11, 2022.

[10] G. Zuccaro, D. De Gregorio, and M. F. Leone, “Theoretical model for cascading effects analyses,” *International Journal of Disaster Risk Reduction*, vol. 30, pp. 199–215, 2018, understanding and mitigating cascading crises in the global interconnected system.

[11] B. Huppertz, H.-G. Frank, and P. Kaufmann, “The apoptosis cascade—morphological and immunohistochemical methods for its visualization,” *Anatomy and Embryology*, vol. 200, no. 1, pp. 1–18, 1999.

[12] S. Malik, F. Du, M. Monroe, E. Onukwugha, C. Plaisant, and B. Shneiderman, “Cohort comparison of event sequences with balanced integration of visual analytics and statistics,” in *Proceedings of the International Conference on Intelligent User Interfaces*, 2015, pp. 38–49.

[13] Z. Liu, B. Kerr, M. Dontcheva, J. Grover, M. Hoffman, and A. Wilson, “Coreflow: Extracting and visualizing branching patterns from event sequences,” *Computer Graphics Forum*, vol. 36, no. 3, pp. 527–538, 2017.

[14] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao, “Eventthread: Visual summarization and stage analysis of event sequence data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 56–65, 2018.

[15] Y. Wang, T.-Q. Peng, H. Lu, H. Wang, X. Xie, H. Qu, and Y. Wu, “Seek for success: A visualization approach for understanding the dynamics of academic careers,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 475–485, 2022.

[16] J. Wu, D. Liu, Z. Guo, Q. Xu, and Y. Wu, “Tacticflow: Visual analytics of ever-changing tactics in racket sports,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 835–845, 2022.

[17] R. Guo, T. Fujiwara, Y. Li, K. M. Lima, S. Sen, N. K. Tran, and K.-L. Ma, “Comparative visual analytics for assessing medical records with sequence embedding,” *Visual Informatics*, vol. 4, no. 2, pp. 72–85, 2020.

[18] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman, “Lifelines: Visualizing personal histories,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1996, pp. 221–227.

[19] M. Monroe, R. Lan, J. Morales del Olmo, B. Shneiderman, C. Plaisant, and J. Millstein, “The challenges of specifying intervals and absences in temporal queries: A graphical language approach,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 2349–2358.

[20] W. Aigner, S. Miksch, B. Thurnher, and S. Biffl, “Planninglines: novel glyphs for representing temporal uncertainties and their evaluation,” in *Proceedings of the International Conference on Information Visualisation*, 2005, pp. 457–463.

[21] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman, “Temporal event sequence simplification,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2227–2236, 2013.

[22] Y. Guo, S. Guo, Z. Jin, S. Kaul, D. Gotz, and N. Cao, “A survey on visual analysis of event sequence data,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2021.

[23] J. Chen, M. Ling, R. Li, P. Isenberg, T. Isenberg, M. Sedlmair, T. Möller, R. S. Laramee, H.-W. Shen, K. Wünsche, and Q. Wang, “Vis30k: A collection of figures and tables from ieee visualization conference publications,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 9, pp. 3826–3833, 2021.

[24] D. Deng, Y. Wu, X. Shu, J. Wu, S. Fu, W. Cui, and Y. Wu, "Visimages: A fine-grained expert-annotated visualization dataset," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, pp. 3298–3311, 2022.

[25] H. L. Gantt, *Work, wages, and profits*. The Engineering magazine Co., 1913.

[26] W. Clark, *The Gantt chart: A working tool of management*. Ronald Press Company, 1922.

[27] D. Huang, M. Tory, S. Staub-French, and R. Pottinger, "Visualization techniques for schedule comparison," *Computer Graphics Forum*, vol. 28, no. 3, pp. 951–958, 2009.

[28] L. Yuan, B. Li, S. Li, K. K. Wong, R. Zhang, and H. Qu, "Tax-scheduler: An interactive visualization system for staff shifting and scheduling at tax authorities," *Visual Informatics*, vol. 7, no. 2, pp. 30–40, 2023.

[29] S. Luz and M. Masoodian, "Visualisation of parallel data streams with temporal mosaics," in *Proceedings of the International Conference Information Visualization*, 2007, pp. 197–202.

[30] S. Luz, M. Masoodian, D. McKenzie, and W. V. Broeck, "Chronos: A tool for interactive scheduling and visualisation of task hierarchies," in *Proceedings of the International Conference Information Visualisation*, 2009, pp. 241–246.

[31] S. Luz and M. Masoodian, "Comparing static gantt and mosaic charts for visualization of task schedules," in *Proceedings of the International Conference on Information Visualisation*, 2011, pp. 182–187.

[32] ——, "A comparison of linear and mosaic diagrams for set visualization," *Information Visualization*, vol. 18, no. 3, pp. 297–310, 2019.

[33] K. Wongsuphasawat, J. A. Guerra Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman, "Lifeflow: Visualizing an overview of event sequences," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 1747–1756.

[34] M. Monroe, "Interactive event sequence query and transformation," Ph.D. dissertation, University of Maryland, College Park, 2014.

[35] T. Munzner, *Visualization analysis and design*. CRC press, 2014.

[36] E. R. Tufte, *The visual display of quantitative information*. Graphics Press, 1986.

[37] J. P. Ono, C. Dietrich, and C. T. Silva, "Baseball timeline: Summarizing baseball plays into a static visualization," *Computer Graphics Forum*, vol. 37, no. 3, pp. 491–501, 2018.

[38] B. Suchkov, M. Boguslavsky, and A. Reddy, "Development of a real-time stringlines tool to visualize subway operations and manage service at new york city transit," *Transportation Research Record*, vol. 2538, no. 1, pp. 19–29, 2015.

[39] S. Lehmann, A. Reddy, C. Samsundar, and T. Huynh, "Automated train identification and train position monitoring at new york city transit," *Transportation Research Record*, vol. 2674, no. 9, pp. 843–854, 2020.

[40] S. Di Bartolomeo, A. Pandey, A. Leventidis, D. Saffo, U. H. Syeda, E. Carstensdottir, M. Seif El-Nasr, M. A. Borkin, and C. Dunne, "Evaluating the effect of timeline shape on visualization task performance," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, p. 1–12.

[41] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[42] C. Plaisant and B. Shneiderman, "The diversity of data and tasks in event analytics," in *Proceedings of the IEEE VIS Workshop on Temporal & Sequential Event Analysis*, 2016.

[43] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman, "Aligning temporal data by sentinel events: Discovering patterns in electronic health records," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 457–466.

[44] S. Haroz and D. Whitney, "How capacity limits of attention influence information visualization effectiveness," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2402–2410, 2012.

[45] M. Stone, "How we designed the new color palettes in tableau 10," Website, July 2016, retrieved January 5th, 2024 from https://www.tableau.com/blog/colors-upgrade-tableau-10-56782. [Online]. Available: https://www.tableau.com/blog/colors-upgrade-tableau-10-56782

[46] M. Brehmer, B. Lee, P. Isenberg, and E. K. Choe, "Visualizing ranges over time on mobile phones: A task-based crowdsourced evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 619–629, 2019.

[47] V. Peña-Araya, E. Pietriga, and A. Bezerianos, "A comparison of visualizations for identifying correlation over space and time," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 375–385, 2020.

[48] M. Abdelaal, N. D. Schiele, K. Angerbauer, K. Kurzhals, M. Sedlmair, and D. Weiskopf, "Comparative evaluation of bipartite, node-link, and matrix-based network representations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 896–906, 2023.

[49] Y. Yang, W. Xia, F. Lekschas, C. Nobre, R. Krüger, and H. Pfister, "The pattern is in the details: An evaluation of interaction techniques for locating, searching, and contextualizing details in multivariate matrix visualizations," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2022.

[50] K. Koffka, *Principles of Gestalt psychology*, 2013, vol. 44.

[51] J. Heer and M. Bostock, "Crowdsourcing graphical perception: Using mechanical turk to assess visualization design," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 203–212.

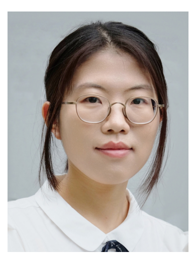
[52] P. Dragicevic, *Fair Statistical Communication in HCI*. Springer International Publishing, 2016, pp. 291–330.

[53] B. C. Kwon, V. Anand, K. A. Severson, S. Ghosh, Z. Sun, B. I. Frohnert, M. Lundgren, and K. Ng, "Dpvis: Visual analytics with hidden markov models for disease progression pathways," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 9, pp. 3685–3700, 2021.

[54] T. Baumgartl, M. Petzold, M. Wunderlich, M. Hohn, D. Archambault, M. Lieser, A. Dalpke, S. Scheithauer, M. Marschollek, V. M. Eichel, N. T. Mutters, H. Consortium, and T. V. Landesberger, "In search of patient zero: Visual analytics of pathogen transmission pathways in hospitals," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 711–721, 2021.

[55] J. Qi, C. Liu, B. C. M. B. Cappers, and H. v. d. Wetering, "Visual Analysis of Parallel Interval Events," in *EuroVis 2018 - Short Papers*, 2018.

Junxiu Tang received the BS degree in digital media technology from Zhejiang University in 2019. He is currently working toward a doctoral degree with the State Key Lab of CAD&CG. His research interests include spatial-temporal visual analysis and narrative visualization. For more information, please visit https://ahugh19.github.io/.



Fumeng Yang received her Ph.D. in Computer Science from Brown University and M.Sc. in Computer Science from Tufts University. She is currently a postdoctoral fellow at Northwestern University, funded by the CRA/CCC's CIFellows program. Her research focuses on human-centered visual computing, including visual explanations, uncertainty visualization, decision-making, perception and cognition, as well as virtual and augmented reality. For more information, please visit https://www.fmyang.com.

Jiang Wu is currently a Ph.D. candidate at the State Key Lab of CAD&CG, Zhejiang University. Before that, he received his BS degree in computer science from Zhejiang University. His research focuses on data mining and visual analytics of multivariate event sequence data, with a recent focus on visual analytics of racket sports. For more information, please visit https://wujiang5521.github.io/.

Yifang Wang is currently a Post-Doctoral Fellow in the Center for Science of Science and Innovation (CSSI) at Kellogg School of Management at Northwestern University. Before that, she obtained a Ph.D. from The Hong Kong University of Science and Technology. Her research interests lie broadly in visual analytics of computational social science and digital humanities. She also explores dara-driven art and data storytelling. For more information, please visit https://wangyifang.top/.

Jiayi Zhou is currently a senior industrial design student at Zhejiang University. Her research interest lies in Human-Computer Interaction (HCI), especially data Visualization and Human-AI Collaboration. For more information, please visit https://jiayzhou.github.io/

Xiwen Cai is currently a Ph.D. candidate in the State Key Lab of CAD&CG, Zhejiang University in China. He received his B.Sc. degree in Psychology and M.Sc degree in Computer Science from Zhejiang University. His research interests include human computer interaction and visual analytics.

Lingyun Yu is an Associate Professor at the Department of Computing, Xi'an Jiaotong-Liverpool University. She obtained the PhD degree on Scientific Visualization and Interaction Techniques from the University of Groningen in 2013. After that, she worked as a Lecturer and Associate Researcher at Hangzhou Dianzi University from 2014 to 2017, and a Postdoctoral Research Fellow at the University of Groningen and the University Medical Center Groningen from 2016 to 2019. Her research focuses on interactive visualization, immersive visualization, human-computer interaction and virtual/augmented reality. For more information, please visit http://www.yulingyun.com.

Yingcai Wu received the PhD degree in computer science from the Hong Kong University of Science and Technology. He is a professor with the State Key Lab of CAD&CG, Zhejiang University. His main research interests are in information visualization and visual analytics, with focuses on urban computing, sports science, immersive visualization, and social media analysis. Prior to his current position, he was a postdoctoral researcher with the University of California, Davis from 2010 to 2012, and a researcher with Microsoft Research Asia from 2012 to 2015. For more information, please visit http://www.ycwu.org.